Towards Multi-Task Multi-Modal Models:
A Video Generative Perspective

Yu, Lijun 于力军

CMU-LTI-24-003
April 2024

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Alexander G. Hauptmann, Chair
Yonatan Bisk
Lu Jiang
Ming-Hsuan Yang (Google, UC Merced)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies.

Keywords: Multi-Modal, Multi-Task, Video Generation, Visual Tokenization, Generative Transformer, Foundation Models, Representation Learning, Visual Understanding

To generate anything.

Abstract

Advancements in language foundation models have primarily fueled the recent surge in artificial intelligence. In contrast, generative learning of non-textual modalities, especially videos, significantly trails behind language modeling. This thesis chronicles our endeavor to build multi-task models for generating videos and other modalities under diverse conditions, as well as for understanding and compression applications.

We start with two pixel-space prototypes for separate multi-task and multi-modal setups. Despite their effectiveness, these models are constrained by task-specific modules and predefined label spaces, underscoring the need for more universally applicable designs.

Given the high dimensionality of visual data, we pursue concise and accurate latent representations. Our video-native spatial-temporal tokenizers preserve high fidelity. We unveil a novel approach to mapping bidirectionally between visual observation and interpretable lexical terms. Furthermore, our scalable visual token representation proves beneficial across generation, compression, and understanding tasks. This achievement marks the first instances of language models surpassing diffusion models in visual synthesis and a video tokenizer outperforming industry-standard codecs.

Within these multi-modal latent spaces, we study the design of multi-task generative models. Our masked multi-task transformer excels at the quality, efficiency, and flexibility of video generation. We enable a frozen language model, trained solely on text, to generate visual content. Finally, we build a scalable generative multi-modal transformer trained from scratch, enabling the generation of videos containing high-fidelity motion with the corresponding audio given diverse conditions.

Throughout the course, we have shown the effectiveness of integrating multiple tasks, crafting high-fidelity latent representation, and generating multiple modalities. This work suggests intriguing potential for future exploration in generating non-textual data and enabling real-time, interactive experiences across various media forms.

Acknowledgments

We live in an exciting era for artificial intelligence, especially video generation. I am fortunate to have started my video research in 2018 when I visited Dr. Alexander G. Hauptmann, who later became my Ph.D. advisor. Subsequently, I have studied video generation techniques and explored more modalities with Dr. Lu Jiang, my internship mentor, Dr. Ming-Hsuan Yang, and Dr. Yonatan Bisk. The last chapter details the valuable opportunities I have had to collaborate with many brilliant minds at Carnegie Mellon, Google, and other institutions. Despite "My Heart is in the Work" being a valuable factor for success, it turns out that identifying and seizing the correct opportunities play a more significant role in history. I would like to express my sincere gratitude to everyone who has facilitated my unique Ph.D. journey, which has exceeded my expectations.

While this thesis chronicles my notable technical contributions, the side effects of doing a Ph.D. include a so-called "permanent head damage", where I understand myself more profoundly than before. Our curiosity about science and truth drives us to explore the unknown realm, where our desire for fortune and fame accompanies it. My curiosity would not have been satisfied to the current extent without the generous support from the Language Technologies Institute, Google Research, Siebel Scholar Foundation, and Baidu Scholarship, as well as the projects funded by IARPA, NIST, DSTA, and Traffic 21, along with the valuable assistance from the LTI staff team. For further curiosity, "买岛建国当国家元首" might not be an infeasible path.

Ten thousand scrolls are no better than ten thousand miles. The production of this thesis took place around the globe, from the Dalton Highway in the Arctic Circle to the tropical Hainan and Honolulu islands. I appreciate everyone who has shared joyful memories, especially those who "如胡适先生一般爱好打牌" and who "如老大爷一般爱好遛弯". Notable mentions are dedicated to Dr. Wenhe Liu, Kevin, Haoyang, Xiaoyu, Zora, Xinyu, Liying, and Yufei. The unconditional love from my family and my partner has always been my source of strength in pursuing my curiosity.

Happy graduation!

Contents

Acronyms

AI Artificial Intelligence. 3

FVD Fréchet Video Distance. 85

GPT Generative Pre-trained Transformer. 111

HEVC High Efficiency Video Coding. 6, 38

LLM Large Language Model. 3, 5, 6, 25, 38, 49, 63, 84, 111, 131

MAGVIT MAsked Generative VIdeo Transformer. 63, 85

NLP Natural Language Processing. 111

PaLM Pathways Language Model. 111

SPAE Semantic Pyramid AutoEncoder. 49, 111

VDER Visually-rich Document Entity Retrieval. 25, 30

VVC Versatile Video Coding. 6, 38, 63

List of Figures

List of Tables

Introduction

Introduction

Motivation

Since its inception nearly seven decades ago, the field of Artificial Intelligence (AI) [139] has undergone significant evolutionary strides, marked by a succession of pivotal milestones. This journey witnessed the transition from rule-based expert systems [28] to the data-driven paradigms ushered in by machine learning [173], subsequently transcending to the realms of deep learning where the focus shifted from feature engineering [135] to the acquisition of representations directly from raw data [117]. The advent of foundation models [17] further epitomizes this evolutionary trajectory, promoting the sharing of knowledge across tasks, thereby obviating the need for task-specific models. Within this continuum, BERT [49] emerges as a quintessential exemplar of foundation models, epitomized by its training on extensive data via self-supervision and its proficiency in adapting to a plethora of downstream tasks. This dissertation delves into the *multi-task* versatility at the heart of methodological innovations, tracing the evolution from hierarchically structured supervised modules to cohesive, universally applicable self-supervised frameworks.

Large Language Models (LLMs) [7, 25, 191], emblematic of foundation models, are architectured with *generative* goals, crafting text outputs from diverse inputs. Notably, certain adaptations of LLMs [133, 145] have expanded their input capacity to encompass images, though their outputs are exclusively textual. This text-centric output is a manifestation of a human-conceived low-bandwidth abstraction, leading to projections of an impending scarcity of high-quality textual data [202]. In stark contrast, there exists a prodigious generation of raw signal data, particularly *videos*, which often surpasses the computational resources available for their effective utilization in training paradigms. Moreover, the progression of self-supervised generative learning for these non-textual data types significantly lags behind that of language models, thereby curtailing the potential of associated tasks. The crux of this dissertation is anchored in the exploration of generative learning aimed at producing outputs beyond text, including videos, images, and audio, thus embracing a more holistic *multi-modal* approach.

The transformer architecture [201], initially conceived to interpret text tokens, stands as the cornerstone for scalable models across various domains. Yet, when it comes to handling raw signals, such as videos, we encounter a paradigm marked by considerably greater complexity due to their inherently higher dimensional nature, encapsulating high spatial-temporal resolutions alongside multiple channels. While straightforward downscaling techniques [52] may

Modality	Evaluation	Representation	Model
Video	Understanding		*Part I Chapter 1 (A)* Multi-Task Cascaded Modules
Image + Text	Understanding		*Part I Chapter 2 (C)* Masked Transformer
Video	Generation	*Part II Chapter 3 (B)* Spatial-Temporal Vector-Quantized	*Part III Chapter 6 (A)* Masked Generative Video Transformer
Video + Image + Text	Generation + Understanding	*Part II Chapter 4 (BC)* Visual Lexical	*Part III Chapter 7 (AC)* Frozen Large Language Model
Video + Image + Audio + Text	Generation + Compression + Understanding	*Part II Chapter 5 (BC)* Scalable Visual Token	*Part III Chapter 8 (ABC)* Scalable Generative Multi-Modal Transformer

Table 1: **Overview of thesis structure** with involved modalities and evaluations. Chapters in the same row are paired latent representation and generative models. The letters in parentheses refer to the focused component in the thesis statement: (A) integrating multiple tasks, (B) crafting high-fidelity latent representation, and (C) generating multiple modalities.

suffice for discriminative models when predicting labels, they present formidable hurdles for generative models tasked with producing content in these high-dimensional spaces, particularly in the context of high-resolution image or extended video generation. To address this challenge, we embark on a journey to construct learned *latent representations* within highly-compressed spaces and subsequently formulate generative models tailored to operate within these constrained dimensions.

Thesis Organization

In this thesis, we strive to build multi-task models for multi-modal generation and understanding. We start with two pixel-space prototypes for separate multi-task and multi-modal understanding setups in Part I. Given the high dimensionality of visual data, we pursue concise and accurate latent representations in Part II. Within these multi-modal latent spaces, we study the design of multi-task generative models in Part III. Tab. 1 presents the logical structure of this thesis, with a brief over below.

Part I: Prototypes. In the first part of this thesis, we unveil a pair of prototypes designed for multi-task and multi-modal problems that encompass *video, image, and text modalities*. These prototypes showcase effective comprehension outcomes within the designated tasks, yet also underscore the need for additional exploration into generative modeling to achieve broader capabilities.

In Chapter 1, we introduce a versatile system designed to comprehend videos, achieving favorable outcomes across a variety of assessment benchmarks. This system showcases a range of capabilities, including but not limited to object detection, object tracking, foreground segmentation, activity proposal generation, and activity recognition. Its primary emphasis is on spatial-temporal activity recognition and localization, consistently delivering state-of-the-art performance across a series of benchmark scenarios. As a prototype of *multi-task video* system, its ability to incorporate new tasks is noticeably limited, achievable solely through the integration of new modules. In the following chapters, we will explore models adaptable to various tasks without major changes.

In Chapter 2, we embrace the concept of masked *vision-language* pre-training to enhance document understanding. *Masked modeling* represents a form of *generative pre-training* objective that benefits language modeling when applied with transformer architectures. In our case, the model acquires valuable *multi-modal* representations for tasks of visually-rich document entity retrieval, achieved by learning to recover the masked text and pixel information. With a singular inference step, this model resembles a prototype for mask-based generative models. In subsequent parts, we will delve into the realm of generation models trained using masked modeling techniques and inference through multi-step iterative decoding.

Part II: Multi-Modal Latent Spaces. While language models commonly function using subword tokens as their processing units, employing the direct equivalent of pixels for visual generative modeling with transformers presents more difficulties. This challenge stems from the complex, high-dimensional, and repetitive nature of pixel data, which hinders the scalability of transformers to high-resolution images or lengthy videos. As a result, the prevailing approach in contemporary visual generative models involves operating within a learned latent space. This latent space is intricately connected to the pixel space through a bidirectional mapping. In this part, we explore the concept of *multi-modal latent spaces* for generative visual modeling with transformers.

In Chapter 3, we present a *spatial-temporal vector-quantization* model designed to map a video into a discrete latent space (*i.e.* tokenization) defined by a learned codebook. Taking inspiration from the achievements of different image tokenization methods, we devise a unique architecture for this model that incorporates 3D convolutions to effectively model video data with both spatial and temporal dependencies. As a result of this design, the model achieves satisfying reconstruction fidelity even at significant compression ratios, thereby laying the foundation for the subsequent achievements of generative video transformers.

In Chapter 4, a novel strategy is introduced, which involves the mapping of visual data into the latent space of a pre-trained LLM. This model achieves its transformation by utilizing lexical token embeddings from the LLM during the process of vector quantization. This mechanism adeptly converts non-linguistic modalities, like images, into a distinct language using

the vocabulary of the LLM. By adopting a hierarchical arrangement of tokens from broad to intricate, this interpretable *visual lexical representation* effectively encompasses both semantic significance and visual intricacies. This holistic approach facilitates visual reconstruction and empowers the performance of various multi-modal tasks.

In Chapter 5, we delve into an introspective examination of the insights garnered from the explorations in Chapters 3 and 4, setting the stage for introducing an innovative *scalable visual token* representation learning approach. This approach marks a departure from traditional methods by integrating large vocabularies with a novel lookup-free quantization process and leveraging scaled causal architectures that facilitate the joint tokenization of images and videos. The proficiency of this model in visual *generation, compression, and understanding* appears favorable against existing designs. Significantly, it presents the first evidence of LLMs surpassing diffusion models in visual synthesis tasks. Moreover, it pioneers in demonstrating that a visual tokenizer, specifically tailored for video content generation, can achieve performance on par with, if not better than, established codecs such as HEVC and VVC.

Part III: Multi-Task Generative Models. Harnessing the acquired high-fidelity representations detailed in Part II, we possess the capacity to construct latent generative models that adeptly perceive, comprehend, and replicate the intricacies of the world. Within this section, our concentration is directed toward formulating techniques for data modeling and shaping task structures. Notably, we present methodologies tailored to facilitate multi-task learning using a solitary model.

In Chapter 6, we unveil a multi-task video generation model, leveraging the capabilities of *masked generative transformers*. By utilizing the spatial-temporal vector-quantized representation detailed in Chapter 3, videos are conceptualized as sequences of visual tokens within the latent space. To enrich the landscape of multi-task learning, an effective embedding technique for masked video token modeling is introduced. Remarkably, a single model, with no alterations, supports an array of conditional *video generation* tasks, encompassing scenarios where input involves a subset of pixels or an embedding. This model not only exhibits an adaptability spectrum across diverse tasks but also attains a favorable level of video generation quality, alongside an efficient sampling process.

In Chapter 7, we delve into the realm of generating *video, image, and text* through a frozen LLM, fortified by the visual lexical representation introduced in Chapter 4. Our approach introduces a progressive in-context learning methodology, empowering static LLMs to proficiently undertake both *generation and understanding* tasks spanning non-linguistic domains, including images and videos. Remarkably, even without any updates to the LLM's parameters, it showcases prowess in image and video tasks such as classification, captioning, visual question answering, text-to-image, and frame prediction.

In Chapter 8, our exploration advances as we develop *scalable generative multi-modal transformers* from the ground up, utilizing the scalable representation conceptualized in Chapter 5. This development employs modality-specific discrete tokenization to cohesively integrate text, images, videos, and audio within a decoder-only, transformer-based framework akin to LLMs. By pretraining this model on a broad array of multi-modal generative tasks using the established LLM training methodologies, we endow the model with robust capabilities for multi-task video

generation. Notably, this model represents a pioneering achievement in its ability to generate high-quality videos, complete with corresponding audio, based on a wide range of input signals.

Thesis Statement

In this thesis, we build multi-task models for generating videos and other modalities under diverse conditions, as well as for understanding and compression applications.

We show the effectiveness of

(A) *integrating multiple tasks*
into a single framework for understanding and generation;

(B) *crafting high-fidelity latent representation*
for visual data in a discrete space, optionally into text tokens; and

(C) *generating multiple modalities*
from a shared latent space, through a unified interface, and by a single model.

Tab. 1 lists the highlighted component in each chapter.

Part I

Prototypes

Part I Overview. In the first part of this thesis, we unveil a pair of prototypes designed for multi-task and multi-modal problems that encompass *video, image, and text modalities*. These prototypes showcase effective comprehension outcomes within the designated tasks, yet also underscore the need for additional exploration into generative modeling to achieve broader capabilities.

In Chapter 1, we introduce a versatile system designed to comprehend videos, achieving favorable outcomes across a variety of assessment benchmarks. This system showcases a range of capabilities, including but not limited to object detection, object tracking, foreground segmentation, activity proposal generation, and activity recognition. Its primary emphasis is on spatial-temporal activity recognition and localization, consistently delivering state-of-the-art performance across a series of benchmark scenarios. As a prototype of *multi-task video* system, its ability to incorporate new tasks is noticeably limited, achievable solely through the integration of new modules. In the following chapters, we will explore models adaptable to various tasks without major changes.

In Chapter 2, we embrace the concept of masked *vision-language* pre-training to enhance document understanding. *Masked modeling* represents a form of *generative pre-training* objective that benefits language modeling when applied with transformer architectures. In our case, the model acquires valuable *multi-modal* representations for tasks of visually-rich document entity retrieval, achieved by learning to recover the masked text and pixel information. With a singular inference step, this model resembles a prototype for mask-based generative models. In subsequent parts, we will delve into the realm of generation models trained using masked modeling techniques and inference through multi-step iterative decoding.

Chapter 1

Multi-Task Video Understanding

Overview. Activity detection stands out as a captivating computer vision endeavor that capitalizes on video streams garnered from extensively deployed cameras. Despite achieving commendable results, traditional activity detection algorithms are often formulated within specific limitations. For instance, they tend to operate with trimmed or object-focused video clips as inputs. Consequently, these algorithms struggle to effectively address scenarios involving multiple scales and instances within real-world, unconstrained video streams. Such streams remain untrimmed and encompass wide field-of-views. Moreover, the necessity for real-time analysis of streaming data renders the straightforward expansion of these methods impractical.

To overcome these issues, we propose Argus++, a robust real-time *multi-task* activity detection system for analyzing unconstrained video streams. The design of Argus++ introduces overlapping spatial-temporal cubes as an intermediate concept of activity proposals to ensure coverage and completeness of activity detection through over-sampling. The overall system is optimized for real-time processing on standalone consumer-level hardware. Extensive experiments on different surveillance and driving scenarios demonstrated its favorable performance in a series of activity detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

1.1 Motivation

Nowadays, activity detection has drawn a fast-growing attention in both industry and research fields. Activity detection in extended videos [43, 144] is widely applied for public safety in indoor and outdoor scenarios. Activity detection on streaming videos captured by in-vehicle cameras is applied for vision-based autonomous driving. The development of these applications brings several challenges. First, most of these systems take *unconstrained* videos as input, which are recorded in large field-of-views where multi-object and multi-activity occur simultaneously and continuously over time. Second, the unconstrained videos in real world are in multiple scenarios and under multiple conditions, e.g. in dynamically changed road environments from day to night in autonomous driving [178]. Third, efficient algorithms are demanded for real-time processing and responding of streaming video.

Conventional activity detection works [60, 66, 109, 193, 210] have achieved impressive performance. However, they are not suitable for real world unconstrained video understanding. Most of these works are applied under certain constraints, e.g., only for processing trimmed and/or object-centered video clips. Meanwhile, they usually are specified for certain scenarios, such as person activity, etc. Therefore, such algorithms would fail when being transferred to unconstrained videos on both efficiency and effectiveness.

Previous works [134, 164, 239] on unconstrained video analysis proposed to generate and analyze tube/tubelet proposals, which are trajectories extracted from object detection and tracking results. Tube proposal has several drawbacks. First, tube proposals failed to capture the trace of moving objects when cropping the proposals from the original videos. Therefore, learning the activities highly relied on trace would be difficult, e.g. 'vehicle turning right'. Second, the tube proposals still cannot stay away from temporal activity localization to determine the existence of the activities. Besides, most of the previous works [164] utilize non-overlapping proposals, which straightforwardly cuts the tube proposals by fixed length of temporal windows. Inevitably, such methods destroy the completeness of activities. Therefore, it would result in significant degrade of performance. Third, the objects in the tube proposal will suffer from the bounding box shift and distortion across frames, which could result in a high false alarm rate on activity detection.

To overcome the aforementioned challenges, we propose *Argus++*, an efficient robust spatial-temporal activity detection system for extended and road video activity detection. The proposed system contains four-stages: Proposal Generation, Proposal Filtering, Activity Recognition and Activity Deduplication. The major difference between *Argus++* and the former works, such as [134], is the concept of *cube* proposals. Rather than simply adapted tube proposals, i.e. cropped trajectories of detected and tracked objects, we propose to merge and crop the area of detected objects across the frames.

We summarize the contributions of this chapter as follows:

- We propose *Argus++*, a real-time activity detection system for unconstrained video streams, which is robust across different scenarios.
- We introduce overlapping spatial-temporal cubes as the core concept of activity proposals to ensure coverage and completeness of activity detection through over-sampling.
- The proposed system has achieved favorable performance in a large series of activity

detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

1.2 Prior Work

Object detection and tracking. Object detection and tracking are fundamental computer vision tasks that aims to detect and track objects from images or videos. Image-based object detection algorithms, such as Faster R-CNN [163] and R-FCN [44], have demonstrated convincing performance but are often expensive to apply on every frame. Video-based object detection algorithms [153, 261] use optical flow guided feature aggregation to leverage motion information and reduce computation. With the deep features extracted from the backbone convolutional network, multi-object tracking algorithms [217, 223] associates objects across frames based on feature similarity and location proximity.

Activity detection. In recent years, there emerged some systems designed for spatial-temporal activity detection on unconstrained videos [35, 134, 154, 164, 236, 237, 239]. Generally, theses systems first generates activity proposals and then feeds them to classification models. Since there have been a variety of video classification networks [60, 130, 193], the major focus is on the paradigm of proposals and the generation algorithm. In [35, 134], a detection and tracking framework is employed to extract whole object tracklets as tubelets, where temporal localization is required. In [164], an encoder-decoder network is used to generate localization masks on fixed-length clips for tubelet proposal extraction, which has varied spatial locations in different frames.

1.3 Argus++ Activity Detection System

We tackle the activity detection task in unconstrained videos which are untrimmed and with large field-of-views. Given an untrimmed video stream $\mathcal{V}$, the system $\mathcal{S}$ should identify a set of activity instances $\mathcal{S}(\mathcal{V}) = \{A_i\}$. Each activity instance is defined by a three-tuple $A_i = (T_i, L_i, C_i)$, referring to an activity of type C_i occurs at temporal window T_i with spatial location L_i. L_i contains the precise location of A_i in each frame, forming a tube in the timeline. As such, activity detection can often be decomposed into three aspects, i.e., temporal localization (T_i), spatial localization (L_i), and action classification (C_i).

Each of the three aspects poses unique challenges to the video understanding system. Due to its multi-dimensional nature, it remains hard to define and build a useful activity detection system under the strict setting. Therefore, we also evaluates with some loosened requirements.

Strict setting. All activity types are defined as atomic activities with clear temporal boundaries and spatial extents. The evaluation metric performs bipartite matching between predictions and ground truths.

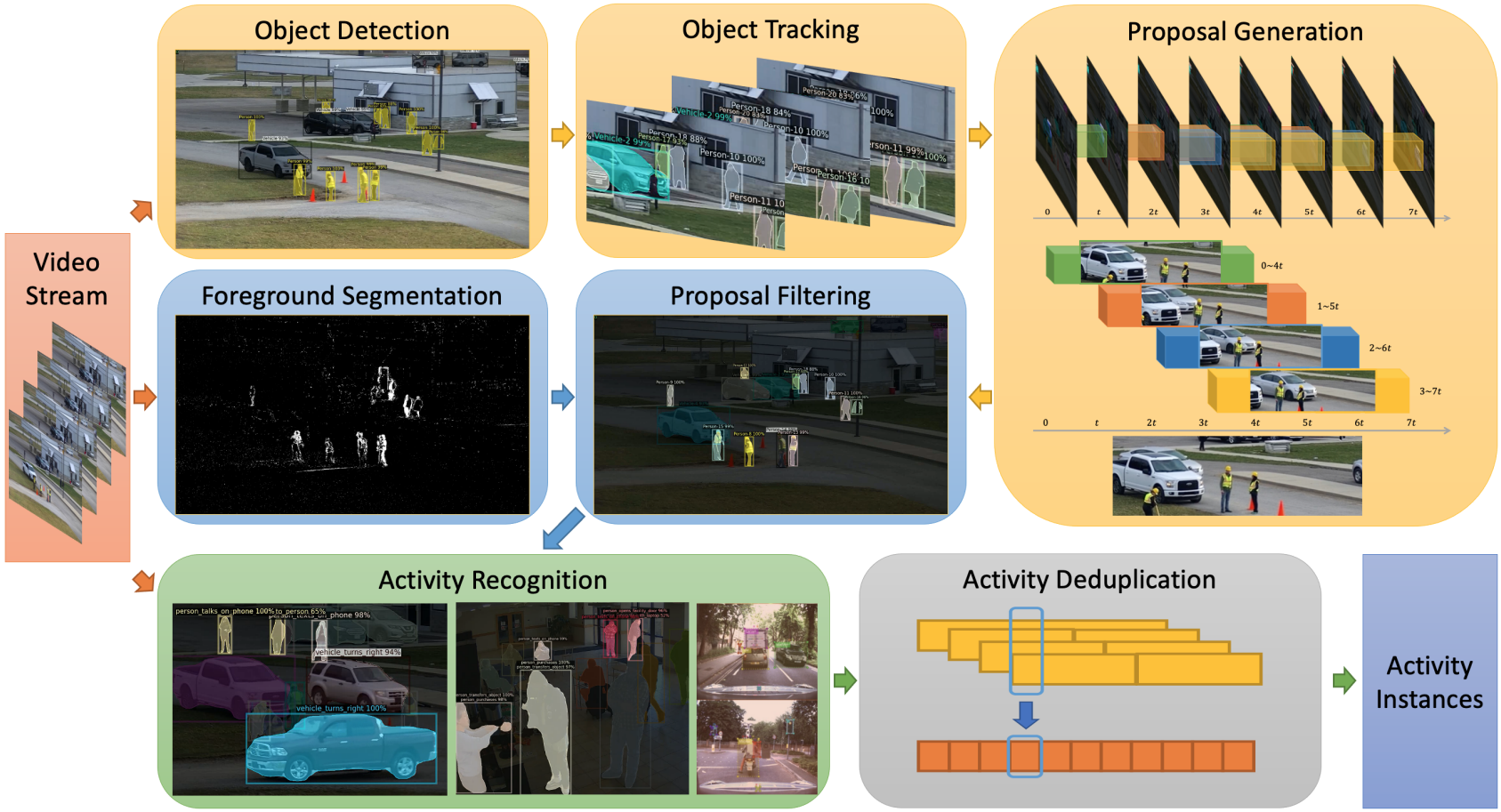


Figure 1.1: **Architecture of Argus++.** A video stream is processed frame-by-frame through object detection and tracking to generate overlapping cube proposals. With frame-level foreground segmentation, stable proposals are filtered out. Activity recognition models determine the classification scores for each proposal. These over-sampled cubes are deduplicated to produce the final activity instances.

Loosened setting. Activity types are either atomic activities within a temporal window (e.g. standing up) or continuous repetitive activities that can be cut into multiple identifiable windows (e.g. walking). The evaluation metric allows multiple non-overlapping predictions to be matched with one ground truth.

1.3.1 Argus++ System

The architecture of the proposed *Argus++* system is shown in Figure 1.1. To tackle the task of activity detection, we adopt an intermediate concept of *spatial-temporal cube proposal* with a much simpler definition than an activity instance:

$$p_i = (x_0^i, x_1^i, y_0^i, y_1^i, t_0^i, t_1^i). \tag{1.1}$$

This six-tuple design relieves the localization precision and caters modern action classification models which works on fixed-length clips with fixed spatial window.

For an input video stream, the system first generates candidate proposals with frame-wise information such as detected objects, which will be covered in Section 1.3.2. These proposals are filtered with a background subtraction model as detailed in Section 1.3.3. Then, action recognition models described in Section 1.3.4 are applied on the proposals to predict per-class confidence scores. Finally, Section 1.3.5 introduces the post-processing stage to merge and filter the proposals with scores and generate final activity instances.

1.3.2 Proposal Generation

Starting this section, we introduce each of the components of *Argus++*. The system begins by generating a set of cube proposals. They are generated based on information from frame-level object detection with multiple object tracking methods. Cubes are sampled densely in the timeline with refined spatial locations.

Detection and tracking. To conduct activity recognition, we first locate the candidate objects (in most cases, person and vehicle) in the video. For each selected frame F_i, we apply an object detection model to get objects $O_i = \{o_{i,j} \mid j = 1, \cdots, n_i\}$ with object types $c_{i,j}$ and bounding boxes $(x_0, x_1, y_0, y_1)_{i,j}$. Objects are detected in a stride of every S_{det} frames. A multiple object tracking algorithm is applied on the detected objects to assign track ids to each of them as $tr_{i,j}$.

Proposal sampling. To sample proposals on untrimmed videos without breaking the completeness of any activity instances, we propose a dense overlapping proposals sampling algorithm. As illustrated in Figure 1.2, this method ensures coverage of activities occurring at any time, with no hard boundaries. Two parameters, duration D_{prop} and stride S_{prop}, controls the sampling process. Each proposal contains a temporal window of D_{prop} frames. New proposals are generated every $S_{prop} \leq D_{prop}$ frames, possibly with overlaps. Generally, non-overlapping proposal system can be treated as a degraded case when $S_{prop} = D_{prop}$.

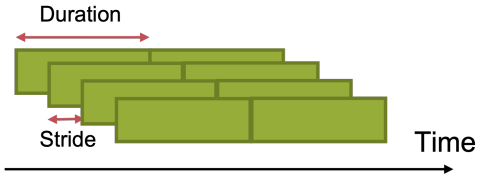


Figure 1.2: **Dense overlapping proposals**.

Proposal refinement. To generate proposals in a temporal window from t_0 to $t_1 = t_0 + D_{prop}$, we select seed track ids Tr_{t_c} from the central frame $t_c = \lfloor \frac{t_0+t_1}{2} \rfloor$. Their bounding boxes are enlarged as the union across the temporal window

$$(x_0, x_1, y_0, y_1)_k = \bigcup(\{(x_0, x_1, y_0, y_1)_{i,j} \mid t_0 \leq i \leq t_1, tr_{i,j} = tr_{t_c,k}\}), \tag{1.2}$$

where $k = 1, \cdots, n_{t_c}$. This algorithm is robust through identity switch in the tracking algorithm as it uses the stable seeds from the central frame. It also ensures the coverage of moving objects by enlarging the bounding box when it's successfully tracked. This design is helpful for efficiency optimization by allowing a large detection stride S_{det}. When later applied for activity recognition, the bounding box can be further enlarged for a fixed rate R_{enl} to include spatial context and compensate for missed tracks.

1.3.3 Proposal Filtering

For now, the proposal generation pipeline applies a frame-wise object detection with slight aid of tracking information. The motion information of video is not yet explored. To produce high quality proposals, we apply a proposal filtering algorithm to eliminate the proposals that are unlikely to contain activities.

Foreground segmentation. For each proposal, a foreground segmentation algorithm is implemented to generate a binary mask for every S_{bg} frames for each video clip. We average the value of pixel masks in its cube to get its foreground score f_i. For proposals generated by object type c, those proposals with $f_i \leq F_c$ will be filtered out. The threshold F_c is determined by allowing up to P_{pos} true proposals to be filtered out.

Label assignment. To determine the above threshold and to train the activity recognition module, we need to assign labels for each generated proposal according to the ground truth activity instances. We first convert the annotation of activity instances into the cube format, denoted as ground truth cubes, by performing dense sampling of duration D_{prop} and stride S_{prop} within each instance. For each proposal, we estimate the spatial intersection-over-union (IoU) between it and ground truth cubes in the same temporal window. Then we follow Faster R-CNN [163] in the assignment process:

- For each ground truth cube, assign it to the proposal with the highest score above S_{low}.
- For each proposal, assign it with each ground truth cube with score above S_{high}.
- For each proposal, assign it as negative if all scores are below S_{low}.

S_{high} and S_{low} are the high and low thresholds. Through this algorithm, each proposal may be assigned one or more positive labels, a negative label, or nothing. Those assigned nothing are redundant detections which will not be used in classifier training.

Proposal evaluation. To measure the quality of proposals before and after the filtering, we need a method for proposal evaluation. This can be achieved by assuming a perfect classifier in the activity recognition part, so the final metrics reflects the upper bound performance with current proposals. To do this, we simply use the assigned labels as the classification outputs and pass through the deduplication algorithm covered later. To further measure other properties of the generated proposals, we can only pass through a subset of them, such as only those with spatial IoU against ground truth above 0.5.

1.3.4 Activity Recognition

In this section, we will elaborately introduce our action recognition modules. Given the input proposal of an activity instance p_i, our action recognition model $\mathbb{V}$ will give out the confidence vector c_i:

$$\mathbb{V}(p_i) = c_i = \{c_i^1, c_i^2, ...c_i^n\}, \tag{1.3}$$

where n represents the number of target actions, and $c_i \in \mathbb{R}^n$. Limited by GPU memory size and temporal length settings of pretrained weights, we need to select t frames out of $t_1^i - t_0^i$ samples from the activity instance. To do this, we strictly followed the sparse-sampling strategy mentioned in [210] for both training and inference stage. To be specific, the video is evenly separated into t segments. From each segment, 1 frame will be randomly selected to generate the sampled clip.

To transform the action recognition modules from previous multi-class task to the realm of multi-label recognition, we modified the loss function for optimization. Instead of traditional cross entropy loss, we implemented a weighted binary cross entropy loss (wBCE), in which two weight parameters are adopted, the activity-wise weight $\mathrm{W}_a = \{w_a^1, w_a^2, ..., w_a^n\}$ and the positive-negative weight $\mathrm{W}_p = \{w_p^1, w_p^2, ..., w_p^n\}$. W_a balances the training samples of different activities and W_p balances the positive and negative samples of a specific activity. With the aligned label sequence of i^{th} instance represented as $Y_i = \{y_i^1, y_i^2, ..., y_i^n\} \in \mathbb{R}^n$. The calculation of w_a^c is derived as:

$$\hat{w}_a^c = \frac{1}{\sum_{i \in [I]} y_i^c}, \tag{1.4}$$

$$w_a^c = n \times \frac{\hat{w}_a^c}{\sum_{c \in [n]} \hat{w}_a^c}. \tag{1.5}$$

And the derivation of w_p^c is:

$$w_p^c = \frac{\sum_{i \in [I]} \mathbf{1}_{y_i^c = 0}}{\sum_{i \in [I]} y_i^c}. \tag{1.6}$$

In which, $[I]$ represents all input instances, and $[n]$ represent all target activities. Compared with vanilla BCE loss, we found wBCE loss can significantly improve the final performance on the validation set.

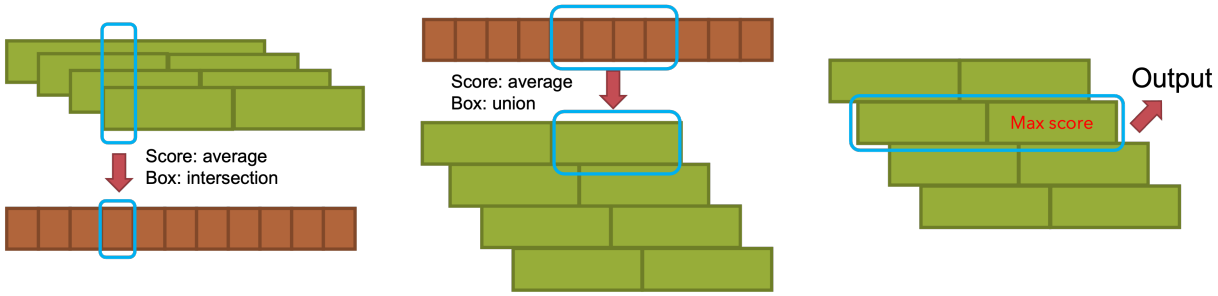


Figure 1.3: **Deduplication algorithm for overlapping proposals**.

Furthermore, we tried multiple action recognition modules and made late fusion action-wisely according to the results on the validation set. We found each classifier does show superiority on certain actions. Through the feedback from the online leaderboard, such fusion strategy can improve the final performance with noticeable margins.

1.3.5 Activity Deduplication

Overlapping instances. As the system generates overlapping proposals, it could have duplicate predictions for some of the proposals. This would result in a large amount of false alarms unless we deduplicate them. Figure 1.3 is a diagram for our deduplication algorithm which applies to each activity type with all proposals:

1. Split the overlapping cubes of duration D_{prop} and stride S_{prop} into non-overlapping cubes of duration S_{prop}. An output cube relies on all original cubes in the temporal window, with an averaged score and an intersected bounding box.
2. Merge the non-overlapping cubes of duration S_{prop} back into $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ groups of non-overlapping cubes of duration D_{prop}. An output cube is merged from $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ cubes with an averaged score and the union of bounding boxes.
3. Select the group where the maximum score resides.

The deduplication algorithm performs an interpolation upon the overlapping cubes. Each group in step 3 contains information from every classification results, maximizing the information utilization.

Adjacent instances. The above deduplication process only transforms overlapping instances to non-overlapping instances with the same duration. This would be sufficient under the *Loosened Setting*, where multiple predictions are allowed for each activity. No threshold would be needed to truncate low-confidence predictions as this happens automatically during the ground-truth matching process.

However, for the *Strict Setting*, we need to further merge adjacent cubes into integrate instances. Currently we adopt a simple yet effective algorithm, by simply merging adjacent cubes where all of them have confidence score above S_{merg}. The merged instance needs to be longer than L_{merg} to be kept in the final output.

1.4 Experimental Results

1.4.1 Implementation Details

In *Argus++*, we apply Mask R-CNN [79] with a ResNet-101 [78] backbone from Detectron2 [228] pre-trained on the Microsoft COCO dataset [131] as the object detector, with $S_{det} = 8$. Only person, vehicle, and traffic light classes are selected. For the tracking algorithm, we apply the work in [217] and reuse the region-of-interest feature from the ResNet backbone as in [152, 238].

The proposals are generated with $D_{prop} = 64$ and $S_{prop} = 16$. The labels are assigned with $S_{high} = 0.5$ and $S_{low} = 0$. The proposal filter is set with a tolerance of $P_{pos} = 0.05$.

For activity classifiers, we adopted multiple state-of-the-art models including R(2+1)D [193], X3D [60], and TRM [155]. During training, frames are cropped with jittering [210] and enlarged with $R_{enl} = 0.13$. For X3D and TRM, we trained modules with weights pre-trained on Kinetics [109]. For R(2+1)D modules, we trained modules with weighst pre-trained on IG65M [66]. We fused confidence scores from these models according to their performance on the validation set.

1.4.2 Evaluation Protocols

To measure the performance, efficiency, and generalizability of *Argus++*, we evaluate it across a series of public benchmarks. *Argus++* is applied to NIST Activities in Extended Videos (ActEV) evaluations on MEVA [43] Unknown Facility , MEVA Known Facility, and VIRAT [144] settings for surveillance activity detection. With slight modifications, it is also tested in the ICCV 2021 ROAD challenge for the action detection task in autonomous driving.

In the NIST evaluations, the metrics [11] are designed in the *Loosened Setting*, where short-duration outputs are allowed and spatial alignment is ignored. The idea is that, after processed by the system, there will still be human reviewers to inspect the activity instances with the highest confidence scores for further usages. The performance is thus measured by the probability of miss detection (P_{miss}) of activity instances within a time limit of all positive frames plus T_{fa} of negative frames, where T_{fa} is referred to as time-based false alarm rate. The major metric, $nAUDC@0.2T_{fa}$, is an integration of P_{miss} on $T_{fa} \in [0, 0.2]$.

In the ROAD challenge, the *Strict Setting* is adopted by using the mean average precision (mAP) at 3D intersection-over-union (IoU) evaluation metric. This metric does exact bipartite matching between predictions and ground truth instances, with challenging localization precision requirements.

For metrics in the following tables, ↓ means lower is better and ↑ means higher is better. For each metric, the best value is bolded and the second best is underscored. For ongoing public evaluations, the result snapshot at 11/01/2021 is presented.

1.4.3 Public Benchmarks

ActEV sequestered-data evaluations. ActEV Sequestered Data Leaderboards (SDL) are platforms where a system is submitted to run on NIST's evaluation servers. This submission format

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.3535**	**0.5747**	0.576
UMD_JHU	<u>0.4232</u>	0.6250	0.345
IBM-Purdue	0.4238	0.6286	0.530
UCF	0.4487	<u>0.5858</u>	0.615
Visym Labs	0.4906	0.6775	0.770
MINDS_JHU	0.6343	0.7791	0.898

Table 1.1: **CVPR 2021 ActivityNet challenge ActEV SDL unknown facility evaluation results**.

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.1635**	**0.3424**	0.413
UCF	<u>0.2325</u>	<u>0.3793</u>	0.751
UMD	0.2628	0.4544	0.380
IBM-Purdue	0.2817	0.4942	0.631
Visym Labs	0.2835	0.4620	0.721
UMD-Columbia	0.3055	0.4716	0.516
UMCMU	0.3236	0.5297	0.464
Purdue	0.3327	0.5853	0.131
MINDS_JHU	0.4834	0.6649	0.967
BUPT-MCPRL	0.7985	0.9281	0.123

Table 1.2: **NIST ActEV'21 SDL known facility evaluation results**.

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.3330**	<u>0.5438</u>	0.776
UCF	<u>0.3518</u>	**0.5372**	0.684
IBM-Purdue	0.3533	0.5531	0.575
Visym Labs	0.3762	0.5559	1.027
UMD	0.3898	0.5938	0.515
UMD-Columbia	0.4002	0.5975	0.520
UMCMU	0.4922	0.6861	0.614
Purdue	0.4942	0.7294	0.239
MINDS_JHU	0.6343	0.7791	0.898

Table 1.3: **NIST ActEV'21 SDL unknown facility evaluation results**.

System/Team	$nAUDC@0.2T_{fa}\downarrow$	Mean $P_{miss}@0.15T_{fa}\downarrow$	Mean $wP_{miss}@0.15R_{fa}\downarrow$
Argus++ (Ours)	**0.39607**	**0.30622**	0.81080
BUPT	0.40853	0.32489	**0.79798**
UCF	0.43059	0.34080	0.86431
M4D	0.84658	0.79410	0.88521
TokyoTech_AIST	0.85159	0.81970	0.94897
Team UEC	0.96405	0.95035	0.95670

Table 1.4: **NIST TRECVID 2021 ActEV evaluation results**.

System/Team	$nAUDC@0.2T_{fa}\downarrow$	Mean $P_{miss}@0.15T_{fa}\downarrow$	Mean $wP_{miss}@0.15R_{fa}\downarrow$
Argus++ (Ours)	**0.42307**	**0.33241**	**0.80965**
UCF	0.54830	0.50285	0.83621
BUPT-MCPRL	0.55515	0.48779	0.84519
TokyoTech_AIST	0.79753	0.75502	0.87889
CERTH-ITI	0.86576	0.84454	0.88237
Team UEC	0.95168	0.95329	0.98300
Kindai_Kobe	0.96267	0.95204	0.93905

Table 1.5: **NIST TRECVID 2020 ActEV evaluation results**.

prevents access to the test data and measures the processing time with unified hardware platform [121]. For these evaluations, *Argus++* was trained on MEVA, a large-scale surveillance video dataset with activity annotations of 37 types. We used 1946 videos in its training release drop 11 as the training set and 257 videos in its KF1 release as validation set. The optimization target is reaching better performance within 1x real-time.

Table 1.1 shows the published results from CVPR 2021 ActivityNet Challenge ActEV SDL Unknown Facility evaluation, where *Argus++* has around 20% advantage in $nAUDC@0.2T_{fa}$ over runner-up system. The test set of unknown facility is captured with a different setting from MEVA, which challenges the generalization of action detection models. Table 1.2 shows the ongoing NIST ActEV'21 SDL Known Facility leaderboard, where *Argus++* shows over 40% advantage in $nAUDC@0.2T_{fa}$. The test set of known facility shares a similar distribution with MEVA, where our system learns well and is getting nearer for real-world usages. Table 1.3 shows the ongoing NIST ActEV'21 SDL Unknown Facility leaderboard continued from ActivityNet, where *Argus++* still holds the leading position with over 5% advantage in $nAUDC@0.2T_{fa}$.

ActEV self-reported evaluations. ActEV self-reported evaluations are where only results are submitted and test data is accessible. This currently includes the annual TRECVID ActEV evaluations on VIRAT. For TRECVID, we use the official splits of VIRAT for training and validation.

Table 1.4 and 1.5 shows the leaderboard of 2020 [11, 239] and 2021 [12, 240] NIST TRECVID ActEV Challenge. In 2020, our systems is 22.8% better in $nAUDC@0.2T_{fa}$, 33.8% better in Mean

System/Team	Action@0.1 ↑	Action@0.2 ↑	Action@0.5 ↑	**Average ↑**
Argus++ (Ours)	**28.54**	**25.63**	6.98	**20.38**
THE IFY	28.15	20.97	6.58	18.57
YAAAHO	26.81	20.40	7.02	18.07
hyj	26.52	20.32	**7.05**	17.97
3D RetinaNet	25.70	19.40	6.47	17.19
LeeC	13.64	9.89	2.23	8.59

Table 1.6: **ICCV 2021 ROAD challenge action detection results**.

P_{miss}@0.15T_{fa}, and 3.5% better in Mean-wP_{miss}@0.15R_{fa} than the runner-up. Although the other competitors improved significantly in 2021, our system still holds the first place with noticeable margins.

ROAD challenge. Different from previous surveillance action detection benchmarks, the videos of ROAD Challenge[178] are gathered from the point of view of autonomous vehicles. It contains 122K frames from 22 annotated videos, where each video is 8 minutes long on average. Totally 7K tubes of individual agents are included and each tube consists on average of approximately 80 bounding boxes linked over time.

Table 1.6 shows the performance of our system with other competitors. Our system ranks the first with 20% average mAP. Although the performance is still far from satisfying in this *Strict Setting*, it demonstrates the capability of *Argus++* in adapting to precise 3D localization and moving camera view points.

1.4.4 Ablation Study

Coverage of proposal formats. We analyze the coverage of dense spatial-temporal proposals and determines the best hyper-parameters for the proposal format. By directly use ground truth cubes as proposals, we estimate the upper bound performance of both overlapping and non-overlapping proposal formats on VIRAT validation set. The results are shown in Table 1.7, where non-overlapping proposals shows at least 6.7% systematic errors while overlapping proposals with duration 64 and stride 16 only has 1.3%.

Performance of proposal filtering. We examine the quality of the proposals with and without the filter, as shown in Table 1.8 and 1.9. With the proposal evaluation procedure introduced in Section 1.3.3, the proposals are further filtered by IoU with reference and coverage of reference at levels from 0, 0.1, to 0.9 to calculate partial results.

With the dense cube proposals, the best $nAUDC$@0.2T_{fa} we can achieve with a ideal classifier is 0.08, as indicated in the IoU ≥ 0 column. The IoU and reference coverage bounded scores are used to measure the spatial matching quality of proposals, as the $nAUDC$@0.2T_{fa} does not consider spatial alignments. We can see that even with a condition of IoU ≥ 0.5, our proposal can achieve up to 0.15, which indicates the spatial preciseness. The proposal filter is also proved effective, which removed 70% of original proposals without dropping the recall level.

Duration / Stride	16	32	64	96
32	0.0705	*0.1208*	-	-
64	**0.0127**	0.0621	*0.0673*	-
96	0.0275	0.0504	-	*0.0688*

Table 1.7: **Lower bounds of $nAUDC@0.2T_{fa}$ on VIRAT validation set with different proposal formats**. Italic values are non-overlapping proposals while the others are overlapping proposals. Duration and stride are in the unit of frames.

Name	Unfiltered	Filtered
Number of Proposals	211271	62831
Positive rate	0.1704	**0.5204**
Rate of unique label	0.4558	0.4415
Rate of two labels	0.4127	0.4252
Rate of three labels	0.1017	0.1060

Table 1.8: **Statistics of proposals on VIRAT validation set**.

$nAUDC@0.2T_{fa}$	IoU			Reference Coverage		
Threshold	Average	≥ 0	≥ 0.5	Average	≥ 0.5	≥ 0.9
Unfiltered Proposals	0.2358	0.0772	0.1518	0.1562	0.1125	0.4211
Filtered Proposals	0.2352	0.0772	0.1469	0.1563	0.1099	0.4280

Table 1.9: **Proposal quality metrics on VIRAT validation set**.

Proposal Filter	$nAUDC@0.2T_{fa} \downarrow$	Processing Time
Enabled	**0.4822**	0.582
Disabled	0.5176	0.925

Table 1.10: **Effect of proposal filter on NIST ActEV'21 SDL unknown facility micro set**.

The effect of the proposal filter is also evaluate on the SDL, as shown in Table 1.10. It not only reduces processing time from 0.925 to 0.582, but also improves $nAUDC@0.2T_{fa}$ due to reduced false alarms.

1.5 Summary

In this chapter, we proposed Argus++, a robust real-time *multi-task* activity detection system for analyzing unconstrained video streams. We introduced overlapping spatial-temporal cubes as an intermediate concept of activity proposals to ensure coverage and completeness of activity detection through over-sampling. The proposed system is able to process unconstrained videos with robust performance across multiple scenarios and real-time effiency on consumer-level hardware. Extensive experiments on different surveillance and driving scenarios demonstrated its favorable performance in a series of activity detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

Future works are suggested to focus on extending the current system to more applications, such as action detection in UAV captured videos, first-person human activity understanding, etc. The proposed system could also be extended to end-to-end frameworks for better performance.

Chapter 2

Masked Multi-Modal Pre-Training

Overview. Document understanding tasks, in particular, Visually-rich Document Entity Retrieval (VDER), have gained significant attention in recent years thanks to their broad applications in enterprise AI. However, publicly available data have been scarce for these tasks due to strict privacy constraints and high annotation costs. To make things worse, the non-overlapping entity spaces from different datasets hinder the knowledge transfer between document types. In this chapter, we propose a method to collect massive-scale and weakly labeled data from the web to benefit the training of VDER models. The collected dataset, named DocumentNet, does not depend on specific document types or entity sets, making it universally applicable to all VDER tasks. The current DocumentNet consists of 30M documents spanning nearly 400 document types organized in a four-level ontology. Empowered by DocumentNet, we present a lightweight *multi-modal* architecture named UniFormer, which can learn a unified representation from text, layout, and image crops without needing extra visual pretraining. Experiments on a set of broadly adopted VDER tasks show significant improvements when DocumentNet is incorporated into the pre-training. With the recent emergence of Large Language Models (LLMs), DocumentNet provides a large data source to extend their multi-modal capabilities for VDER.

2.1 Motivation

Document understanding is one of the most error-prone and tedious tasks many people have to handle every day. Advancements in machine learning techniques have made it possible to automate such tasks. In a typical Visually-rich Document Entity Retrieval (VDER) task, pieces of information are retrieved from the document based on a set of pre-defined entity types, known as the *schema*. For example, "amount", "date", and "item name" are major parts of an invoice schema.

The current setup of VDER tasks presents several unique challenges for acquiring sufficient training data. First, the availability of raw document images is greatly limited due to privacy constraints. Real-world documents, such as a driver's license or a bank statement, often contain personally identifiable information and are subject to access controls. Second, detailed annotation is costly and typically requires intensive training for experienced human annotators. *E.g.*, it takes deep domain knowledge to correctly label different fields in complex tax forms. Finally, knowledge sharing between various types of documents is constrained by inconsistent label spaces and contextual logic. For example, the entity sets (*i.e.*, schema) could be mutually exclusive, or the same entity type could take different semantic meanings in different contexts.

A number of models have been proposed for VDER tasks with various success [8, 70, 99, 118]. To tackle the aforementioned challenges, most prior works initialize from a language model followed by BERT-style [49] pre-training on document datasets with additional layout and visual features. However, even the largest dataset currently in use, *i.e.*, IIT-CDIP [123] dataset, has a limited size and only reflects a subset of document types.

In this chapter, we introduce the method of building the DocumentNet dataset, which enables massive-scale pre-training for VDER modeling. DocumentNet is collected over the Internet using a pre-defined ontology, which spans hundreds of document types with a four-level hierarchy. Experiments demonstrated that DocumentNet is the key to advancing the performance on the commonly used FUNSD [104], CORD [146], and RVL-CDIP [123] benchmarks. More recently, LLMs [7, 145] have shown great potential for VDER tasks given their reasoning capabilities. DocumentNet provides massive-scale multimodal data to boost the performance of LLMs for document understanding.

2.2 Prior Work

Tab. 2.1 provides an overview of relevant document datasets. We divide them into three groups as introduced below.

Single-domain document datasets. Many small document datasets with entity-span annotations have been used for tasks such as entity extraction. They contain less than 100k pages from a single domain. Newer datasets come with high-quality OCR annotation thanks to the advantage of relevant tools, while older ones, such as FUNSD [104], often contain OCR errors. These datasets do not contain sufficient samples for the pre-training of a large model.

Dataset	#Samples↑	Ontology	Diverse Domains	High-quality OCR	Annotation
FUNSD [104]	199				E=3
Kleister-NDA [185]	540			✓	E=4
VRDU-Ad-buy [219]	641			✓	E=14
SROIE [100]	973				E=4
CORD [146]	1K				E=30
DeepForm [18]	1.1K			✓	E=5
VRDU-Registration [219]	1.9K			✓	E=6
Kleister-Charity [185]	2.7K			✓	E=8
DocVQA [138]	12.8K			✓	Q
CC-PDF [150]	350K		✓		
PubLayNet [258]	358K		✓		B=5
RVL-CDIP [123]	400K		✓		T=16
UCSF-IDL [150]	480K		✓		
IIT-CDIP [123]	11.4M		✓		
ImageNet [47]	1.3M images	✓	-	-	T=1K
ActivityNet [26]	20K videos	✓	-	-	T=200
DocumentNet-v1 (ours)	9.9M	✓	✓	✓	**T=398, E=6**
DocumentNet-v2 (ours)	**30M**	✓	✓	✓	**T=398, E=6**

Table 2.1: **Comparison between the proposed DocumentNet dataset and existing document understanding datasets**. Datasets from other areas also built with ontology are listed in gray. Annotation includes sample type (T), bounding box (B), entity (E), and question (Q), where the value refers to the number of classes.

Large document datasets. A few larger datasets contain over 100k pages from different domains. However, they usually do not contain OCR annotations or entity-level labels. IIT-CDIP [123] has been the largest dataset commonly used for pre-training of document understanding models. Although these datasets are large, their image quality and annotation completeness are often unsatisfactory. To complement them, we collect high-quality document images from the Internet to build the DocumentNet datasets with rich OCR and entity annotations, and demonstrate their effectiveness in document model pre-training.

Ontology-based datasets. Large labeled datasets are usually collected following an ontology. ImageNet [47] for image recognition is built upon the synsets of WordNet [142]. ActivityNet [26] for activity recognition adopts an activity taxonomy with four levels. To the best of our knowledge, DocumentNet is the first large-scale document dataset built upon a well-defined ontology.

Pretrained document models. A variety of pretrained document models have emerged, including LayoutLM [231], UDoc [70], LayoutLMv2 [230], TILT [150], BROS [92], DocFormer [8],

Figure 2.1: **Exemplar documents of each of the four top-level hierarchies**. Images are downloaded via keyword searching using a commercial search engine. All images are for demonstration purposes only and do not contain real transactions or personal information.

SelfDoc [127], LayoutLMv3 [99], *etc.*

2.3 DocumentNet Dataset

Blindly crawling the Web for images may seem easy, but it is not a practical solution since most images on the Web are not relevant to document types. We need a scalable pipeline to only select the concerned images. Broadly, this is achievable via a nearest-neighbor search of relevant keywords in a text-image joint embedding space. First, we design a set of query keywords, *i.e.*, the document ontology, and encode them into the embedding space of general Web images. Further, a nearest-neighbor algorithm retrieves the top-K semantically closest images to each query keyword. Finally, a deduplication step consolidates all retrieved images across all query keywords. Fig. 2.1 illustrates several exemplar documents retrieved using our provided keywords.

Ontology creation. Each text string in the ontology list serves as a seed to retrieve the most relevant images from the general Web image pool. An ideal ontology list should therefore cover a broad spectrum of query keywords across and within the concerned downstream application domains. Although algorithmic or generative approaches may exist, we manually curated about 400 document-related query keywords that cover domains of finance, business, personal affairs, legal affairs, tax, education, *etc.*, as illustrated in Fig. 2.2.

Image retrieval from ontology. To retrieve only the most relevant document images out of the hundreds of billions of general Web images, we leverage a highly efficient nearest neighbor pipeline by defining the distance metric as the dot product distance between the semantic

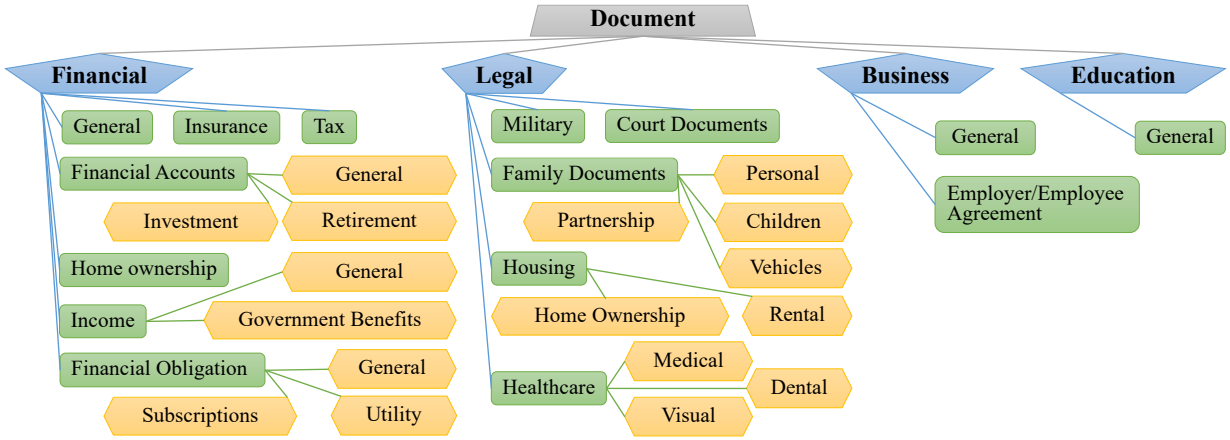


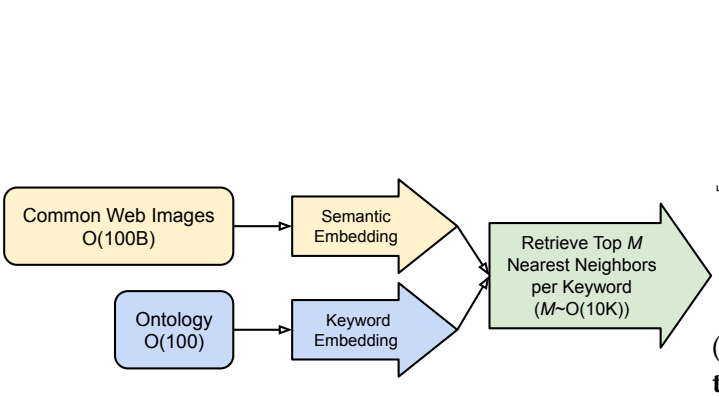
Figure 2.2: **Document ontology tree stub**, based on which the proposed DocumentNet datasets are collected. We create a document ontology with about 400 search keywords hierarchically connected by three intermediate layers.

feature vectors of the image and each of the target query keywords. Here we refer to Graph-RISE [190] for the semantic image embedding, and all query keywords are encoded into the same feature space as the images. Empirically, we pick the top 10k nearest neighbors for each query keyword. Note that the same image might be retrieved via multiple semantically similar keywords, so a de-duplication step is needed afterwards. We summarize the main pipeline steps in Fig. 2.3a. Fig. 2.3b shows statistical insights of the retrieved 30M document images with the mean and standard deviation histogram over each of the query keywords. The majority of the retrieved images are with mean distance values greater than 0.8 and standard deviations no more than 0.03, indicating high relevance to the document ontology.

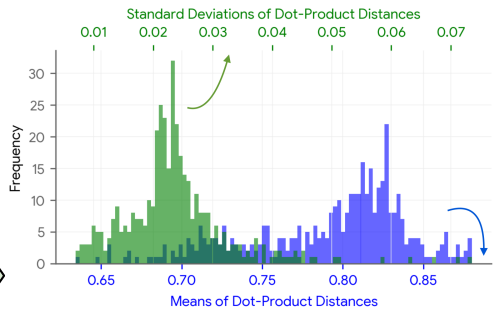
OCR and annotation. The retrieved images are fed into an OCR engine to generate a text sequence in reading order. We apply a text tagging model to weakly annotate the text segments of each sequence into 6 classes, including *email addresses, mail addresses, prices, dates, phone numbers,* and *person names*. Albeit noisy, these classification labels provide additional supervision for pre-training.

Post-processing and open-source tools. We adopt some heuristic-based filtering to improve sample quality. For example, we remove samples where the overall OCR result is poor due to blurry or noisy images. Some proprietary tools are used for scalable processing during the construction of DocumentNet, but open-source alternatives are readily available. *E.g.*, CLIP [156] for text-image embedding, Google ScaNN [72] for scalable nearest-neighbor search, Google Cloud OCR (`https://cloud.google.com/vision/docs/ocr`), and Google Cloud NLP (`https://cloud.google.com/natural-language/docs/reference/rest/v1/Entity#type`) for text tagging.

With all of the above steps, we have obtained a dataset of high-quality document images that are closely relevant to our query ontology. This dataset contains multiple modalities, including the image pixels, the OCR characters, the layout coordinates, and the segment tags.



(a) **Data collection pipeline**.



(b) **Mean and standard deviation of the dot-product distance** between the retrieved 30M document images and each query keyword. A distance of 1.0 indicates the closest semantic relevance.

Figure 2.3: **Data collection pipeline and statistics**.

	Task	Target Modality
MMLM	Multimodal Masked Language Modeling	OCR characters
MCM	Masked Crop Modeling	Image pixels
TT	Token Tagging	Segment tags

Table 2.2: **UniFormer pre-training objectives and corresponding target modalities**.

2.4 UniFormer Model

In this section, we detail our UniFormer model architecture and setups for pretraining and fine-tuning for VDER. UniFormer takes advantage of all the modalities available in DocumentNet, with the pre-training objectives listed in 2.2.

2.4.1 Multimodal Tokenization

Let $\mathbf{D} \in \mathbb{R}^{H \times W \times 3}$ be a visually-rich document image with height H and width W. We obtain a sequence of characters by applying OCR on the document image. The characters are accompanied by their bounding box coordinates. Then we perform a multimodal tokenization process as follows.

With a pre-defined text tokenizer, we first tokenize the character sequence into a sequence of text tokens $\mathbf{c}$. $\mathbf{p}$ represents the 1D position of the tokens ranging from 0 to $|\mathbf{c}| - 1$. For each token c_i, we obtain its bounding box $b_i = (x_0, y_0, x_1, y_1)_i$ by taking the union of the bounding boxes of its characters. We enlarge the bounding box by a context ratio r on each side and obtain the corresponding visual image crop v_i for each token from $\mathbf{D}$.

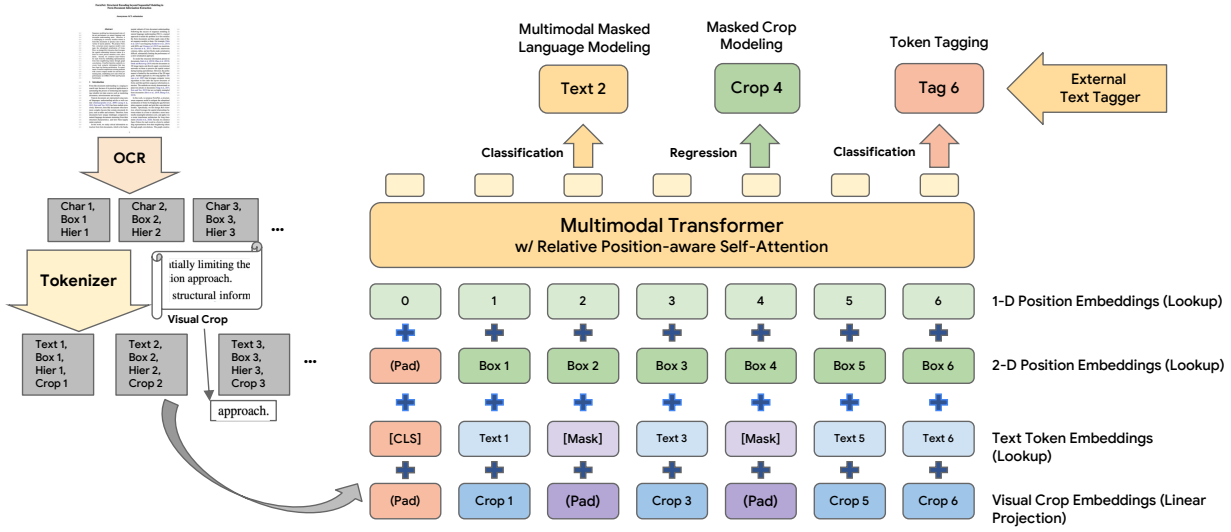


Figure 2.4: **UniFormer pre-training pipeline**. The multimodal tokenization process (left) outputs tokens with aligned image crops. The UniFormer model (right) learns a unified token representation with three objectives (top).

2.4.2 UniFormer Architecture

Fig. 2.4 illustrates the model architecture for our proposed UniFormer. UniFormer is built upon BERT [49] and utilizes its tokenizer and pretrained weights. The input for each token consists of a text embedding and a 1D position embedding for p.

Following LayoutLM [231], we add 2D position embeddings x_0, y_0, x_1, y_1, w, h, where $w = x_1 - x_0$ and $h = y_1 - y_0$. These embeddings are used to represent the spatial location of each token. All the embeddings mentioned above are obtained from trainable lookup tables.

Following LayoutLMv2 [230], UniFormer adopts relative position-aware self-attention layers by adding biases to the attention scores according to relative 1D locations $\triangle$p and relative 2D locations $\triangle \frac{x_0+x_1}{2}$, $\triangle \frac{y_0+y_1}{2}$.

Image Crop Input To model visual information, we add a crop embedding by linearly projecting the flattened pixels in the image crop, following ViT [52]. Different from prior works using either uniform patches [99], regional features [70, 127], or global features [8], our multimodal tokenization and linear embedding of image crops has the following advantages:

- It eliminates the separate preprocessing for the visual modality, such as feature extraction with a pretrained CNN [230] or manually defined patches [99].
- It obtains an aligned partition of the visual information with the text tokens, encouraging better cross-modal interaction.
- It eliminates the need for separate visual tokens as in [99, 230], resulting in a shorter token sequence and better efficiency, as shown in Fig. 2.5.
- It provides a unified joint representation for text and visual modalities in document modeling with semantic-level granularity.

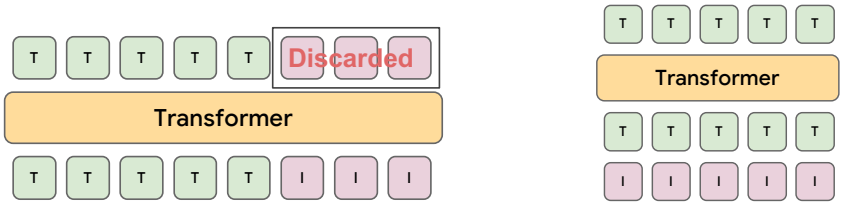


Figure 2.5: **Unaligned (left) vs. aligned (right) visual features**. The unaligned visual features result in a longer sequence but are usually discarded in downstream tasks. T: Text, I: Image.

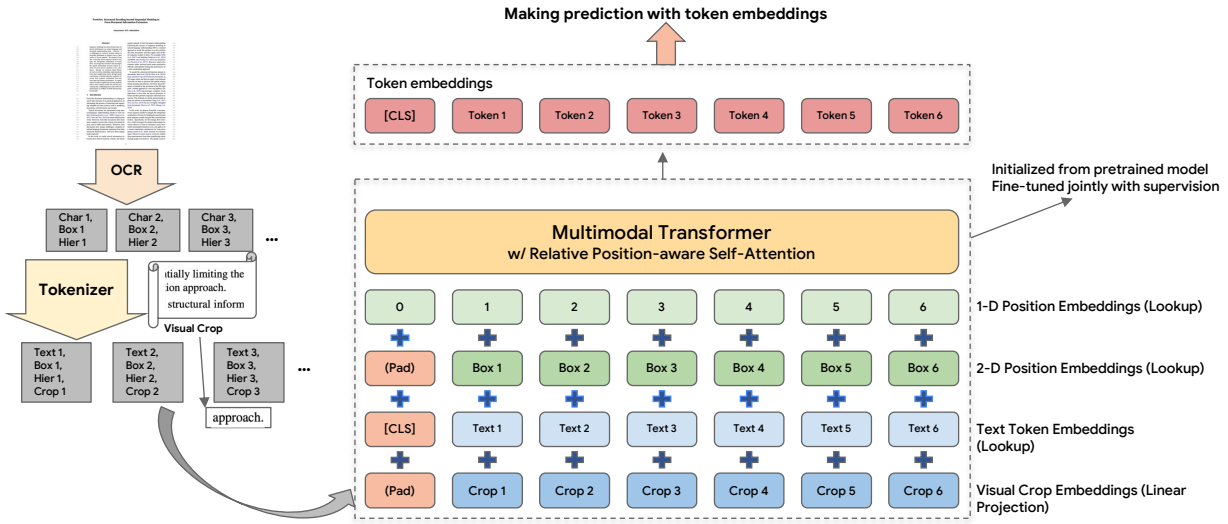


Figure 2.6: **UniFormer finetuning pipeline**.

2.4.3 Pretraining

During pretraining, we adopt the following objectives on a UniFormer parameterized by θ. For each objective, we use a separate head upon the last attention layer. Let ρ denote the always available input embeddings, including the 1D and 2D positions.

Multimodal Masked Language Modeling (MMLM) We randomly select 15% [49] of the tokens, denoted as $\mathcal{M}$, to mask and predict the language modality. In the masked language input $\overline{\mathbf{c}}$, 80% of the masked tokens are replaced with a special [MASK] token, while another 10% are replaced with a random token and the remaining 10% are kept as is. In the masked crop input $\overline{\mathbf{p}}$, crops for all masked tokens are replaced with an empty image. The language prediction is formulated as a multi-class classification problem with the cross-entropy loss as

$$\mathcal{L}_{MMLM} = \mathop{\mathbb{E}}_{\mathrm{D}} \Big[\sum_{\mathrm{c}_i \in \mathcal{M}} -\log p_\theta(\mathrm{c}_i \mid [\overline{\mathbf{c}}, \overline{\mathbf{v}}, \rho]) \Big]. \tag{2.1}$$

Masked Crop Modeling (MCM) We also predict the visual modality by reconstructing the image crops for the masked tokens in MMLM, in a way similar to MAE [80]. It is formulated as

a regression problem with a linear layer over flattened pixels. The MCM loss is defined as

$$\mathcal{L}_{MCM} = \mathop{\mathbb{E}}_{\mathbf{D}} \Big[\sum_{\mathrm{c}_i \in \mathcal{M}} \|\hat{\mathrm{v}}_i - \mathrm{v}_i\|_2^2 \Big], \tag{2.2}$$

where $\hat{\mathbf{v}} = f_\theta(\bar{\mathbf{c}}, \bar{\mathbf{v}}, \rho])$.

Token Tagging (TT) We add an extra pretraining task by predicting the tags $\mathbf{t}$ for each token in an unmasked sequence. The tags are extracted from an external text tagger as described in Sec. 2.3. Since each token may have multiple tags, it is formulated as a multi-label classification problem with the binary cross-entropy loss as

$$\begin{aligned} \mathcal{L}_{TT} = \mathop{\mathbb{E}}_{\mathbf{D}} \Big[\sum_{i,k} & -\mathrm{t}_{i,k} \log p_\theta(\mathrm{t}_{i,k} \mid [\mathbf{c}, \mathbf{v}, \rho]) \\ & - (1 - \mathrm{t}_{i,k}) \log(1 - p_\theta(\mathrm{t}_{i,k} \mid [\mathbf{c}, \mathbf{v}, \rho]))\Big], \end{aligned} \tag{2.3}$$

where $k = 1, 2, \cdots, K$ refers to the K types of tags.

Pretraining Loss The overall pretraining objective is given as

$$\mathcal{L}_{pretrain} = \mathcal{L}_{MMLM} + \alpha \mathcal{L}_{MCM} + \beta \mathcal{L}_{TT}, \tag{2.4}$$

where α, β are the corresponding loss weights.

2.4.4 Finetuning

Fig. 2.6 illustrates the pipeline for the finetuning of UniFormer. During finetuning, no tokens are masked. We adopt the following two tasks in finetuning.

Entity Extraction Entity extraction is formulated as a sequence tagging problem. The ground-truth entity spans are converted into a sequence of BIO tags $\mathbf{e}$ over all tokens. The BIO tagging is formulated as follows: $\mathbf{e}$ is initialized with all $\mathcal{O}$ tags which indicates "Other" refering to background tokens. For each entity span with type $\mathcal{T}$, start position i and end position j (both inclusive), we assign

$$\mathrm{e}_i = \mathcal{T}_{\text{Begin}}, \tag{2.5}$$

$$\mathrm{e}_{i+1} = ... = \mathrm{e}_j = \mathcal{T}_{\text{Intermediate}}. \tag{2.6}$$

The prediction of BIO tags is modeled as a multi-class classification problem with the objective as

$$\mathcal{L}_{EE} = \mathop{\mathbb{E}}_{\mathbf{D}} \Big[\sum_i - \log p_\theta(\mathrm{e}_i \mid [\mathbf{c}, \mathbf{v}, \rho]) \Big]. \tag{2.7}$$

Document Classification We use the embedding of the starting [CLS] token for document classification. The logits are predicted with an MLP head on top of the [CLS] embedding. Let l be the correct class, the objective is

$$\mathcal{L}_{DC} = \mathop{\mathbb{E}}_{\mathbf{D}} \Big[- \log p_\theta(l \mid [\mathbf{c}, \mathbf{v}, \rho]) \Big]. \tag{2.8}$$

Model	Inputs	Pre-training Data	Pre-training Objectives	FUNSD Entity F1↑	CORD Entity F1↑	RVL-CDIP Accuracy↑
BERT	T	-	MLM	60.26	89.68	89.81
LayoutLM	T + L	IIT-CDIP	MVLM	78.66	94.72	91.78
UniFormer	T + L + C	IIT-CDIP	MMLM	80.63	95.17	93.47
UniFormer	T + L + C	IIT-CDIP + *DocumentNet-v1*	MMLM	82.61	95.91	94.86
			MMLM + MCM	83.45	96.08	95.15
			MMLM + MCM + TT	**84.18**	**96.45**	**95.34**

Table 2.3: **Ablation studies on three document understanding benchmarks** regarding pretraining datasets, pretraining objectives, and model architectures. Input modalities include text (T), layout (L), and crop (C). Metrics are entity F1 and classification accuracy ×100.

2.5 Experimental Results

We pre-train UniFormer on DocumentNet and evaluate on three document understanding benchmarks.

2.5.1 Implementation Details

Pre-training. We initialize our UniFormer with BERT weights using the uncased vocabulary. The models are pre-trained using the Adam optimizer [110]. We adopt a cosine learning rate schedule with linear warmup during the first 2% steps and a peak learning rate of 10^{-4}. We use 20% of the samples for the token tagging pre-training task. The models are trained for 500K steps with a batch size of 2048 on 128 TPUv3 devices.

Downstream tasks. We evaluate the performance of pre-trained UniFormer models on three commonly used benchmarks: entity extraction on FUNSD and CORD, and document classification on RVL-CDIP. FUNSD contains 199 documents with 149 for training and 49 for evaluation. It is labeled with 3 entity types, i.e., header, question, and answer. CORD contains 1000 documents with 800 for training, 100 for validation, and 100 for testing. It is labeled with 30 entity types for receipts, such as menu name, price, *etc.* RVL-CDIP contains 400K documents in 16 classes, with 320K for training, 40K for validation, and 40K for testing.

Finetuning. For entity extraction on FUNSD and CORD, we add a multi-class classification head on top of all text tokens to perform BIO tagging. We fine-tune with a peak learning rate of 5×10^{-5}, following a schedule of linear warm-up in the first 10% steps and then linear decay. Dropout with 0.1 probability is applied in the head layers. UniFormer is fine-tuned for 1000 steps with a batch size of 32 on FUNSD and 256 on CORD. For document classification on RVL-CDIP, we add a multi-class classification head on top of the [CLS] token. We fine-tune with a constant learning rate of 10^{-5} for 15000 steps with a batch size of 2048.

Model	Initialization	Total Parameters	Pretrain Data Source	FUNSD Entity F1↑	CORD Entity F1↑	RVL-CDIP Accuracy↑
LayoutLM	BERT	113M	IIT-CDIP	78.66	94.72	91.78
	BERT + ResNet-101	160M	IIT-CDIP	79.27	-	94.42
UDoc	BERT + ResNet-50	272M	IIT-CDIP	-	-	95.05
LayoutLMv2	UniLM + ResNeXt-101	200M	IIT-CDIP	82.76	94.95	95.25
TILT	T5 + U-Net	230M	RVL-CDIP + UCSF-IDL + CC-PDF	-	95.11	95.25
BROS	BERT	110M	IIT-CDIP	83.05	95.73	-
DocFormer	LayoutLM + ResNet-50	183M	IIT-CDIP	83.34	96.33	96.17
SelfDoc	BERT + ResNeXt-101	137M	RVL-CDIP	83.36	-	92.81
LayoutLMv3*	RoBERTa	126M	IIT-CDIP	-	96.11	95.00
UniFormer	BERT	115M	IIT-CDIP + *DocumentNet-v1*	**84.18**	**96.45**	95.34

Table 2.4: Comparison with state-of-the-art document pretraining approaches on three document understanding benchmarks. * denotes a variant that does not use its proprietary tokenizer in pre-training. Metrics are entity F1 and classification accuracy ×100.

2.5.2 Quantitative Evaluation

Ablation Studies. Table 2.3 lists the ablation results for pre-training data, pre-training objectives, and model design. Compared to LayoutLM, our unified embedding of the visual modality and MMLM pre-training results in a much stronger baseline. Adding our DocumentNet into the pre-training leads to a significant performance boost across all three tasks. Further incorporating MCM and TT pre-training objectives to fully leverage DocumentNet yields consistent improvements, where the entity extraction tasks benefit more from TT and the document classification task gains more from MCM.

Comparisons with state-of-the-art. We compare the performance on the three benchmarks with state-of-the-art approaches in Table 2.4. As shown, most prior methods use stronger language or image initialization compared to our lightweight UniFormer, but all of them are only pre-trained on datasets no larger than IIT-CDIP. Although UniFormer is only using 115M parameters and BERT initialization, it outperforms all baseline approaches after pre-training on our DocumentNet dataset, with FUNSD entity F1 84.18%, CORD entity F1 96.45%, and RVL-CDIP accuracy 95.34%.

2.6 Summary

In this chapter, we proposed a *multi-modal* method to use massive and noisy web data to benefit the training of VDER models. Our approach has the benefits of providing a large amount of document data with little cost compared to usual data collection processes in the VDER domain. Our experiments demonstrated significantly boosted performance for document understand-

ing tasks. There are a number of areas that would warranty extensions or future work. First, a systematic study on the exact keywords and strategies of collecting such a data that would optimize the model outcome is yet to be studied. The methods proposed in this chapter is merely a starting point for methods along this direction. Secondly, architecture changes that specifically targets the proposed methods of massive and noisy data collecting remains an open research question. One observation we had when examining the data is that many of them contains empty forms while others have filled in content. Models that can explicitly take advantage of both formats should further boost the performance of the model.

Part II

Multi-Modal Latent Spaces

Part II Overview. While language models commonly function using sub-word tokens as their processing units, employing the direct equivalent of pixels for visual generative modeling with transformers presents more difficulties. This challenge stems from the complex, high-dimensional, and repetitive nature of pixel data, which hinders the scalability of transformers to high-resolution images or lengthy videos. As a result, the prevailing approach in contemporary visual generative models involves operating within a learned latent space. This latent space is intricately connected to the pixel space through a bidirectional mapping. In this part, we explore the concept of *multi-modal latent spaces* for generative visual modeling with transformers.

In Chapter 3, we present a *spatial-temporal vector-quantization* model designed to map a video into a discrete latent space (*i.e.* tokenization) defined by a learned codebook. Taking inspiration from the achievements of different image tokenization methods, we devise a unique architecture for this model that incorporates 3D convolutions to effectively model video data with both spatial and temporal dependencies. As a result of this design, the model achieves satisfying reconstruction fidelity even at significant compression ratios, thereby laying the foundation for the subsequent achievements of generative video transformers.

In Chapter 4, a novel strategy is introduced, which involves the mapping of visual data into the latent space of a pre-trained LLM. This model achieves its transformation by utilizing lexical token embeddings from the LLM during the process of vector quantization. This mechanism adeptly converts non-linguistic modalities, like images, into a distinct language using the vocabulary of the LLM. By adopting a hierarchical arrangement of tokens from broad to intricate, this interpretable *visual lexical representation* effectively encompasses both semantic significance and visual intricacies. This holistic approach facilitates visual reconstruction and empowers the performance of various multi-modal tasks.

In Chapter 5, we delve into an introspective examination of the insights garnered from the explorations in Chapters 3 and 4, setting the stage for introducing an innovative *scalable visual token* representation learning approach. This approach marks a departure from traditional methods by integrating large vocabularies with a novel lookup-free quantization process and leveraging scaled causal architectures that facilitate the joint tokenization of images and videos. The proficiency of this model in visual *generation, compression, and understanding* appears favorable against existing designs. Significantly, it presents the first evidence of LLMs surpassing diffusion models in visual synthesis tasks. Moreover, it pioneers in demonstrating that a visual tokenizer, specifically tailored for video content generation, can achieve performance on par with, if not better than, established codecs such as HEVC and VVC.

Chapter 3

Spatial-Temporal Vector-Quantized Representation

Overview. In this chapter, we unveil an innovative spatial-temporal vector-quantization model meticulously crafted for the purpose of video tokenization *latent representation.* Drawing inspiration from the notable accomplishments attained by diverse image tokenization techniques, we embark on the creation of a distinctive architecture tailored to this model. This architecture ingeniously integrates 3D convolutions, enabling the model to adeptly capture the intricate interplay of spatial and temporal dependencies inherent in video data. As a testament to our innovative approach, the model achieves a level of reconstruction fidelity that stands out even when operating under significant compression ratios. This achievement not only demonstrates the efficacy of our spatial-temporal vector-quantization model but also lays a robust foundation for the subsequent strides made in the realm of generative video transformers. Through this pivotal chapter, we pave the way for the exciting possibilities that lie ahead in the fusion of advanced video tokenization and generative transformer techniques.

3.1 Motivation

Transformers [201] have demonstrated strong modeling capabilities for language [49], where each sample usually consists of less than 1k text tokens. On the contrary, visual modalities are in much higher dimensional spaces. For example, a 256×256 RGB image has nearly 200k pixels, and a 16-frame clip of 128×128 has almost 1M. With super long sequences, the quadratic computation cost and the long-term dependency limit the performance of pixel-space generative modeling with transformers [37] beyond 64×64.

To achieve feasible generation at higher resolutions and better fidelity, recent generative image transformers such as DALL·E [160] and others [33, 51, 57, 235] operate in a compressed discrete latent space produced by variants of Vector-Quantized Variational AutoEncoders (VQ-VAE) [200]. The VQ-VAE encoder maps an input image into a spatially downsampled feature map. A vector quantizer (VQ) discretizes the feature map with a learned codebook, followed by a decoder reconstructing the input image.

Inspired by the successful practice for images, we design a 3D-VQ model to tokenize a video into a low-dimensional spatial-temporal manifold [63, 232]. Instead of modeling each frame independently, we introduce spatial-temporal downsampling to achieve higher compression ratios by exploiting the temporal redundancy while preserving high reconstruction quality. This way, generative transformers can model longer videos with more compact representation.

The only existing 3D-VQ model before our work is a shallow model from TATS [63]. We present comprehensive comparisons of the model architectures. Quantitative and qualitative evaluations show that our deep network design and advanced training strategy produce much higher visual quality. This successful design fosters the strong video generation results of MAGVIT [242] presented in Chapter 6.

3.2 MAGVIT 3D-VQ Model

3.2.1 Model Design

Our video VQ autoencoder is built upon the image VQGAN [57]. Let $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$ be a video clip of T frames. The VQ encoder tokenizes the video as $f_{\mathcal{T}} : \mathbf{V} \to \mathbf{z} \in \mathbb{Z}^N$, where $\mathbb{Z}$ is the codebook. The decoder $f_{\mathcal{T}}^{-1}$ maps the latent tokens back to video pixels.

The VQ autoencoder is a crucial module as it not only sets a quality bound for the generation but also determines the token sequence length, hence affecting generation efficiency. Existing methods apply VQ encoders either on each frame independently (2D-VQ) [74, 116] or on a supervoxel (3D-VQ) [63, 232]. We propose different designs that facilitate MAGVIT to perform favorably against other VQ models for video (see Tab. 3.2).

3D architecture. We design a 3D-VQ network architecture to model the temporal dynamics as follows. The encoder and decoder of VQGAN consist of cascaded residual blocks [78] interleaved by downsampling (average pooling) and upsampling (resizing plus convolution) layers. We expand all 2D convolutions to 3D convolutions with a temporal axis. As the overall downsampling rate is usually different between temporal and spatial dimensions, we use both 3D

and 2D downsampling layers, where the 3D ones appear in the shallower layers of the encoder. The decoder mirrors the encoder with 2D upsampling layers in the first few blocks, followed by 3D ones. Note that a token is not only correlated to its corresponding supervoxel but depends on other patches due to the non-local receptive field.

Inflation and padding. We initialize our 3D-VQ with weights from a 2D-VQ in a matching architecture to transfer learned spatial relationships [29], known as 3D inflation. We use inflation on small datasets such as UCF-101 [184]. We use a central inflation method for the convolution layers, where the corresponding 2D kernel fills in the temporally central slice of a zero-filled 3D kernel. The parameters of the other layers are directly copied. To improve token consistency for the same content at different locations [63], we replace the `same` (zero) padding in the convolution layers with `reflect` padding, which pads with non-zero values.

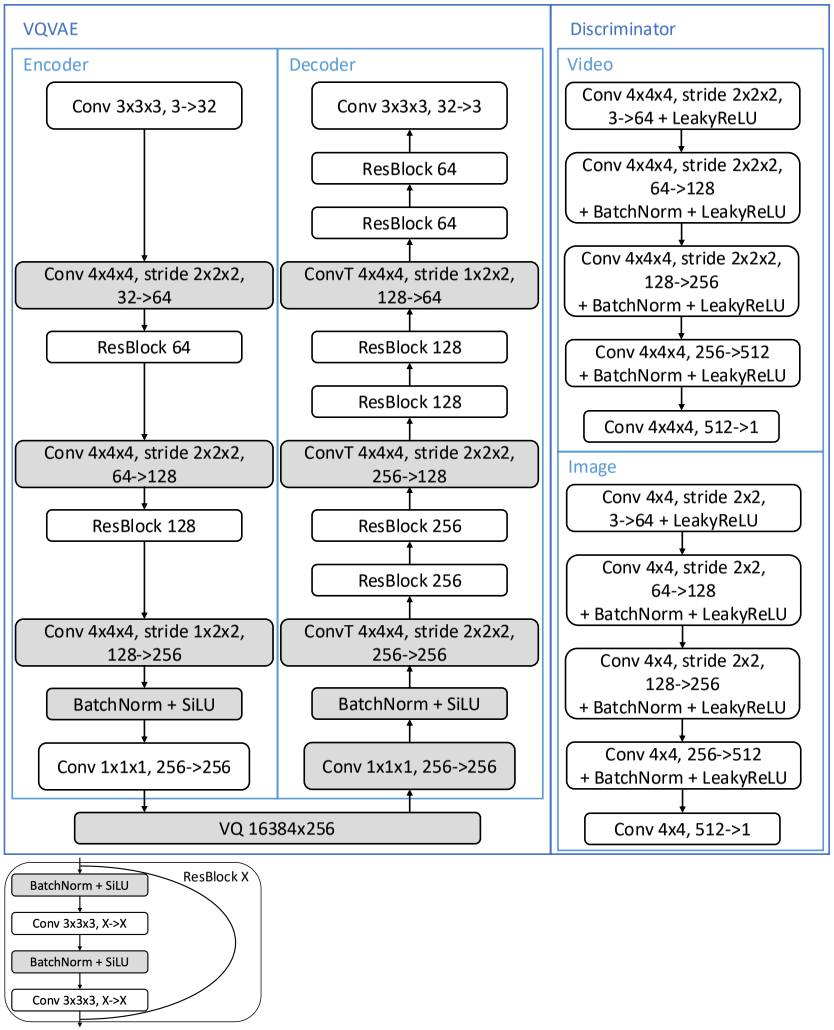
Training. We apply the image perceptual loss [57] on each frame. The LeCam regularization [194] is added to the GAN loss to improve the training stability. We adopt the discriminator architecture from StyleGAN [108] and inflate it to 3D. With these components, unlike VQGAN, our model is trained stably with GAN loss from the beginning.

3.2.2 Architecture Comparison

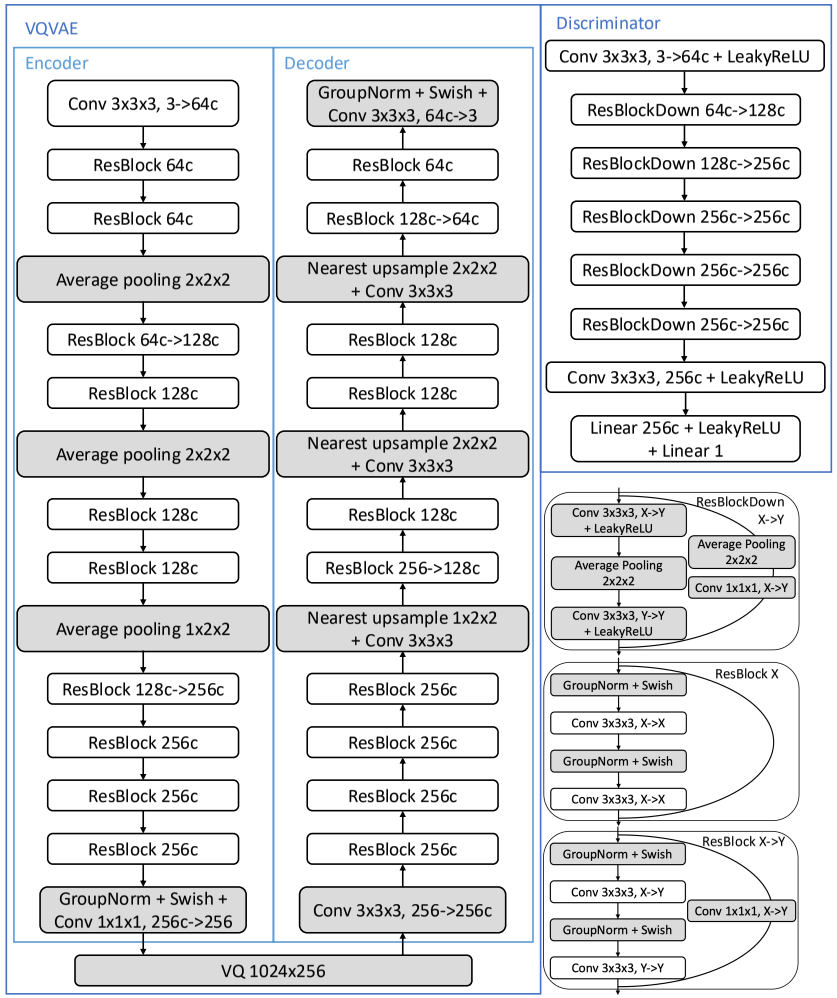
We design two variants of the MAGVIT 3D-VQ module, *i.e.*, the base (B) with 41M parameters and the large (L) with 158M parameters, excluding the discriminators.

Fig. 3.1 shows the architectures of the MAGVIT 3D-VQ module and compares it with the 3D-VQ module in TATS [63] which held the previous state-of-the-art for video generation. Compared with TATS, the major design choices in MAGVIT 3D-VQ are listed below.

- Average pooling, instead of strided convolution, is used for down-sampling.
- Nearest resizing and convolution are used for up-sampling.
- We use spatial down- and up-sampling layers near the latent space and spatial-temporal down- and up-sampling layers near the pixel space, resulting in mirrored encoder-decoder architecture.
- A single deeper 3D discriminator is designed rather than two shallow discriminators for 2D and 3D separately.
- We quantize into a much smaller vocabulary of 1,024 as compared to 16,384.
- We use group normalization [227] instead of batch normalization [101] and Swish [159] activation function instead of SiLU [83].
- We use the LeCAM regularization [194] to improve the training stability and quality.



(a) TATS [63] 3D-VQ. (32M+14M parameters)



(b) *MAGVIT* (ours) 3D-VQ.
(41M+15M parameters at $c = 1$ (B), 158M+61M at $c = 2$ (L))

Figure 3.1: **Comparison of 3D-VQ model architectures between MAGVIT and the TATS** [63]. We highlight the blocks with major differences in gray background and detail their design differences in Section 3.2.2. We train the models to quantize 16-frame clips of 128×128 resolution into $4 \times 16 \times 16$ tokens. The number of parameters in parentheses are broken down between VQVAE and discriminators.

Dataset	B	L
UCF-101	500	2000
BAIR	400	800
Kinetics-600	45	180
SSv2	135	400
nuScenes	1280	5120
Objectron	1000	2000
Web12M	5	20

Table 3.1: **Training epochs of MAGVIT 3D-VQ for each dataset.**

3.3 Experimental Results

3.3.1 Implementation Details

We follow the same learning recipe across all datasets, with the only variation in the number of training epochs. The implementation is available at Here are the training details for the 3D-VQ model:

- Video: 16 frames, frame stride 1, 128×128 resolution. (64×64 resolution for BAIR)
- Base channels: 64 for B, 128 for L.
- VQVAE channel multipliers: 1, 2, 2, 4. (1, 2, 4 for 64×64 resolution).
- Discriminator channel multipliers: 2, 4, 4, 4, 4. (2, 4, 4, 4 for 64×64 resolution)
- Latent shape: 4×16×16.
- Vocabulary size: 1,024.
- Embedding dimension: 256.
- Initialization: central inflation from a 2D-VQ trained on ImageNet with this setup.
- Peak learning rate: 10^{-4}.
- Learning rate schedule: linear warm up and cosine decay.
- Optimizer: Adam with $\beta_1 = 0$ and $\beta_2 = 0.99$.
- Generator loss type: Non-saturating.
- Generator adversarial loss weight: 0.1.
- Perceptual loss weight: 0.1.
- Discriminator gradient penalty: r1 with cost 10.
- EMA model decay rate: 0.999.
- Batch size: 128 for B, 256 for L.
- Speed: 0.41 steps/sec on 16 TPU-v2 chips for B, 0.56 steps/sec on 32 TPU-v4 chips for L.

Using more hardware resources can speed up the training. We train MAGVIT 3D-VQ models for each dataset separately. The training epochs for each dataset are listed in Tab. 3.1.

3.3.2 Quantitative Evaluations

We evaluate the design options of our MAGVIT 3D-VQ model. Tab. 3.2 lists the reconstruction FVD and IS metrics on the UCF-101 training set, which provides an upper bound for generation

Tokenizer	From Scratch		ImageNet Initialization			
	FVD↓	IS↑	FVD↓	IS↑	FVD↓	IS↑
MaskGIT [33] 2D-VQ	240	80.9	216	82.6	-	
TATS [63] 3D-VQ	162	80.6	-		-	
			Average		Central	
MAGVIT 3D-VQ-B (ours)	127	82.1	103	84.8	58	87.0
MAGVIT 3D-VQ-L (ours)	45	87.1	35	88.3	**25**	**88.9**

Table 3.2: **Comparison of tokenizer architectures and initialization methods** on UCF-101 training set reconstruction results. The 2D-VQ compresses by 8×8 spatially and the 3D-VQ compresses by 4×8×8 spatial-temporally.

VQ Tokenizer	From Scratch			ImageNet Initialization					
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
MaskGIT 2D	21.4	0.667	0.139	21.5	0.685	0.114		-	
					Average			Central	
MAGVIT 3D-L	21.8	0.690	0.113	21.9	0.697	0.103	**22.0**	**0.701**	**0.099**

Table 3.3: **Image quality metrics of different tokenizers** on UCF-101 training set reconstruction.

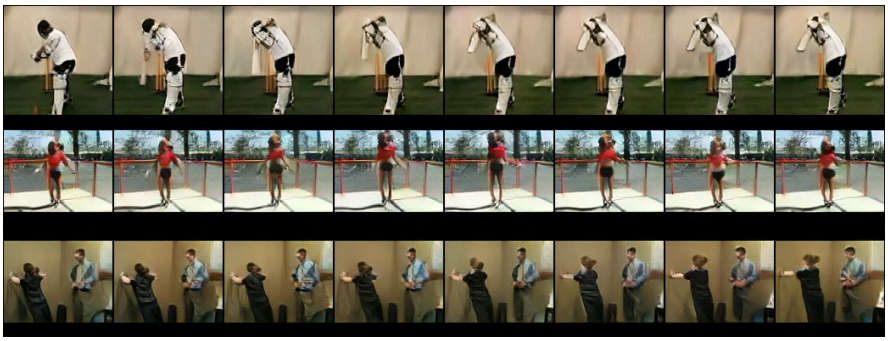
quality. In addition, we report the image quality metrics (PSNR, SSIM, LPIPS) for the VQGAN reconstruction in Tab. 3.3. We compare the proposed 3D architecture with existing 2D [33] and 3D [63] VQ architectures. We train the MaskGIT [33] 2D-VQ and our 3D-VQ with the same protocol and evaluate the official TATS [63] 3D-VQ model. We compare two inflation methods for our 3D-VQ model, *i.e.*, average [29] and central inflation.

The results show the following. First, 3D-VQ models, despite producing a higher compression rate, show better video reconstruction quality than 2D-VQ, even with fewer parameters. Second, the proposed VQ performs favorably against baseline architectures with a similar size and gets much better with a larger model. Third, ImageNet [47] initialization boosts the performance for 2D and 3D models, where the central inflation outperforms the average inflation. The results demonstrate the excellent reconstruction fidelity of our tokenizer design.

3.3.3 Qualitative Evaluations

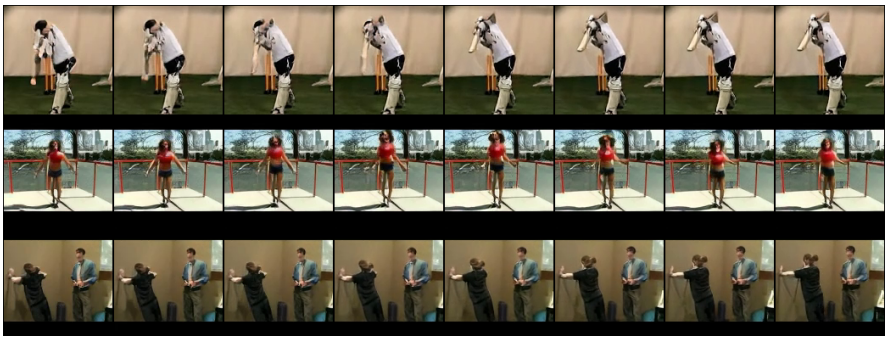
We qualitatively compare the reconstruction quality of MAGVIT 3D-VQ and the baseline models on UCF-101. In addition, we present MAGVIT's high-quality reconstruction on example YouTube videos with more samples at `https://magvit.cs.cmu.edu`.

Comparison of reconstruction quality. Fig. 3.2 compares the reconstruction quality of the three VQ tokenizers on the UCF-101, including the 2D-VQ from MaskGIT [33], the 3D-VQ from TATS [63], and MAGVIT 3D-VQ, where the videos are taken from the UCF-101 training set.



(a) MaskGIT [33] 2D-VQ

(b) TATS [63] 3D-VQ



(c) MAGVIT 3D-VQ-L (ours)

(d) Real

Figure 3.2: **Comparison of tokenizers on UCF-101 training set reconstruction**. Videos are reconstructed at 16 frames 64×64 resolution 25 fps and shown at 12.5 fps, with the ground truth in (d).

We obtain the TATS model from their official release at https://songweige.github.io/projects/tats/. We train the MaskGIT 2D-VQ and MAGVIT 3D-VQ using the same protocol on the UCF-101 dataset.

We can see that the MaskGIT 2D-VQ produces a reasonable image quality, but falls short of frame consistency which causes significant flickering when played as a video (*e.g.*, the curtain color in the first row and the wall color in the third row). TATS 3D-VQ has a better temporal consistency but loses details for moving objects (*e.g.*, the woman's belly in the second row). In contrast, our 3D VQ produces consistent frames with greater details reconstructed for both static and moving pixels.

Scalable tokenization. Since the tokenizers are trained in an unsupervised manner, they exhibit remarkable generalization performances and can be scaled to big data as no labels are required. To demonstrate this, we train a large MAGVIT 3D-VQ on the large YouTube-8M [1] dataset while ignoring the labels, and use the model to quantize randomly sampled videos on YouTube.

Figs. 3.3 and 3.4 show the original and reconstructed videos from YouTube at 240p (240 × 432) resolution with arbitrary lengths (*e.g.* 4,096 frames). Although the tokenizer is only trained with 16-frame 128×128 videos, it produces high reconstruction fidelity for high spatial-temporal resolutions that are unseen in training. Our 3D-VQ model compresses the video by a factor of 4 temporally, by 8×8 spatially, and by 2.4 (24 bits → 10 bits) per element. Our 3D-VQ model represents a N-frame video as $\frac{N}{4}$ ×30× 54 discrete tokens with a codebook of size 1024, representing a total compression rate of 614.4. Despite such high compression, the reconstructed results show stunning details and are almost indistinguishable from the real videos.

3.4 Summary

In this chapter, we introduce the MAGVIT 3D-VQ model, which tokenizes a video into a low-dimensional spatial-temporal *latent representation* with high reconstruction quality. Quantitative and qualitative evaluations show the favorable performance of our model compared to previous best video tokenizers. This successful design presents a compact but comprehensive latent space for video generation, which fosters the strong video generation results of MAGVIT [242] presented in Chapter 6.

Figure 3.3: **Our 3D-VQ model produces high reconstruction fidelity with scalable spatial-temporal resolution.** For each group, the top row contains real YouTube videos and the bottom row shows the reconstructed videos from the discrete tokens.

Figure 3.4: **Our 3D-VQ model produces high reconstruction fidelity with scalable spatial-temporal resolution**. For each group, the top row contains real YouTube videos and the bottom row shows the reconstructed videos from the discrete tokens.

Chapter 4

Visual Lexical Representation

Overview. In this chapter, a revolutionary approach is presented for the mapping of non-linguistic modalities, like images, onto the token space of an immutable Large Language Model (LLM). The innovation is encapsulated in the form of the Semantic Pyramid AutoEncoder (SPAE), which is designed to learn a *multi-modal* lexical *latent representation* for images and videos. By doing so, it empowers the cross-modal proficiencies of LLMs that have been exclusively trained on textual data. The core functionality of SPAE lies in its capacity to transmute visual content into intelligible lexical tokens that are harnessed from the vocabulary of the LLM. These tokens possess the dual capability of encapsulating both semantic significance and appearance details. As a result, they not only facilitate visual reconstruction but also serve as catalysts for a range of diverse multi-modal tasks.

4.1 Motivation

Large Language Models (LLMs) have made significant advances in solving a wide range of NLP tasks, while expanding their capabilities beyond language into other modalities. To facilitate LLMs for such cross-modal tasks, we propose to learn a vector quantizer to map an image, or some other non-linguistic ("foreign") modality, to the token space of a frozen LLM.

In this chapter, we introduce Semantic Pyramid AutoEncoder (SPAE) for a multimodal lexical representation for images and videos. SPAE converts between raw pixels and interpretable lexical tokens (or words) extracted from the language model's vocabulary. The resulting tokens capture both the semantic meaning and the fine-grained details needed for visual reconstruction, effectively translating the visual content into a language comprehensible to the LLM, and empowering it to perform a wide array of multimodal tasks.

In contrast to the majority of VQ-VAE approaches [200], our encoder maps to an interpretable discrete latent space, *i.e.*, words. As depicted in Fig. 4.1, SPAE tokens have a multi-scale representation arranged in a pyramid structure. The upper layers of the pyramid comprise semantic-central concepts, while the lower layers prioritize appearance representations that captures the fine details for image reconstruction. This design enables us to dynamically adjust the token length to accommodate various tasks, such as using fewer tokens for understanding tasks and more tokens for generation tasks.

4.2 Prior Work

Tokenization via vector quantization. VQ-VAE [200] compresses data into a discrete latent space defined by a codebook via vector quantization. VQGAN [57] enhances the reconstruction quality with adversarial and perceptual objectives. These discrete latent quantities, often referred to as *tokens*, are widely used to learn generative transformer models for image [33, 165], video [63, 203, 242], image-video [245], and audio [19, 46].Our SPAE model is built upon the VQGAN framework and applicable to different modalities.

Tokenization into lexical representations. The codebooks in typical VQGANs are learned jointly with the encoder and decoder stacks, which are not directly interpretable via natural languages. LQAE [132] replaces the learned codebook with frozen word embeddings from BERT [49] to connect with an English vocabulary. However, the LQAE tokens seldom contain semantic concepts in an image, and the reconstruction quality is worse than that with a learned codebook. Our SPAE quantizes an input sample into semantically related tokens in a multilingual vocabulary while preserving the high reconstruction quality of a VQGAN for generative tasks. In addition, SPAE tokens are organized in a multi-layer coarse-to-fine pyramid for flexible usage in different tasks.

4.3 SPAE: Semantic Pyramid AutoEncoder

Our goal is to model an image, or some other non-linguistic modality (*e.g.*, video or audio), as a language sequence that LLMs can comprehend. *Semantic Pyramid AutoEncoder* (SPAE) gener-

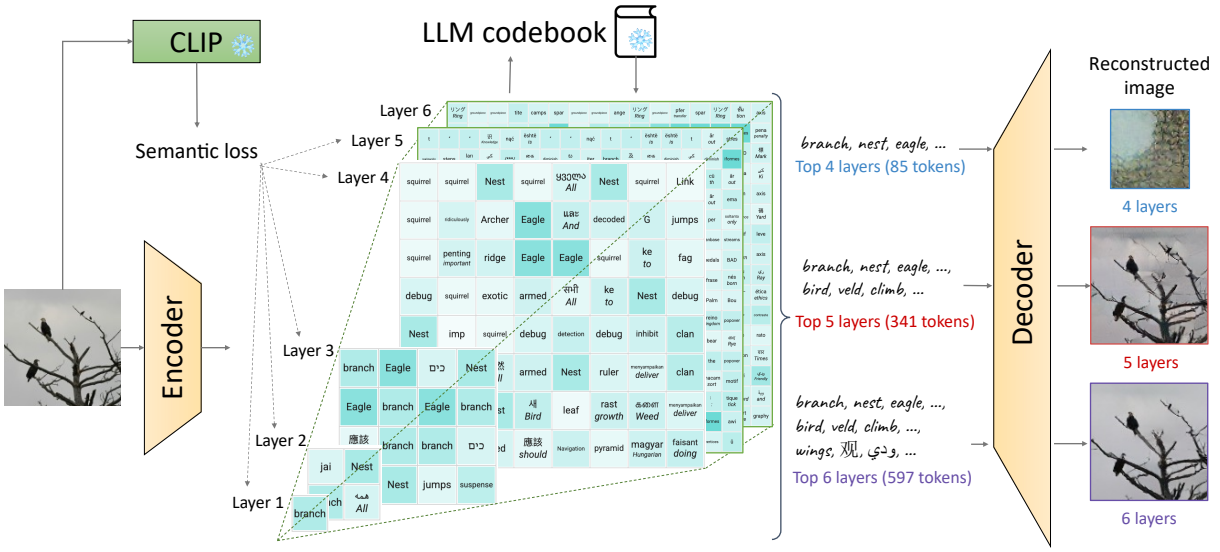


Figure 4.1: **Framework of the proposed SPAE model**. An image is encoded into a pyramid of lexical tokens capturing semantic concepts and appearance details necessary for reconstruction.

ates a lexical word sequence with dynamically adjustable length that carries rich semantics and retains fine details for signal reconstruction. To work with a frozen LLM via in-context learning, we introduce a progressive in-context denoising method to facilitate image generation. We use the image modality in this section to introduce our SPAE model in 2D, and later showcase the results of a 3D variant with the video modality in our experiments.

4.3.1 Model Architecture

Our SPAE model extends the VQ-VAE [200] framework, which comprises an encoder, a quantizer, and a decoder. The CNN encoder maps an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ into continuous embeddings $\mathbf{Z} \in \mathbb{R}^{h \times w \times c}$. Each element $\mathbf{z} \in \mathbf{Z}$ is then passed through the quantizer, which assigns it to the closest entry in a codebook, resulting in the quantized embedding. Let $\hat{\mathbf{Z}}$ represent the quantized embeddings for the entire image. The CNN decoder receives $\hat{\mathbf{Z}}$ as input and generates the reconstructed image $\hat{\mathbf{I}}$. Below we highlight the design differences in SPAE.

As illustrated in Fig. 4.1, SPAE generates lexical tokens arranged in a pyramid structure, which contains semantic concepts in the upper layers and appearance with progressively refined details in the lower layers. We introduce a semantic loss to encourage the usage of conceptually relevant tokens.

Frozen language codebook. To generate lexical tokens, we utilize a pretrained LLM codebook $\mathbb{C} = \{(k, \mathbf{e}(k)) \mid k \in \mathbb{T}\}$ and freeze it during training, where $\mathbb{T}$ is a subset of the LLM vocabulary. Here, $\mathbf{e}(\cdot)$ produces the text embedding for a sub-word k which may be obtained from any layer of the LLM. Since the codebook is aligned with the language vocabulary, we use the terms "token" and "word" interchangeably.

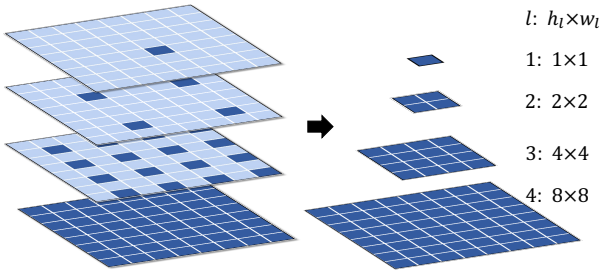


Figure 4.2: **Dilation subsampler visualization**.

Token pyramid. The SPAE quantizer produces D layers of tokens where the tokens at layer l are denoted as $\mathbf{k}_l \in \mathbb{T}^{h_l \times w_l}$. Prior works use Residual Quantization (RQ) to generate multi-layer tokens [119, 249]. In these methods, tokens from all layers have uniform shapes and do not carry specific semantic meanings. In contrast, we propose a pyramid token structure by enforcing the constraint $h_l \leq h_{l+1} \wedge w_l \leq w_{l+1}$. The pyramid structure is purposefully designed to concentrate semantics within the upper layers of the pyramid. This design allows for representing semantic concepts with notably fewer tokens, *e.g.*, as few as five tokens for understanding tasks. The high token efficiency stems from the pyramid structure, as a conventional layer without pyramid structures needs a minimum of hw tokens (*e.g.*, 256) to represent the image. Token efficiency is crucial for in-context learning as it enables the accommodation of more examples within the context. Fig. 4.2 illustrates the formation of the pyramid with a dilation subsampler $\mathbf{P}(l)$, which selects the positions for quantization at layer l as

$$\mathbf{P}(l) = \{(h'i - \left\lceil \frac{h'}{2} \right\rceil + 1, w'j - \left\lceil \frac{w'}{2} \right\rceil + 1) \mid (i, j) \in ([1, h_l] \times [1, w_l]) \cap \mathbb{Z}^2\}, \tag{4.1}$$

where $h' = \frac{h_D}{h_l}$, and $w' = \frac{w_D}{w_l}$ are the downsample ratios.

Streaming average quantization. For each embedding $\mathbf{z}$ at position (x, y), we obtain its discrete tokens k_l sequentially from layer 1 to D. At layer l, if $(x, y) \in \mathbf{P}(l)$, the quantizer assigns discrete token $k_l = \arg\min_{k \in \mathbb{T}} \|\mathbf{z}_l - \mathbf{e}(k)\|_2^2$, where $\mathbf{z}_l$ is the current layer embedding, calculated from

$$\mathbf{z}_l = \mathbf{z} + \sum_{i=1}^{l-1} \mathbf{1}_{(x,y) \in \mathbf{P}(i)} (\mathbf{z} - \mathbf{e}(k_i)). \tag{4.2}$$

The quantized embedding reconstructed with the first l layers is given by the average of the existing token embeddings as

$$\hat{\mathbf{z}}_{\leq l} = \frac{\sum_{i=1}^{l} \mathbf{1}_{(x,y) \in \mathbf{P}(i)} \mathbf{e}(k_i)}{\sum_{i=1}^{l} \mathbf{1}_{(x,y) \in \mathbf{P}(i)}}. \tag{4.3}$$

Using the input of $\hat{\mathbf{Z}}_{\leq l}$ from tokens up to layer l, the decoder can progressively reconstruct the image with dynamic token lengths, resulting in gradually improved quality with refined appearance details. We term this approach as *Streaming Average Quantization* (SAQ) due to its resemblance to computing the average on streaming data, where $\hat{\mathbf{z}}_{\leq l+1} = \hat{\mathbf{z}}_{\leq l} + \frac{1}{\hat{l}+1} \mathbf{e}(k_{l+1}), \hat{l} =$

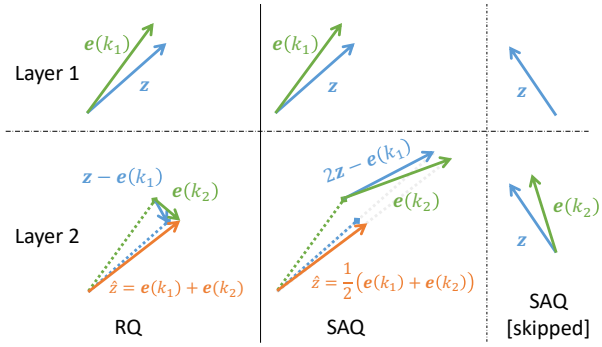


Figure 4.3: **Comparison between RQ and SAQ**. We show a 2-layer quantization process in a 2-dimensional space as an example. At layer l, we use blue for the current remainder embeddings $\mathbf{z}_l$, green for current post-quantization embeddings $\mathbf{e}(k_l)$, and orange for the reconstructed embeddings up to layer l as $\hat{\mathbf{z}}_{\leq l}$.

$\sum_{i=1}^{l} \mathbf{1}_{(x,y)\in \mathbf{P}(i)}$. Fig. 4.3 compares our proposed SAQ with Residual Quantization (RQ) [119, 249]. RQ is applicable but yields worse results in this context, as revealed by our ablation studies. This can be attributed to (1) varying scales of embeddings in residual layers, potentially dividing the codebook into multiple parts, and (2) misalignment in the summation of word embeddings, which undermines learning semantically meaningful tokens in later layers. At layer 2, the SAQ remainder embedding $\mathbf{z}_2 = 2\mathbf{z} - \mathbf{e}(k_1)$ is at a more similar scale to $\mathbf{z}$, compared to the RQ remainder $\mathbf{z} - \mathbf{e}(k_1)$. We find that the scale consistency promotes better utilization of the frozen language codebook despite a large number of layers being used. Due to the pyramid structure, quantization in the first few layers may be skipped for those positions not selected by the dilation subsampler. Considering the scale consistency across quantization layers, the use of SAQ is more appropriate in this case.

4.3.2 Training Objectives

Semantic loss. We encourage the semantic similarity between the image $\mathbf{I}$ and each lexical token k denoted by $s(\mathbf{I}, k)$. During training, we build per-layer candidate token pools as

$$\mathbf{C}_l(\mathbf{I}) = \{k \in \mathbb{T} \mid s(\mathbf{I}, k) \geq \rho_l\}, \tag{4.4}$$

where ρ_l is a threshold. We set $\rho_l \geq \rho_{l+1}$ to allow deeper layers to have a larger pool of candidate tokens while sacrificing some semantics.

To define the similarity score, we employ a pretrained CLIP model [156]. In more details, let $f_{\mathcal{I}}$ and $f_{\mathcal{T}}$ be a pair of image and text CLIP embedding functions. We precompute the text feature for each token $k \in \mathbb{T}$ as

$$f'_{\mathcal{T}}(k) = \frac{1}{|\mathbf{p}|} \sum_{i=1}^{|\mathbf{p}|} f_{\mathcal{T}}(\mathrm{p}_i(k)), \tag{4.5}$$

where $\mathbf{p}$ is a list of prompt templates, such as `"a photo of ..."`. During training, we extract the image feature $f_{\mathcal{I}}(\mathbf{I})$ and compute the dot-product similarity as $\mathbf{s}'(\mathbf{I}, k) = f_{\mathcal{I}}(\mathbf{I}) \cdot f'_{\mathcal{T}}(k)$.

The similarity score is then normalized to account for the varying scales across different images.

$$\mathbf{s}(\mathbf{I}, k) = \frac{\mathbf{s}'(\mathbf{I}, k) - \min_j \mathbf{s}'(\mathbf{I}, j)}{\max_j \mathbf{s}'(\mathbf{I}, j) - \min_j \mathbf{s}'(\mathbf{I}, j)}. \tag{4.6}$$

We define the semantic loss for the encoder parameters θ_e as

$$\mathcal{L}_{\text{semantic}}(\theta_e; \mathbf{I}) = \mathop{\mathbb{E}}_{l \in [1, D']} \mathop{\mathbb{E}}_{\mathbf{z}_l} \mathop{\mathbb{E}}_{c \in \mathbf{C}_l(\mathbf{I})} -\log \frac{\exp(-\|(\mathbf{z}_l - \mathbf{e}(c)\|_2^2)}{\sum_{k \in \mathbb{T}} \exp(-\|\mathbf{z}_l - \mathbf{e}(k)\|_2^2)}, \tag{4.7}$$

where we randomly sample semantically similar target codes c for each layer embedding in the first D' layers.

Appearance loss. Using an improved objective from [242], the appearance loss is calculated:

$$\mathcal{L}_{\text{appearance}}(\theta_e, \theta_d; \mathbf{I}) = \|\mathbf{I} - \hat{\mathbf{I}}\|_2^2 + \beta \sum_{l=1}^{D} \|\mathbf{Z} - \text{sg}(\hat{\mathbf{Z}}_{\leq l})\|_2^2 + \lambda \mathcal{L}_{\text{GAN}} + \eta \mathcal{L}_{\text{Perceptual}} + \phi \mathcal{L}_{\text{LeCAM}}, \tag{4.8}$$

where $\mathcal{L}_{GAN}$, $\mathcal{L}_{Perceptual}$, and $\mathcal{L}_{LeCAM}$ are the VQGAN [63], perceptual [105], and LeCAM [194] losses. In addition, $\text{sg}(x)$ is the stop-gradient operation. The appearance loss is applied to both the encoder θ_e and decoder parameters θ_d, excluding the frozen codebook embedding.

Overall objective. To stabilize the training and balance between appearance and semantics, we add a dynamic weight for the semantic guidance loss as $w = \text{sg}\left(\frac{\mathcal{L}_{\text{appearance}}(\mathbf{I})}{\mathcal{L}_{\text{semantic}}(\mathbf{I})}\right)$. The total training loss excluding the GAN discriminator is

$$\mathcal{L}_{\text{SPAE}}(\theta_e, \theta_q) = \mathop{\mathbb{E}}_{\mathbf{I}} \Big[\mathcal{L}_{\text{appearance}}(\theta_e, \theta_q; \mathbf{I}) + \alpha w \mathcal{L}_{\text{semantic}}(\theta_e; \mathbf{I})\Big]. \tag{4.9}$$

4.4 Experimental Results

4.4.1 Experimental Settings

Language codebooks. To verify the compatibility with different LLMs, we train two variants of SPAE, namely $\text{SPAE}_{\text{PaLM}}$ and SPAE_{GPT}. The $\text{SPAE}_{\text{PaLM}}$ codebook is taken from the input embedding layer of a PaLM 2-S checkpoint with a 65k vocabulary of the most frequent sentence pieces. SPAE_{GPT} uses a byte-pair encoding vocabulary with 99k UTF-8 tokens (https://github.com/openai/tiktoken), where we obtain the contextual token embeddings from OpenAI `text-embedding-ada-002` (https://platform.openai.com/docs/models/embeddings).

Image SPAE. An image SPAE encodes a 128×128 image into 16×16 embeddings. Following the VQGAN [57] architecture, we use 128 base filters with channel multipliers [1, 2, 2, 4] and 2 residual blocks at each scale, which results in 59M parameters in total. The embeddings are quantized into a token pyramid of 6 layers where each layer has $2^k \times 2^k$ tokens and $k =$

$[0, 1, 2, 3, 4, 4]$. We apply semantic guidance loss to the first five layers, with thresholds of 0.98, 0.95, 0.9, 0.85, and 0.8. The CLIP with a ViT-L/14 [52] vision backbone is used. We use 80 prompt templates from the zero-shot ImageNet classification setup to precompute the CLIP text embeddings for the vocabulary. In addition, we use the Adam [110] optimizer with loss weights $\alpha = 1, \beta = 0.33, \lambda = 0.1, \eta = 0.1, \phi = 10^{-4}$ and a learning rate of 10^{-4} following a linear warmup/cooldown and root square decay schedule. Following the prior work [132], SPAE is trained on the ImageNet ILSVRC2012 [47] dataset. We train with a batch size of 256 for 450k steps, which takes 1.4k TPUv3-hours.

Image SPAE-8. In addition to the primary SPAE model with six pyramid layers, we also train an SPAE-8 model with eight layers to conduct a more in-depth analysis of the coarse-to-fine reconstruction process. The two extra layers each contain 16×16 tokens. The semantic loss is still applied on the first 5 layers as in the primary model.

Video SPAE. We initialize a video SPAE by VQGAN inflation [242] from a pretrained image SPAE, which encodes 16 frames at 128×128 resolution into 4×16×16 embeddings. A video SPAE consists of 176M parameters. The pyramid layers contain 1×1×1, 1×2×2, 1×4×4, 2×8×8, 4×16×16, and 4×16×16 tokens. The video embedding is obtained as the average CLIP embedding for all frames. The model is trained on the Kinetics-600 [30] dataset which contains 384k videos. We train with a batch size of 512 for 130k steps, which takes 5.8k TPUv4-hours.

4.4.2 Quantitative Evaluations

We compare the image and video reconstruction quality using the tokens produced by SPAE and the VQGAN baseline used in state-of-the-art image [33, 34, 126] and video generation [242]. We use FID [85], Inception Score (IS) [172], and LPIPS [254] to compare with the image VQGAN from MaskGIT [33] on the ImageNet validation set, and FVD [198] to compare the 3D-VQGAN from MAGVIT [242] on the Kinetics-600 validation set. To quantify the semantics, we compute the CLIP and relative CLIP scores (Eq. (4.6)), both averaged across all lexical tokens.

Comparison to VQGAN. The results are presented in Tab. 4.1 for image models and in Tab. 4.3 for video models. Unlike VQGAN tokens, which lack specific semantic meaning, SPAE tokens demonstrate high semantic CLIP scores, more evident in the lower layers. As the number of layers increases, more tokens are utilized, resulting in improved reconstruction quality. This flexibility allows for dynamic adjustment of the token length to accommodate various tasks, such as using fewer tokens for understanding tasks. While SPAE may have more lossy reconstruction compared to VQGAN when using a similar number of tokens, this is compensated by going into deeper layers, as shown in the layer 6 of SPAE in Tab. 4.1. In Fig. 4.4, we show the training curves of FID, IS, LPIPS, and CLIP score of SPAE and VQGAN. As shown, within 40% of the training steps, SPAE shows better FID than the final VQGAN checkpoint. The CLIP score keeps improving as the training proceeds, while the LPIPS saturates quite early. In Tab. 4.2, we showcase the scalability of our model by training on the ImageNet-21k dataset with 13M images and list the comparable variant from LDM [165] as a reference.

Model	# Layers : # Tokens	FID↓	IS↑	LPIPS↓	CLIP↑	Classification Accuracy↑
VQGAN	1: 256	5.48	119.69	0.13	n/a	19.6
+ frozen codebook	1: 256	7.44	101.39	0.17	0.1464	19.5
+ semantic guidance	1: 256	5.17	124.41	0.13	0.1518	46.2
+ 2-layer RQ	1: 256	11.94	89.01	0.22	0.1595	56.2
	2: 512	6.05	113.93	0.15	0.1547	-
+ 2-layer SAQ	1: 256	12.30	93.33	0.21	0.1613	56.6
	2: 512	5.08	125.27	0.14	0.1595	-
+ 6-layer pyramid SAQ	1: 1	-	-	-	**0.1879**	52.0
	2: 5	-	-	-	0.1868	64.2
	3: 21	-	-	-	0.1815	**65.1**
SPAE (ours)	4: 85	-	-	-	0.1711	58.5
	5: 341	9.49	109.46	0.17	0.1604	46.3
	6: 597	**4.41**	**133.03**	**0.12**	0.1577	-
no per-layer thresholds	6: 597	4.33	122.25	0.11	0.1650	59.4 (layer 3)
no dynamic semantic weight	6: 597	9.00	85.14	0.19	0.1847	65.1 (layer 3)
no perceptual loss	6: 597	40.47	33.41	0.20	0.1994	69.5 (layer 3)
SPAE-8 (ours)	1: 1	-	-	-	**0.2051**	-
	2: 5	-	-	-	0.2046	-
	3: 21	-	-	-	0.2012	-
	4: 85	-	-	-	0.1896	-
	5: 341	43.42	49.78	0.32	0.1709	-
	6: 597	8.93	116.12	0.18	0.1667	-
	7: 853	4.78	135.01	0.13	0.1647	-
	8: 1109	**3.89**	**140.55**	**0.11**	0.1634	-

Table 4.1: **Comparison of reconstruction quality and semantic relevance for image tokenization**. We compare SPAE and VQGAN baselines used in state-of-the-art image [33, 34, 126] generation models with ablation studies on codebook, training objective, quantization method, and token structure at 128×128 resolution. Classification accuracy is evaluated under the mini-ImageNet 5-way 1-shot setup (*c.f.* Section 7.4.1).

Method	Dataset	# Layers : # Tokens	FID↓	IS↑	LPIPS↓
VQGAN (LDM [165])	OpenImages	1: 256	5.15	144.55	-
SPAE (ours)	ImageNet-21k	6: 597	3.08	173.79	0.19

Table 4.2: **Comparison of reconstruction quality with scalability** between SPAE and VQGAN baselines on 256×256 images.

Method	# Layers : # Tokens	FVD↓	CLIP↑	Relative CLIP↑
VQGAN	1: 1024	6.79	n/a	n/a
SPAE (ours)	1: 1	-	**0.2061**	**0.8425**
	2: 5	-	0.2056	0.8402
	3: 21	-	0.2032	0.8286
	4: 149	-	0.1896	0.7620
	5: 1173	52.28	0.1670	0.6531
	6: 2197	**6.35**	0.1635	0.6367

Table 4.3: **Comparison of reconstruction quality and semantic relevance for video tokenization** between SPAE and the VQGAN baselines used in state-of-the-art video [242] generation models.

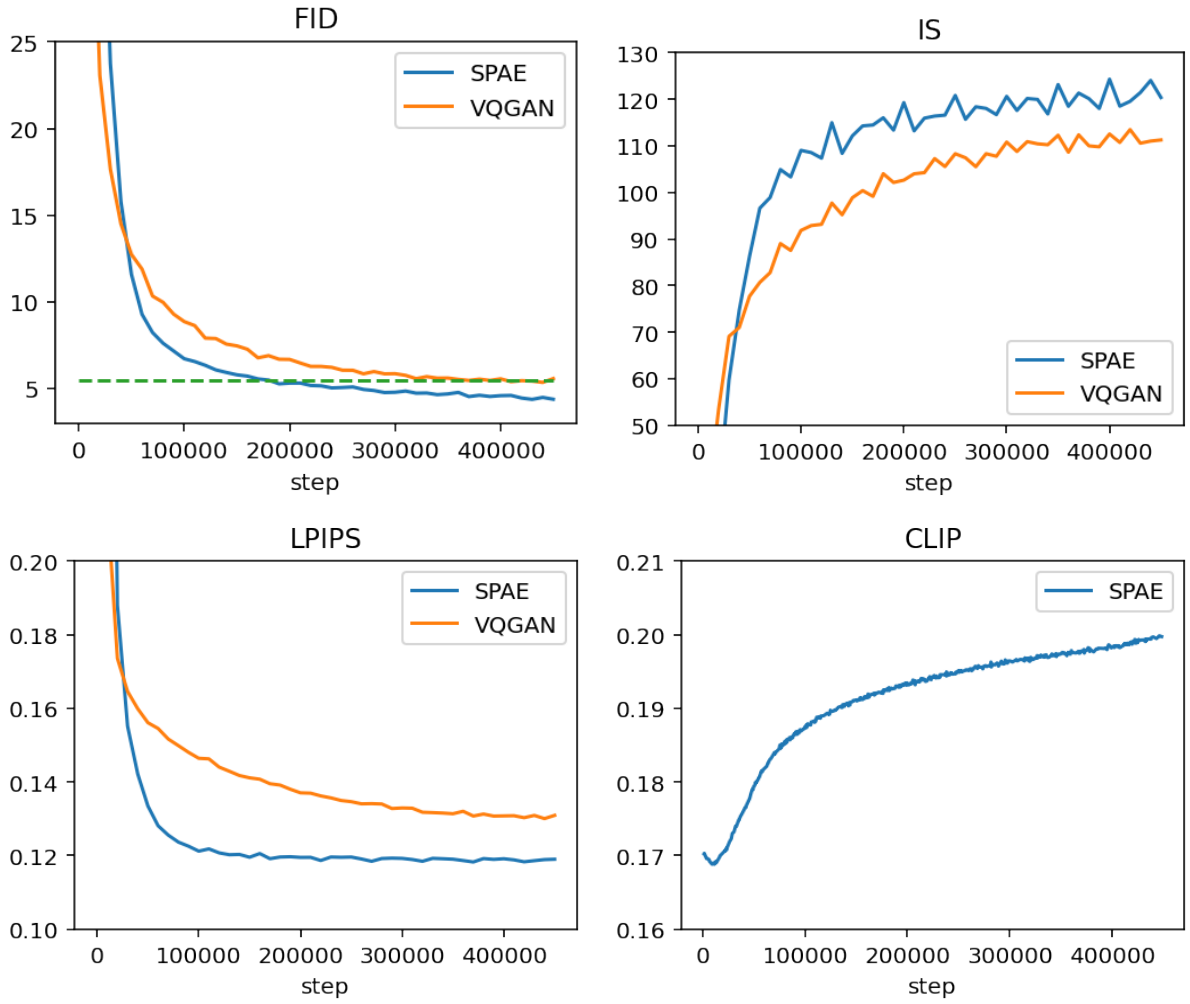


Figure 4.4: **Training curves of SPAE in comparison to VQGAN.**

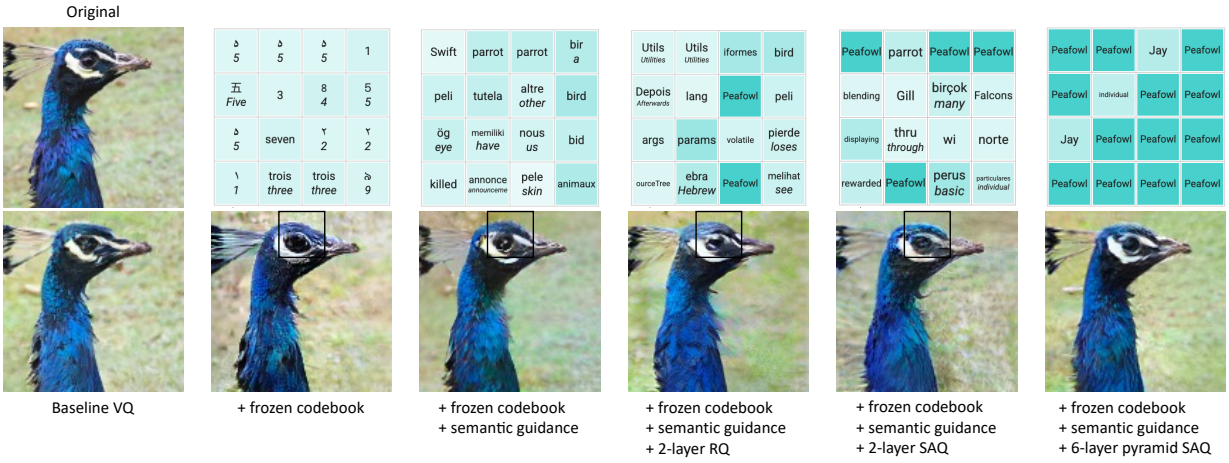


Figure 4.5: **Ablation examples with reconstructed image and semantic tokens** for models listed in Tab. 4.1. For non-pyramid tokens, we show a 4×4 crop from the first layer corresponding to the region indicated by the black box. For pyramid tokens, we use the third layer which consists of 4×4 tokens. We use darker cells to show tokens with higher CLIP similarity to the original image. For non-English sub-word tokens, we show automatic translation for reference in italic fonts below the original token.

Ablation studies. The results in Tab. 4.1 and Fig. 4.5 verify the effectiveness of the proposed designs in SPAE, as evaluated by reconstruction quality (FID, IS, LPIPS) and semantic relevance (CLIP score, few-shot classification accuracy). We have the following findings. First, simply using a frozen codebook negatively affects the reconstruction results, but with semantic guidance it performs comparably with the original VQGAN while producing meaningful lexical words. Second, RQ hurts reconstruction quality with a frozen codebook. This is different from RQ's standard setup [119] where the codebook is learned. Third, SAQ improves both quality and semantic similarity, where the pyramid enables representation with much fewer tokens. This allows for accommodating more examples within the fixed and constrained in-context length. Finally, per-layer semantic thresholds benefit understanding and the dynamic semantic loss weight helps reconstruction. The perceptual loss leverages a trained network with access to classification labels, but removing it results in a surprising improvement in classification accuracy while greatly hurting the reconstruction.

Token quality with more SPAE layers. The bottom section of Tab. 4.1 shows the per-layer reconstruction quality and semantic relevance of tokens from the SPAE-8 model in comparison to the default model. With more token layers, the model gains larger capacity for both semantic and appearance, where the appearance gets pushed into deeper layers. At layer 1 to 6, SPAE-8 yields consistently higher CLIP scores than SPAE. At the last three layers, SPAE-8 also has better reconstruction quality than the last two layers of SPAE. These results suggest the potential of better reconstruction quality and semantic relevance from using more token layers. Fig. 4.6 shows the coarse-to-fine image reconstruction by the last four layers of SPAE-8.

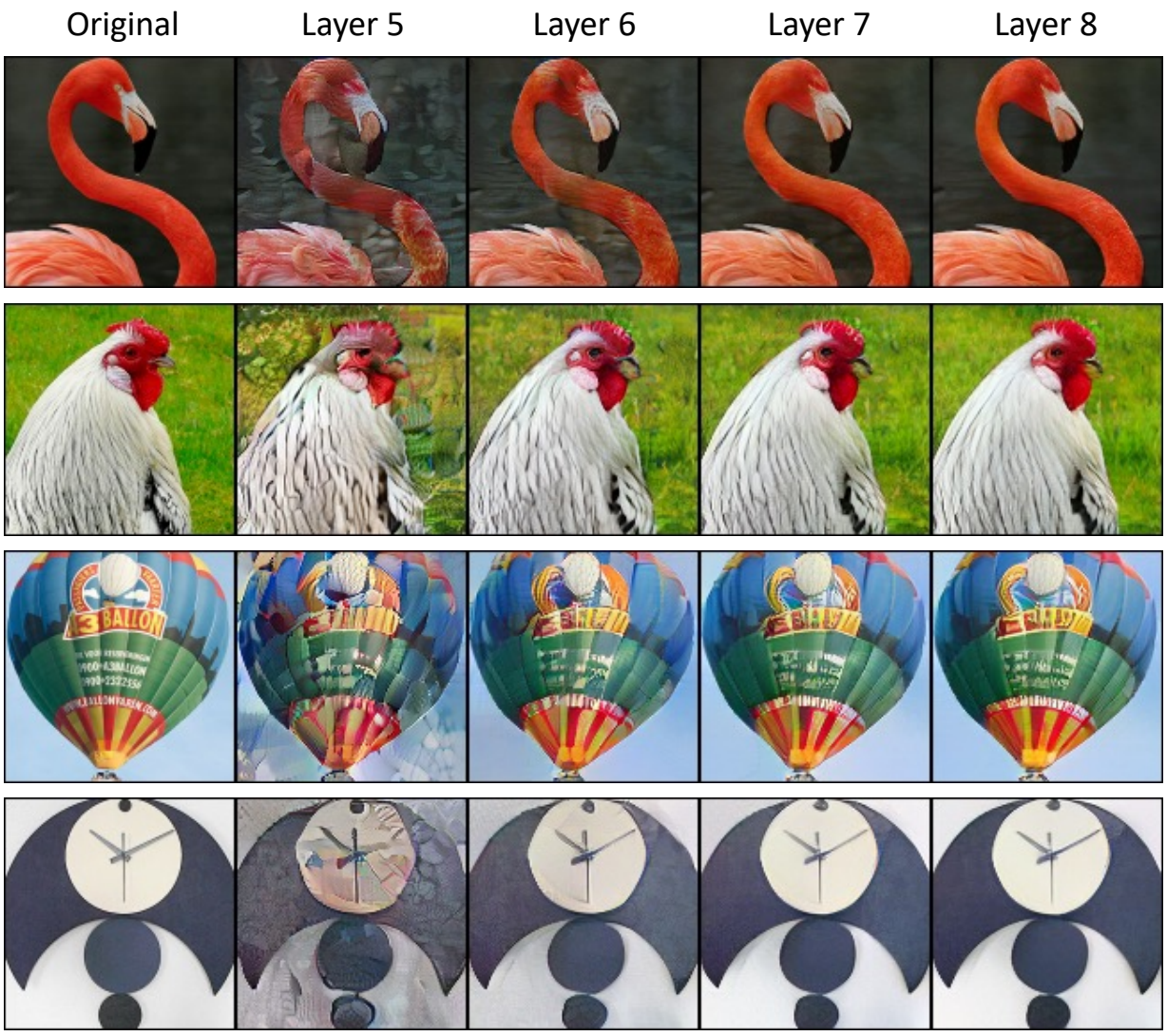


Figure 4.6: **Examples of coarse-to-fine image reconstruction** by SPAE-8. The top 5 layers reconstruct a noisy image. The appearance details gradually get refined as more token layers are aggregated by the streaming average quantization process.

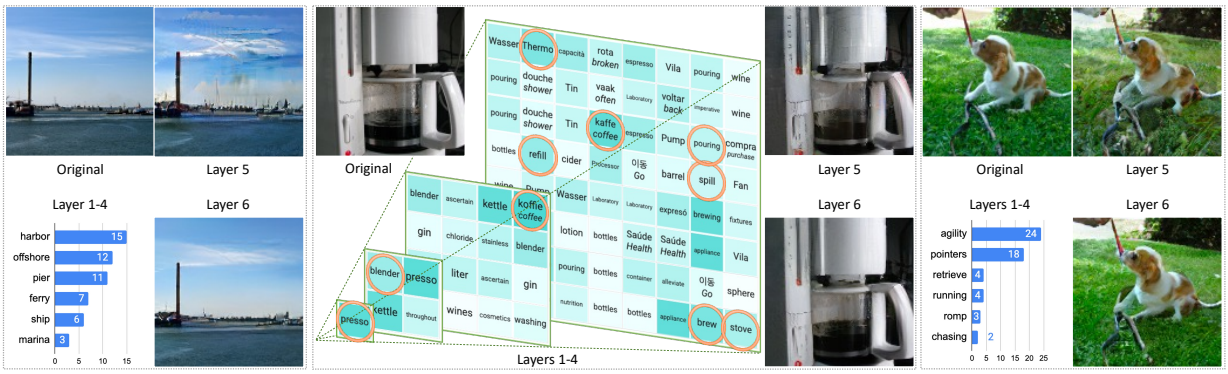


Figure 4.7: **Examples of pyramid image tokenization and reconstruction** by a 6-layer SPAE. We show the raw pyramid or histogram of most frequent tokens for the first four layers, and reconstructed images from layer 5 and 6. In the pyramid, we use darker cells to show tokens with higher CLIP similarity to the original image. For non-English sub-word tokens, we show automatic translation for reference in italic fonts below the original token. Circled tokens are mentioned in Section 7.4.1.

4.4.3 Qualitative Evaluations.

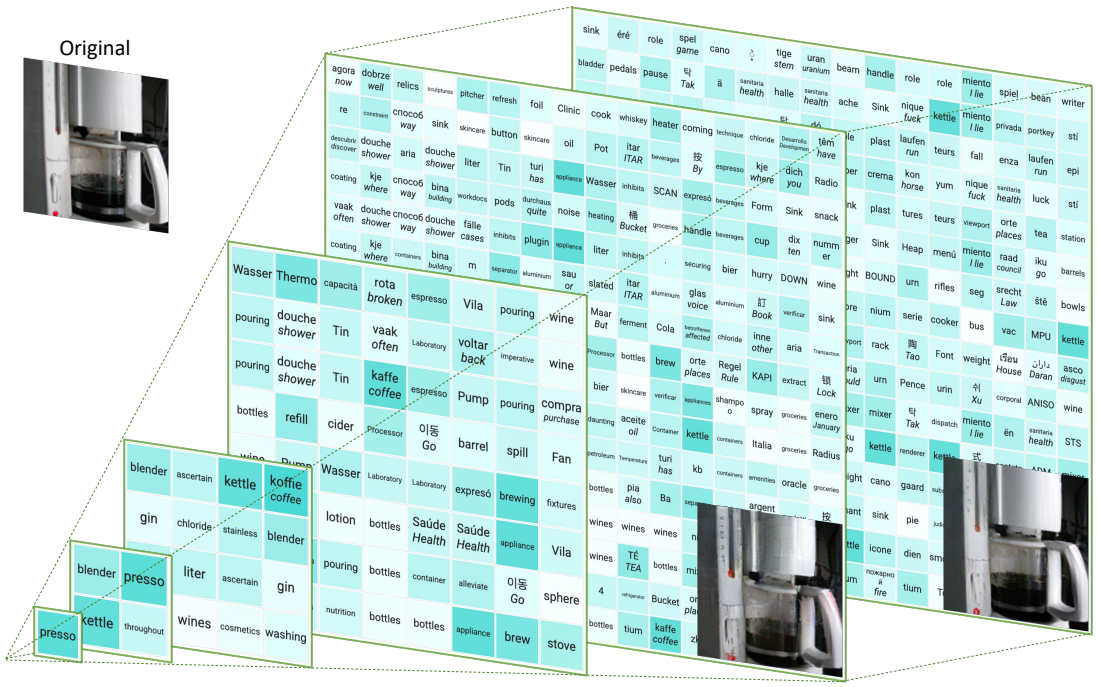
We visualize the tokens produced by SPAE in Figs. 4.7 and 4.8, where we show the raw pyramid or histogram of tokens with top frequencies for the first four layers, with reconstructed images from layer 5 and 6. We have the following findings.

First, the SPAE tokens are organized in a pyramid structure, with every layer comprising semantically related tokens to the image. The few tokens in the top layers seem to capture the primary theme of the image. For instance, in Fig. 4.7, the token `presso` (highlighted in orange) represents the espresso machine and other tokens like `blender` refer to related regions. Layer 3 and Layer 4 reveal additional details about localized objects. For example, the token `Thermo` refers to the thermometer in the top-left region, while `stove` appears in the bottom-right area. In addition to nouns, related verbs also show up, including `pouring`, `refill`, `spill`, and `brew`.

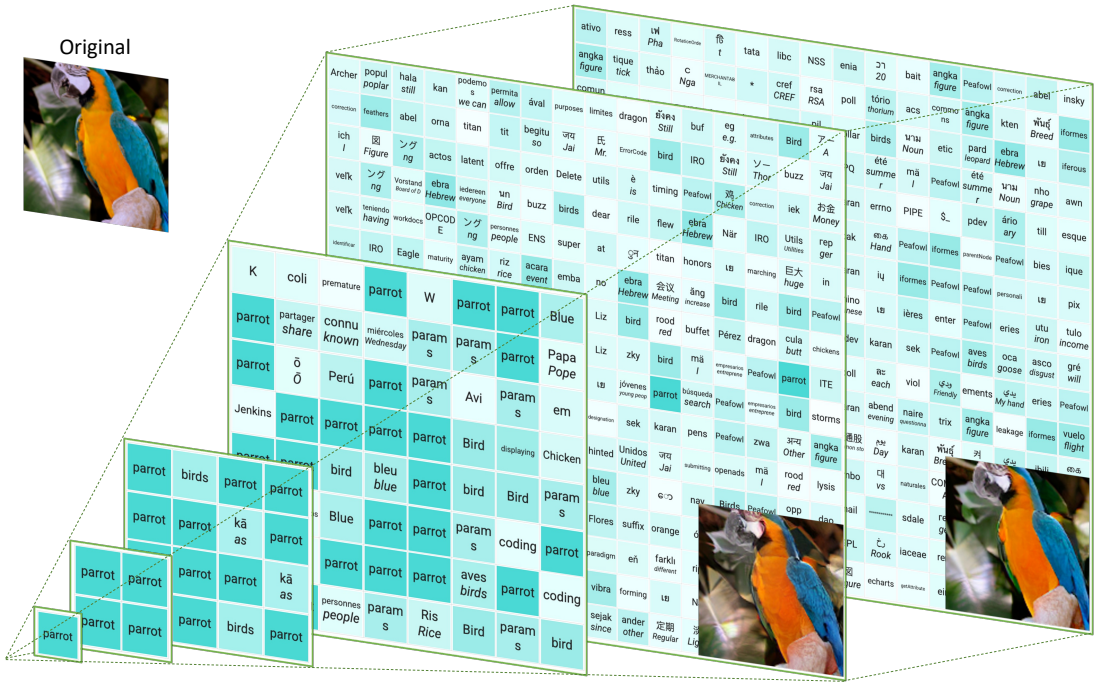
Second, it is worth noting that the CLIP model has an English-only vocabulary. However, thanks to the multilingual vocabularies and embeddings from the LLM, SPAE's semantic guidance is able to map to similar concepts in other languages, such as `koffie` in Dutch and `kaffe` in Danish as corresponding terms to the concept of coffee.

Third, similar to RQ tokens [119], SPAE tokens can reconstruct the image with progressively refined details when more layers, and thus tokens, are utilized. Fig. 4.7 shows Layer 5 begins to produce a reasonable reconstruction while Layer 6 further enhances the level of detail and smoothness.

Fig. 4.8 shows full tokenization and reconstruction samples by a 6-layer SPAE from ImageNet validation set. Key concepts are captured in the first few layers, whereas the later layers focus on the visual appearance. In the coffee machine example, many keywords are present to describe various aspects from the stove to the thermometer. In the parrot case, a single unified concept is repeatedly highlighted.



(a) Many keywords are present to describe various aspects from the stove to the thermometer.



(b) A single unified concept is repeatedly highlighted.

Figure 4.8: **Examples of pyramid image tokenization and reconstruction** by a 6-layer SPAE. For visualization purposes only, we use darker cells to show tokens with higher CLIP scores regarding the original image. For non-English sub-word tokens, we show automatic translation for reference in italic fonts below the original token. We show tokens in all six layers, along with reconstructed images from the last two layers.

4.5 Summary

In this chapter, we introduce the Semantic Pyramid AutoEncoder (SPAE) to map an image, or some other non-linguistic modality, to the token space of a frozen LLM. We use SPAE to produce a multi-scale *multi-modal* lexical *latent representation* for images and videos. The resulting tokens capture both the semantic meaning and the fine-grained details needed for visual reconstruction, effectively translating the visual content into a language comprehensible to the LLM. In Chapter 7, we will use SPAE to empower LLMs to perform a wide array of multimodal tasks.

Chapter 5

Scalable Visual Token Representation

Overview. While Large Language Models (LLMs) are the dominant models for generative tasks in language, they do not perform as well as diffusion models on image and video generation. To effectively use LLMs for visual generation, one crucial component is the visual tokenizer that maps pixel-space inputs to discrete tokens appropriate for LLM learning. In this chapter, we introduce MAGVIT-v2, a video tokenizer designed to generate concise and expressive tokens as *multi-modal latent representation* for both videos and images using a common token vocabulary. Equipped with this new tokenizer, we show that LLMs outperform diffusion models on standard image and video generation benchmarks including ImageNet and Kinetics. In addition, we demonstrate that our tokenizer surpasses the previously top-performing video tokenizer on two more tasks: (1) video compression comparable to the next-generation video codec (VVC) according to human evaluations, and (2) learning effective representations for action recognition tasks.

5.1 Motivation

Large transformer-based language models, commonly referred to as LMs or LLMs, are the de facto models for natural language generation [7, 145, 191]. Over time, LMs have expanded their capabilities to generate content in various modalities, asserting their dominance in other domains like audio [3], speech [167], code generation [128], medical applications [179] and robotics [262].

LMs are capable of generating images and videos. To do so, the image pixels are mapped into a sequence of discrete tokens by a visual tokenizer (*c.f.* Section 5.2). These tokens are then fed into the LM transformer, as if they were lexical words, for generative modeling. Despite notable advancements in employing LMs for visual generation [33, 57], LMs still do not perform as well as diffusion models [165]. For instance, when evaluating on the ImageNet dataset, a gold standard benchmark for image generation, the best language model [120] underperforms the diffusion model [62] by a substantial 48% margin (FID 3.41 *vs.* 1.79 when generating images at the 256×256 resolution).

Why do language models lag behind diffusion models in visual generation? This chapter suggests that a primary reason is the lack of a good visual representation, resembling our natural language system, for effectively modeling the visual world. To substantiate this hypothesis, this chapter shows that, when utilizing a good visual tokenizer, the masked language model [33, 49, 242] surpasses the state-of-the-art diffusion models in terms of both generation fidelity and efficiency across image and video benchmarks, given the same training data, comparable model size, and training budget. To the best of our knowledge, this provides the first evidence that language models beat diffusion models on the hallmark ImageNet benchmark.

It is worth emphasizing that our intention is not to assert whether the language model is superior to others, but to promote the exploration of visual tokenization methods for LLMs. A fundamental difference of LLMs from other models, such as diffusion models, is that LLMs utilize a discrete latent format: tokens obtained from a visual tokenizer. We show that the values of these discrete visual tokens should not be overlooked considering their distinct advantages as follows.

1. **Compatibility with LLMs.** The main advantage of a token representation is that it shares the same form as language tokens, making it straightforward to leverage the optimizations our community has developed over many years for LLMs. This includes faster training and inference speeds [122, 176], advancements in model infrastructure [45, 54], learning recipes for model scaling [25, 40], and GPU/TPU optimization, among other innovations. Unifying vision and language by the same token space could set the stage for a true multimodal LLM that can understand, generate, and reason within our visual environment.
2. **Compressed representation.** The discrete token may offer a fresh perspective on video compression. The visual tokens can serve as a new video compression format to reduce disk storage and bandwidth during internet transfers. Unlike compressed RGB pixels, these tokens can be fed directly into generative models, bypassing the conventional decompression and latent encoding steps. This allows for faster processing in generative video applications, especially beneficial in edge computing cases.

3. **Visual understanding benefits**. Prior research has shown that the discrete tokens are valuable as a pre-training target in self-supervised representation learning, as discussed in BEiT [14] and BEVT [211]. Additionally, research finds that using tokens as the model inputs improves the robustness and generalization [137].

In this chapter, we introduce MAGVIT-v2, a video tokenizer designed to map videos (and images) into compact discrete tokens. Our model is built on the state-of-the-art video tokenizer, MAGVIT [242], within the VQ-VAE framework [200]. We propose two new techniques. First, a novel lookup-free quantization method enables the learning of a large vocabulary that is able to improve generation quality of the language model. Second, through extensive empirical analyses, we have identified modifications to the tokenizer that not only enhance generation quality but also enable the tokenization of both images and videos using a shared vocabulary.

We empirically demonstrate that our model outperforms the previously top-performing video tokenizer, MAGVIT, in three key areas. First, our model significantly improves the generation quality of MAGVIT, establishing the state of the art on the common image and video benchmarks. Second, user studies indicate that its compression quality exceeds that of MAGVIT and the current video compression standard, HEVC [186]. Moreover, it is on par with the next-generation video codec, VVC [24]. Finally, we show that, compared to MAGVIT, our new tokens are stronger for video understanding tasks across two setups and three datasets. The main contributions of this chapter are:

- A new video tokenizer that outperforms the previously best-performing video tokenizer in three areas: visual generation, video compression, and action recognition.
- A novel lookup-free quantization approach that enables improving the visual generation quality of language models by learning a large vocabulary.
- To the best of our knowledge, the first evidence suggesting that a language model can outperform diffusion models on ImageNet when provided with the same training data, an equivalent model size, and a similar training budget.
- A video compressor with better quality than HEVC and VVC, at similar bit rates, according to user studies. To our knowledge, this is the first successful attempt of a visual tokenizer designed for video generation to achieve comparable results to standard codecs.

5.2 Background

Language Model (LM) for visual generation. LMs have been extended to generate images and videos. A visual tokenizer f is used to first map visual inputs into a sequence of discrete tokens. A video $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$ (or image when $T = 1$) is tokenized into a discrete representation $\mathbf{X} = f(\mathbf{V}) \in \{1, 2, \cdots, K\}^{T'\times H'\times W'}$, where K is the codebook (vocabulary) size of the visual tokenizer. $\mathbf{X}$ is flattened into a 1D token sequence obtained using raster scan ordering and then fed into an LM transformer for generative modeling.

Two types of LMs are commonly used for visual generation. The *Autoregressive LM (AR-LM)* includes ImageGPT [37], DALL-E [160], Parti [235], *etc.* An AR-LM predicts the next token given the previous tokens along with additional conditioning information $\mathbf{c}$ using a categorical distribution for $p_\theta(\mathrm{x}_i \mid \mathbf{x}_{<i}; \mathbf{c})$. During inference, AR-LMs use the standard autoregressive de-

coding over the tokens. Finally, the tokens are converted back to pixels by a decoder associated with the visual tokenizer.

The *Masked LM (MLM)* is another type of language model for visual generation, such as: MaskGIT [33], MAGVIT [242], Phenaki [203], and MUSE [34], among others. An MLM is trained using a masked token objective [49], where some tokens in the sequence are randomly masked and need to be predicted given the observed tokens. Let $\mathbf{m} \in \{0,1\}^n$ be a random binary sequence where $\mathbf{m}^\top \mathbf{1} \in [0, n-1]$. The MLM learns $p_\theta(\mathrm{x}_i \mid \{\mathrm{x}_j : \mathrm{m}_j = 1, \forall j\}; \mathbf{c})$ for all i where $\mathbf{m}_i = 0$. To generate a video or image during inference, the MLM uses the non-autoregressive decoding algorithms for images and videos [33, 242]. The decoding starts with a fully masked sequence, which is iteratively filled by repeating two steps: (1) sample the whole sequence $\hat{\mathbf{x}}^{(t)}$ from p_θ given the non-masked tokens from the previous step, (2) re-mask the $\lfloor \lambda(t) \cdot n \rfloor$ tokens in $\hat{\mathbf{x}}^{(t)}$ with the lowest probability, following a decreasing masking ratio schedule $\lambda(t)$, according to timestamp t.

Denoising Diffusion Models (DDM). DDMs [181, 183] are regarded as the state-of-the-art in visual generation due to their high-quality image [50, 50, 88, 169] and video generation [88, 90, 177]. For instance, DDPM [87] learns a denoising process parameterized as conditional Gaussian distributions over image pixels. Recently, diffusion models and language models have displayed a significant overlap, where DDMs diffuse over latents rather than raw pixels. These latents are obtained using models similar to the visual tokenizer used by LMs. In fact, the very first latent in diffusion, proposed by [165], is derived from a visual tokenizer. Binary latents for image modeling are also used in [216], where the diffusion process is parameterized with Bernoulli distributions. Later studies have identified advantages in substituting the U-Net [166] denoising backbone with a Transformer [102, 147] or a hybrid of both [94], making the distinctions between diffusion and language models in visual generation more blurred. Yet, a fundamental difference between DDMs and LMs lies in the latent format, *i.e.*, continuous *vs.* discrete. We have discussed the benefits of having discrete tokens in Section 5.1 and will show that the proposed tokenizer improves in these aspects.

Visual tokenization. Visual tokenization plays an essential role in mapping pixels into a discrete representation suitable for generative modeling. VQ-VAE [200] is a cornerstone work in image tokenization. A VQ-VAE model consists of a convolutional neural network (CNN) encoder, a vector-quantization (VQ) bottleneck, and a CNN decoder. Given a video $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$, the VQ-VAE's encoder E produces latent embeddings $\mathbf{Z} = E(\mathbf{V}) \in \mathbb{R}^{T' \times H' \times W' \times d}$. Each embedding vector $\mathbf{z} \in \mathbb{R}^d$ in $\mathbf{Z}$ is then passed through the vector quantizer q, which assigns it to the closest entry $\mathbf{c} \in \mathbb{R}^d$ in the learned codebook embedding $\mathbf{C} \in \mathbb{R}^{K \times d}$:

$$q(\mathbf{z}) = \mathbf{c}_i, \text{ where } i = \underset{j \in \{1,2,\cdots,K\}}{\arg\min} \|\mathbf{z} - \mathbf{c}_j\|_2. \tag{5.1}$$

To get discrete tokens, we drop the embedding dimension and represent $\mathbf{Z}$ by its indices $\mathbf{X} \in \{1, 2, \cdots, K\}^{T' \times H' \times W'}$. For decoding, embeddings of all image tokens are given as input to the decoder D to reconstruct the input $\hat{\mathbf{V}} = D(\mathbf{Z})$. Following VQ-VAE, VQGAN [57] introduces an adversarial loss and feature-level perceptual losses to enhance the image quality.

Video tokenization is more challenging since it requires modeling the visual dynamics within the compressed spatial-temporal latent space. Early studies on video tokenization treat frames as independent images with no temporal compression [74, 225]. Later research [63, 232, 242] integrates 3D CNNs to tokenize spatial-temporal volumes. The state of the art in video tokenization is MAGVIT [242], which introduces a better 3D architecture, an inflation technique for initialization using image pre-training, and robust training losses. With MAGVIT, the LMs achieve leading generation quality across multiple video benchmarks. However, MAGVIT struggles to tokenize images and often results in noticeable flickering in longer videos. Beyond the CNN-based models discussed above, additional models have been proposed. ViT-VQGAN [234] introduces transformer blocks as a substitute for CNNs for image tokenization. C-ViViT [203] further extends this idea for video tokenization. Despite these advances in vector quantization (VQ), the codebook learned by previous VQ models is relatively small (*e.g.*, 8k) due to the difficulty in improving the generation quality with larger vocabularies. In contrast, our tokenizer can induce a large vocabulary (*e.g.*, 262k) that can be effectively modeled by an LM, leading to enhanced image and video generation quality.

Text-to-{image, video}. Text-to-image and text-to-video generation has garnered significant rapid advancements using both language models [34, 246] and diffusion models [16, 64, 88, 161, 177]. Although diffusion models, such as Midjourney, are considered the top performers in these tasks, it is unclear whether their advantage stems from the model, data, or some other unidentified factors. Indeed, it is challenging to scientifically compare these text-to-image models as they are trained on varied datasets, with some even being proprietary data, under inconsistent training conditions. To facilitate a fairer comparison, this chapter prioritizes using the ImageNet and Kinetics benchmarks.

5.3 MAGVIT-v2 Video Tokenizer

We introduce a new *video tokenizer* designed to map the spatial-temporal dynamics from a visual scene into compact discrete tokens suitable for language models. Compared with image generation, video generation still faces substantial challenges in generating consistent and realistic motion. We are interested in exploring the capabilities of language models in tackling this unsolved challenge. Therefore, we focus on a video tokenizer that can effectively represent video for generative modeling. Our approach builds upon the state-of-the-art video tokenizer, MAGVIT, as detailed in [242]. This section highlights two new designs: a lookup-free quantizer and a collection of enhancements to the tokenizer model.

5.3.1 Lookup-Free Quantizer

Although the community has made great progress in developing VQ-VAEs, the relationship between improvements in the reconstruction quality and subsequent generation quality is still not well understood. A common misconception is that improving reconstruction equates to improving the generation of the language model. For example, enlarging the vocabulary can improve reconstruction quality. However, such improvement only extends to generation when

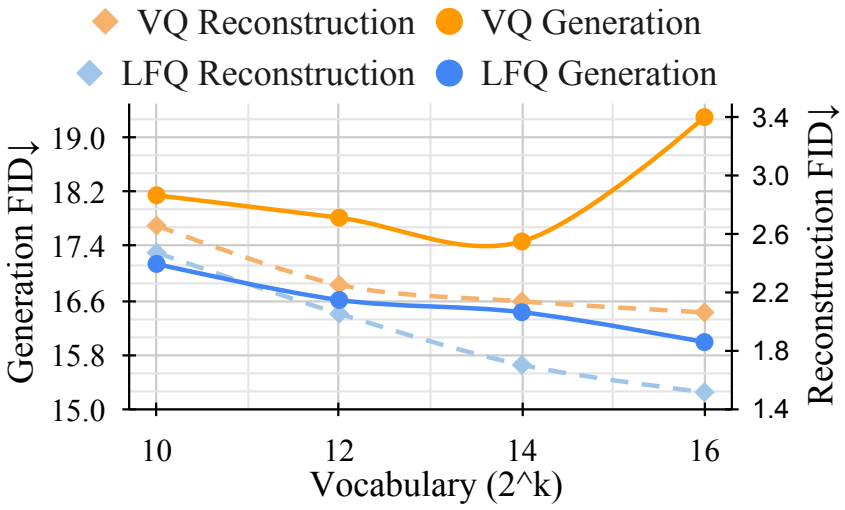


Figure 5.1: **Reconstruction and generation quality curves** in FID on ImageNet when scaling the tokenizer's vocabulary size with Vector Quantization (VQ) and Lookup-Free Quantization (LFQ). Comparison is done at 128×128 resolution using an MLM with 306-372M parameters.

the vocabulary size is small, and a very large vocabulary can actually hurt the performance of the language model.

As illustrated by the dashed curves in Fig. 5.1, the reconstruction FID, indicated by the right y-axis (where a lower value is better), improves as the vocabulary size (the x-axis) increases. The orange solid curve in Fig. 5.1 represents the LM's generation quality (the left y-axis). The generation FID initially improves but deteriorates for larger vocabulary. This may shed light on why the vocabulary size of most language models for visual generation is around 1-8k [57, 203], which is significantly smaller than the size of natural language vocabulary, *i.e.* over 200k.

A simple trick for training a larger codebook involves decreasing the code embedding dimension when increasing the vocabulary size [234]. This trick captures the intuition of limiting the representational capacity of individual tokens, which in turn facilitates learning over the distribution of a large vocabulary.

Lookup-Free Quantization (LFQ). Motivated by the above observation, we reduce the VQ-VAE codebook's embedding dimension to zero. Formally, the codebook $\mathbf{C} \in \mathbb{R}^{K \times d}$ is replaced with an integer set $\mathbb{C}$ where $|\mathbb{C}| = K$. Recall that in VQ-VAE models, the quantizer must look up all K d-dimensional embeddings in the codebook, where d is typically 256, when computing the closest codebook entry to the encoder output. This new design eliminates the need for such embedding lookup entirely hence we call it *lookup-free quantization (LFQ)*. We found that LFQ can grow the vocabulary size in a way benefiting the generation quality of language models. As shown by the blue curves in Fig. 5.1, both reconstruction and generation consistently improves as the vocabulary size increases – a property not observed in current VQ-VAE methods.

While various LFQ methods are available, this chapter discusses a straightforward variant that assumes independent codebook dimensions and binary latents. Specifically, the latent space of LFQ is decomposed as the Cartesian product of single-dimensional variables, as

$\mathbb{C} = \bigtimes_{i=1}^{\log_2 K} C_i$. Given a feature vector $\mathbf{z} \in \mathbb{R}^{\log_2 K}$, each dimension of the quantized representation $q(\mathbf{z})$ is obtained from:

$$q(\mathrm{z}_i) = C_{i,j}, \text{ where } j = \arg\min_k \|\mathrm{z}_i - C_{i,k}\|, \tag{5.2}$$

where $C_{i,j}$ is the j-th value in C_i. With $C_i = \{-1, 1\}$, the arg min can be computed by the sign function as

$$q(\mathrm{z}_i) = \mathrm{sign}(\mathrm{z}_i) = -\mathbf{1}\{\mathrm{z}_i \leq 0\} + \mathbf{1}\{\mathrm{z}_i > 0\}. \tag{5.3}$$

With LFQ, the token index for $q(\mathbf{z})$ is given by:

$$Index(\mathbf{z}) = \sum_{i=1}^{\log_2 K} \arg\min_k \|\mathrm{z}_i - C_{i,k}\| \prod_{b=0}^{i-1} |C_b| = \sum_{i=1}^{\log_2 K} 2^{i-1}\mathbf{1}\{\mathrm{z}_i > 0\}, \tag{5.4}$$

where $|C_0| = 1$ sets the virtual basis.

We add an entropy penalty during training to encourage codebook utilization:

$$\mathcal{L}_{entropy} = \mathbb{E}[H(q(\mathbf{z}))] - H[\mathbb{E}(q(\mathbf{z}))]. \tag{5.5}$$

This penalty is inspired by a similar loss used in image VQGAN model [33], which is also found in entropy-based clustering [103]. In LFQ, given the independence among dimensions, we rewrite $H(q(\mathbf{z})) = \sum_{i=1}^{\log_2 K} H(q(\mathrm{z}_i))$. The $H[\mathbb{E}(q(\mathbf{z}))]$ term can be approximated with sub-groups of dimensions for $K > 2^{18}$ where direct estimation is memory bound.

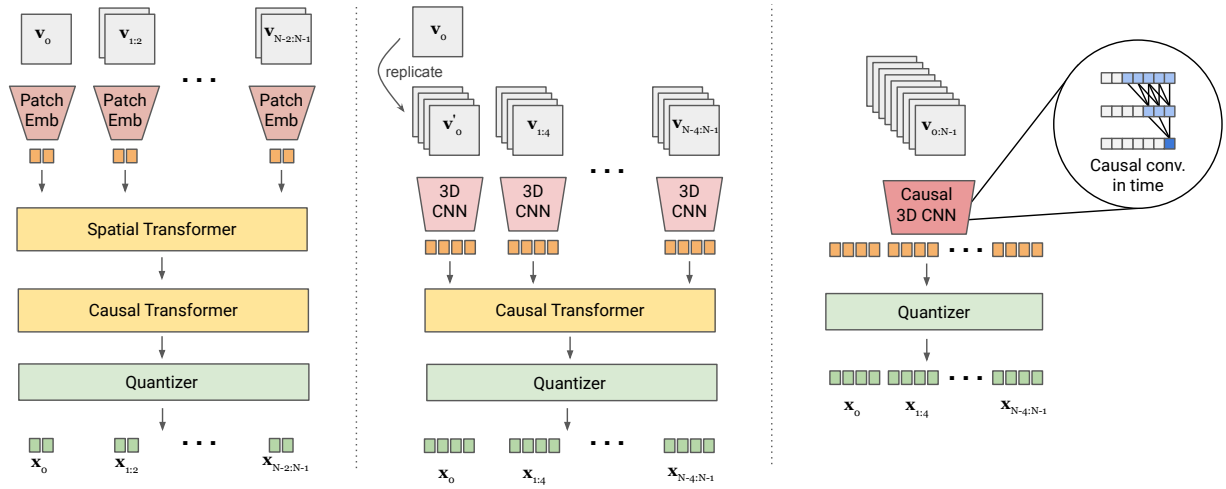
We note that there are various other variants of LFQ, *e.g.*, opting for the multivariant over the binary codebook C_i or employing other quantization techniques such as [4]. As the first to introduce this concept, we focus on the simplest form with independent binary dimensions, which shows promising improvements. Other LFQ methods merit further research.

In addition to the entropy penalty (Eq. (5.5)), an LFQ-based tokenizer is trained using the standard combination of *reconstruction*, *GAN*, *perceptual*, and *commitment* losses [57], excluding the inapplicable codebook loss. Following [242], we use LeCAM regularization [194] for improved stability.

5.3.2 Visual Tokenizer Model Improvement

Joint image-video tokenization. A desirable feature of visual tokenization is the capability to tokenize images and videos using a shared codebook. However, the MAGVIT tokenizer, which utilizes the 3D CNN, faces challenges in tokenizing images due to the temporal receptive field.

To build a joint image-video tokenizer, a new design is needed. We begin our discussion by revisiting an existing method C-ViViT [203]. As depicted in Fig. 5.2a, C-ViViT employs full spatial transformer blocks combined with causal temporal transformer blocks. This approach performs reasonably well but has two drawbacks. First, unlike CNNs, the positional embeddings makes it difficult to tokenize spatial resolutions that were not seen during training. Second, empirically we found that 3D CNNs perform better than spatial transformer and produce tokens with better spatial causality of the corresponding patch.



(a) C-ViViT (b) C-ViViT + MAGVIT (c) Causal 3D CNN

Figure 5.2: **Causal tokenizer architecture comparison**. The decoders, which are omitted from the figure, employ an architecture that is symmetric to the encoder.

To tackle these drawbacks, we explore two plausible designs. Fig. 5.2b combines C-ViViT and MAGVIT. Assuming a temporal compression ratio of 4, a 3D CNN processes blocks of 4 frames followed by a causal transformer. In Fig. 5.2c, we use the temporally causal 3D convolution to replace the regular 3D CNN. Specifically, the temporal padding scheme for a regular 3D convolution layer with kernel size (k_t, k_h, k_w) includes $\lfloor \frac{k_t - 1}{2} \rfloor$ frames before and $\lfloor \frac{k_t}{2} \rfloor$ frames after the input frames. In contrast, a causal 3D convolution layer pads with $k_t - 1$ frames before the input and nothing after, so that the output for each frame only depends on the previous frames. In consequence, the first frame is always independent of other frames, allowing the model to tokenize single images.

Temporal convolutional subsampling with stride s is sufficient for $s\times$ down-sampling by mapping $1 + s \times t$ frames into $1 + t$. After a regular $s\times$ up-sampling, we drop the first $s - 1$ resulting frames, which maps $1 + t$ frames into $1 + s \times t$ and allows for the tokenization of a single image. Tab. 5.7a empirically compares the designs in Fig. 5.2, and we find that the causal 3D CNN performs the best.

Architecture modifications. In addition to using causal 3D CNN layers, we made several other architectural modifications to improve upon the MAGVIT model. First, we change the encoder downsamplers from average pooling into strided convolutions to leverage learned kernels, and replace the decoder upsamplers from nearest resizing followed by convolution with a depth-to-space operator. Second, we defer the temporal downsampling from the first few encoder blocks to the last ones. In addition, the downsampling layer in the discriminator now utilizes 3D blur pooling [253] to encourage shift invariance. Finally, we add one adaptive group normalization layer before the residual blocks at each resolution in the decoder to pass in the quantized latents as the control signal following StyleGAN [108]. Tabs. 5.7b and 5.7c empirically verify these designs.

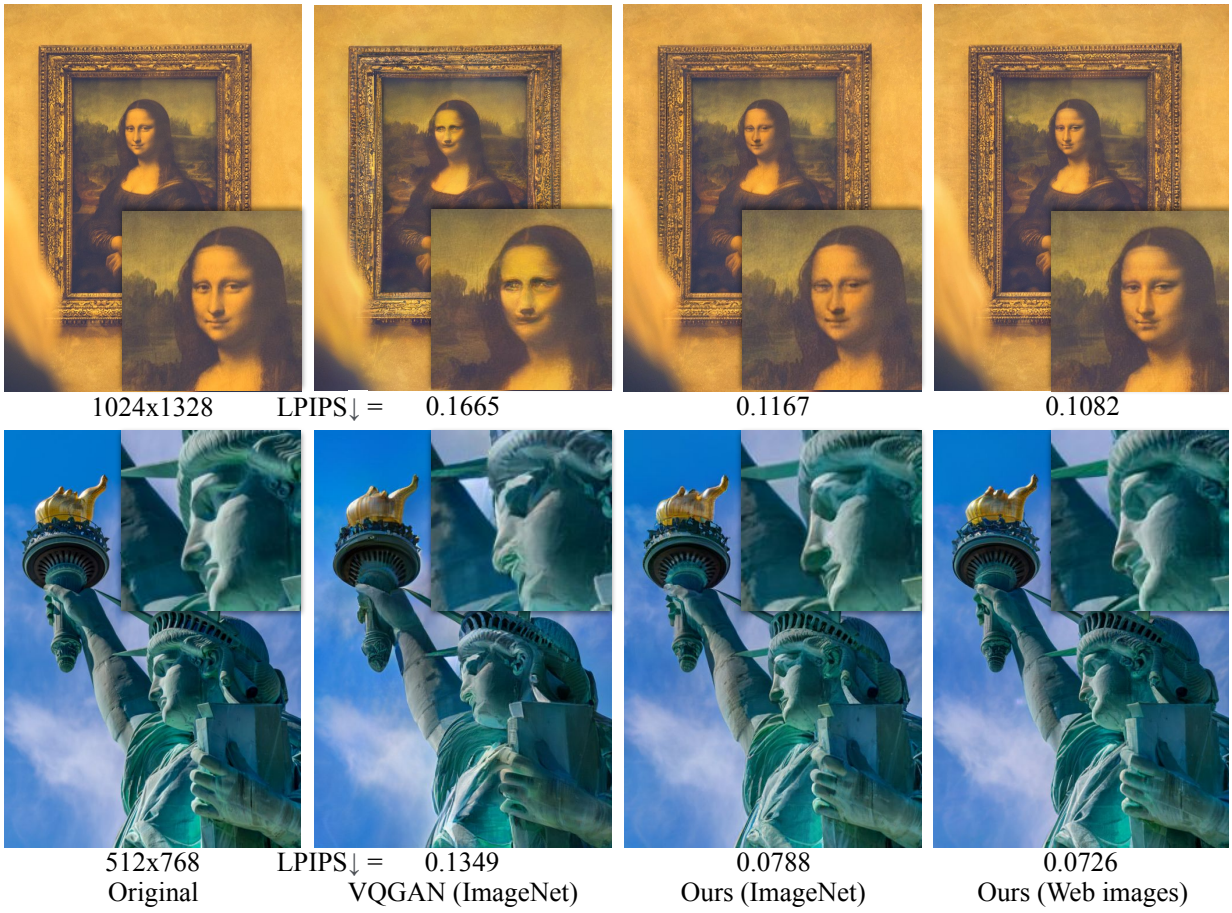


Figure 5.3: **Image reconstruction samples with different tokenizers**. We compare the VQ-GAN used in MaskGIT [33] with two of our models trained on ImageNet and web images [39]. Original images are by Eric TERRADE and Barth Bailey on Unsplash.

Token factorization for efficient prediction. The output tokens can be fed into language models to generate videos. To assist smaller transformers predicting in a large vocabulary, we can factorize the LFQ token's latent space into equal subspaces. For instance, rather than predicting using a codebook of size 2^{18}, we can predict in two concatenated codebooks, each of size 2^9. We embed each subspace token separately and use their embedding summation as the token embedding for the transformer input. We find it beneficial to use weight tying [151], a common technique in language modeling, which involves sharing the weights between the embedding and softmax layers. For the output layer with a factorized vocabulary, we use the embedding matrix for each subspace to obtain logits with seperate prediction heads.

5.4 Experimental Results

This section empirically verifies the proposed tokenizer across three distinct tasks: video and image generation, video compression, and action recognition. Fig. 5.3 visually compares the

reconstruction quality of our tokenizer with prior works. More qualitative samples are shown at https://magvit.cs.cmu.edu/v2.

5.4.1 Experimental Setups

Datasets. We use Kinetics-600 (K600) [30] and UCF-101 [184] for video generation experiments, along with ImageNet [47] for image generaton. In addition, MCL-JCV [208] is used as the testbed for video compression, with Kinetics-400 (K400) [109] and SSv2 [68] for video understanding.

Implementation details. We follow the tokenizer training setting and hyperparameters in [242], unless stated otherwise. LFQ is used, which eliminates the codebook embedding, to increase the default codebook size to $K = 2^{18}$. The weight of $\mathcal{L}_{entropy}$ follows an annealing schedule with a 3× higher starting point and linearly decays to a fixed value of 0.1 within 2k steps.

Fig. 5.4 illustrates the architecture of our proposed MAGVIT-v2. We provide detailed training hyperparameters for our models as listed below:

- Video input: 17 frames, frame stride 1, 128×128 resolution.
- Base channels: 128.
- VQVAE channel multipliers: $1, 2, 2, 4$.
- Discriminator channel multipliers: $2, 4, 4, 4, 4$.
- Number of residual blocks: 4.
- Latent shape: $5 \times 16 \times 16$.
- Vocabulary size: 2^{18}.
- Initialization: central inflation from a 2D model trained on ImageNet with this setup.
- Entropy loss weight: 0.1.
- Entropy loss annealing steps: 2000.
- Entropy loss annealing factor: 3.
- Reconstruction loss weight: 5.0.
- Generator loss type: Non-saturating.
- Generator adversarial loss weight: 0.1.
- Discriminator gradient penalty: r1 with cost 10.
- Perceptual loss weight: 0.1.
- Commitment loss weight: 0.25.
- LeCAM weight: 0.001.
- Peak learning rate: 10^{-4}.
- Learning rate schedule: linear warm up and cosine decay.
- Optimizer: Adam with $\beta_1 = 0$ and $\beta_2 = 0.99$.

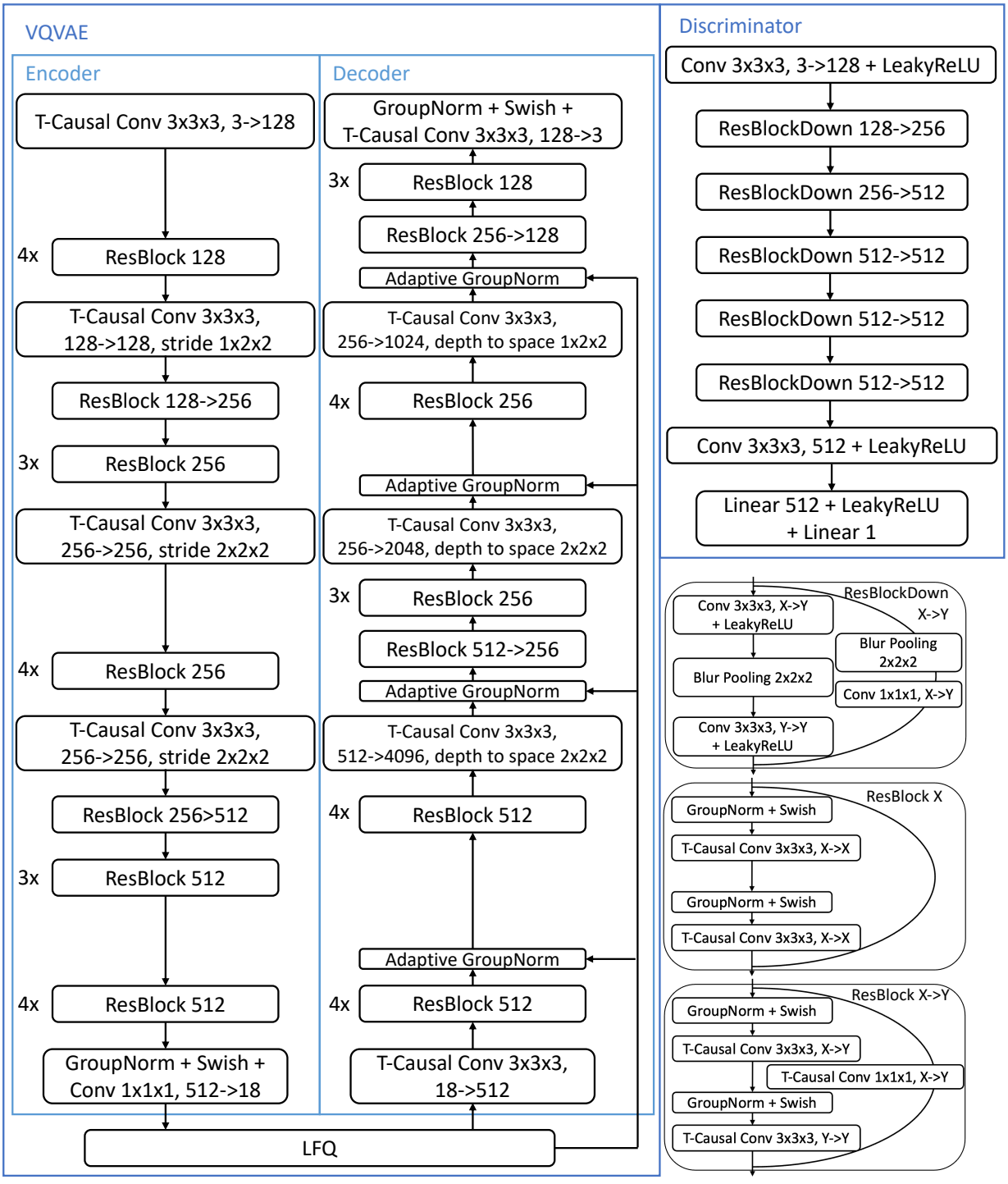


Figure 5.4: **MAGVIT-v2 tokenizer architecture**. T-Causal Conv refers to temporally causal convolution.

Type	Method	K600 FVD↓	UCF FVD↓	#Params	#Steps
GAN	TrIVD-GAN-FP [136]	$25.7_{\pm 0.7}$			1
Diffusion	Video Diffusion [90]	$16.2_{\pm 0.3}$		1.1B	256
Diffusion	RIN [102]	10.8		411M	1000
AR-LM + VQ	TATS [63]		$332_{\pm 18}$	321M	1024
MLM + VQ	Phenaki [203]	$36.4_{\pm 0.2}$		227M	48
MLM + VQ	MAGVIT [242]	$9.9_{\pm 0.3}$	$76_{\pm 2}$	306M	12
MLM + LFQ	non-causal baseline	$11.6_{\pm 0.6}$		307M	12
MLM + LFQ	*MAGVIT-v2 (ours)*	$5.2_{\pm 0.2}$		307M	12
		4.3$_{\pm 0.1}$	**58**$_{\pm 3}$		24

Table 5.1: **Video generation results**: frame prediction on Kinetics-600 and class-conditional generation on UCF-101. We adopt the evaluation protocol of MAGVIT.

- EMA model decay rate: 0.999.
- Batch size: 256.

5.4.2 Visual Generation

The masked language model (MLM) [49] is used in image and video generation. To verify the tokenizer, we employ the same MLM transformers in MAGVIT [242]. We select the MLM due to its competitive performance on benchmark datasets [126, 242]. As we use a smaller MLM (~300M parameters) with a large codebook (2^{18} ≈262K), the token factorization as discussed in Section 5.3.2 is applied using two heads with each predicting from a codebook of size 2^9.

Video generation. We consider two standard video benchmarks, UCF-101 for class-conditional generation and K600 for frame prediction with 5-frame condition. FVD [198] is used as our primary evaluation metric.

We inflate an image tokenizer trained at 128×128 for video modeling. Different from the inflation in [242], we fill in the temporally last slice to correspond to the causal padding scheme. In addition, we disable the inflation for the discriminator and train it from scratch for better stability. We train the causal video tokenizer on Kinetics-600 training set for 190 epochs with batch size 256. This tokenizer is also used in subsequent evaluations of video compression and action recognition.

With the causal tokenizer producing 5×16×16 tokens for a 17×128×128 clip, the first 2×16×16 tokens are provided as the condition of the first 5 frames, per the standard setup of Kinetics-600 frame prediction benchmark. We train the MLM transformer following [242] with token factorization for 360 epochs with batch size 256. The model is sampled with a cosine schedule using temperature 32.

Tab. 5.1 shows that our model surpasses all prior arts in both benchmarks. Specifically, it outperforms the previous best model MAGVIT by a large margin, while using the same MLM transformer backbone. In addition, it significantly outperforms the non-causal baseline on

Figure 5.5: **Frame prediction samples on Kinetics-600**.

Type	Method	w/o guidance FID↓	w/o guidance IS↑	w/ guidance FID↓	w/ guidance IS↑	#Params	#Steps
GAN	StyleGAN-XL [174]			2.41	267.8	168M	1
Diff. + VAE*	DiT-XL/2 [147]	12.03	105.3	3.04	240.8	675M	250
Diffusion	ADM+Upsample [50]	9.96	121.8	3.85	221.7	731M	2000
Diffusion	RIN [102]	3.95	216.0			320M	1000
Diffusion	simple diffusion [94]	3.54	205.3	3.02	248.7	2B	512
Diffusion	VDM++ [111]	2.99	232.2	2.65	278.1	2B	512
MLM + VQ	MaskGIT [33]	7.32	156.0			227M	12
MLM + VQ	DPC+Upsample [126]	3.62	249.4			619M	72
MLM + LFQ	*MAGVIT-v2 (ours)*	4.61	192.4			307M	12
		3.07	213.1	**1.91**	**324.3**		64

Table 5.2: **Image generation results**: class-conditional generation on ImageNet 512×512. Guidance indicates the classifier-free diffusion guidance [86]. * indicates usage of extra training data. We adopt the evaluation protocol and implementation of ADM.

frame prediction, highlighting the contribution of the causal tokenizer. These results demonstrate the essential role of a good visual tokenizer in enabling LMs to generate high-quality videos. Fig. 5.5 shows qualitative samples from the model.

Image generation on ImageNet. We evaluate MAGVIT-v2 on image generation under the standard ImageNet class-conditional setting. We present results for resolution 512×512 and 256×256. FID [85] and Inception Score (IS) [172] are used as evaluation metrics.

We set up two image tokenizers to downsample by 16× and 32×, where they are used for generation at 256×256 and 512×512, respectively. In both cases, an image is represented as 16×16 tokens. We train them on the ImageNet training set for 270 epochs using a batch size of 256, both with 256×256 images. With this tokenizer we train a Masked Language Model following [242], using the token factorization described in Section 5.3.2. We train for 1080 epochs in accordance with the prior best model MDT [62], with batch size 1024 for better efficiency. For preprocessing and data augmentation, we randomly crop 80-100% of an image while keeping the aspect ratio, followed by random horizontal flipping. The class label is dropped for 10% of

Type	Method	w/o guidance FID↓	w/o guidance IS↑	w/ guidance FID↓	w/ guidance IS↑	# Params	Steps
GAN	BigGAN-deep [21]	6.95	171.4			160M	1
GAN	StyleGAN-XL [174]			2.30	265.1	166M	1
Diff. + VAE*	LDM-4 [165]	10.56	103.5	3.60	247.7	400M	250
Diff. + VAE*	DiT-XL/2 [147]	9.62	121.5	2.27	278.2	675M	250
Diff. + BAE	Binary latent diffusion [216]	8.21	162.3			172M	64
Diffusion	ADM+Upsample [50]	7.49	127.5	3.94	215.8	608M	2000
Diff. + VAE*	MDT [62]	6.23	143.0	1.79	283.0	676M	250
Diff. + VAE*	MaskDiT [257]	5.69	178.0	2.28	276.6	736M	40
Diffusion	CDM [89]	4.88	158.7				8100
Diffusion	RIN [102]	3.42	182.0			410M	1000
Diffusion	simple diffusion [94]	2.77	211.8	2.44	256.3	2B	512
Diffusion	VDM++ [111]	2.40	225.3	2.12	267.7	2B	512
AR-LM + VQ	VQGAN [57]	15.78	78.3			1.4B	256
MLM + VQ	MaskGIT [33]	6.18	182.1			227M	8
MLM + VQ	Token-Critic [125]	4.69	174.5			368M	36
MLM + VQ	Contextual RQ-Transformer [120]	3.41	224.6			1.4B	72
MLM + VQ	DPC [126]	4.45	244.8			454M	180
MLM + LFQ	*MAGVIT-v2 (ours)*	3.65	200.5	**1.78**	**319.4**	307M	64

Table 5.3: **Image generation results**: class-conditional generation on ImageNet 256×256. Guidance indicates the classifier-free diffusion guidance [86]. * indicates usage of extra training data. We adopt the evaluation protocol and implementation of ADM.

the training batches to enable classifier-free guidance [86]. For unguided generation, we use temperature 30 for 512×512 and 15 for 256×256 in the non-autoregressive decoding. For guided generation, we adopt the guidance schedule from [62] with temperature scaling [126], where we use guidance scale 25 with temperature 15.

As shown in Tabs. 5.2 and 5.3, our model surpasses the best performing diffusion models both in sampling quality (FID and IS) and inference-time efficiency (sampling steps). It is worth noting that all the models compared are trained using the same ImageNet training data, with a comparable model size and training budget. Therefore, the performance primarily evaluates the model's capabilities. The masked language model, equipped with our tokenizer, exhibits a notable improvement in FID over the best diffusion model baseline at 512×512 (FID=1.91 *vs.* 2.65, 28%↓). While this margin narrows at 256×256 resolution, the MLM uses a 50% reduced model size and needs much fewer decoding steps (*e.g.*, 64 *vs.* 250) to get the image generation quality. Qualitative samples in comparison with other models are shown in Fig. 5.6.

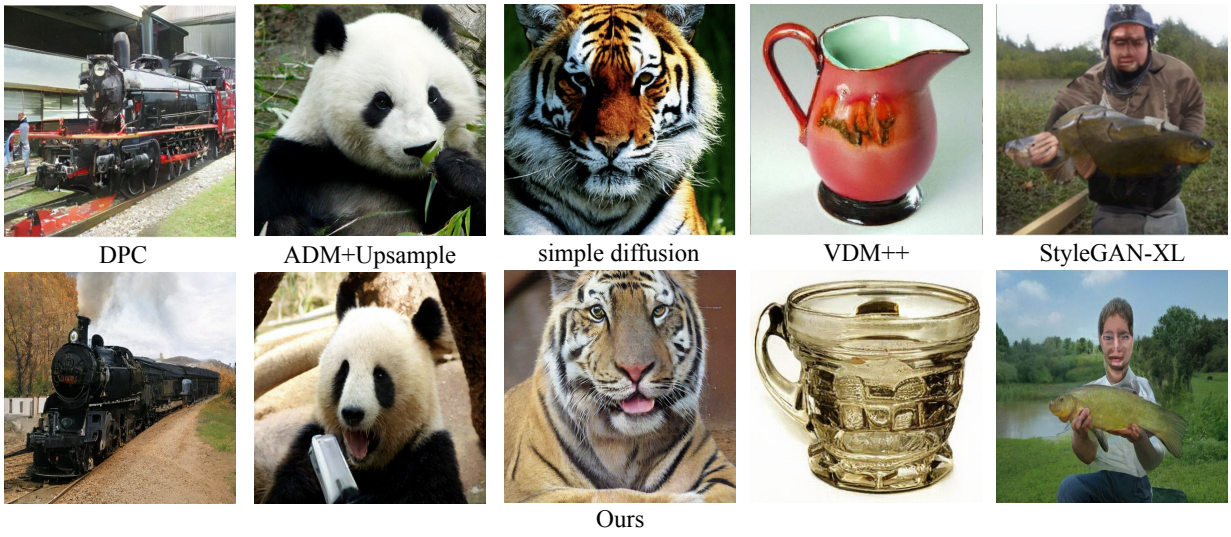


Figure 5.6: **Class-conditional generation samples on ImageNet 512×512**. We compare with each of the previous works with a random sample from the same image class.

5.4.3 Video Compression

We conduct a subjective rater study to assess the compression quality of MAGVIT-v2. The study is conducted on the 30 videos of the MCL-JCV dataset, resized to a resolution of 640×360. Sixteen raters are engaged, each providing responses to an average of roughly 800 pairwise-preference questions.

To rate the quality of the different methods, we use a two-alternative forced choice rating methodology [59]. As this methodology produces a sequence of binary decisions, we calculate Elo scores [56] based on pairwise preferences to quantify the relative visual quality between the models. The study was conducted on the 30 videos of the MCL-JCV dataset [208], scaled down to a resolution of 640×360 pixels. Sixteen raters are engaged, each providing responses to an average of roughly 800 pairwise-preference questions. The questions are presented with an interface that parallels the one used for the Challenge on Learned Image Compression (`http://compression.cc/`), extended to comparing videos, as shown in Fig. 5.7. Raters are instructed to compare the two videos and are not allowed to pause the videos.

We calculate Elo scores [56] based on pairwise preferences to quantify the relative visual quality between the models. The study compares our model with MAGVIT as well as the current video compression standard HEVC (H.265) video codec [186] and the next-generation codec VVC (H.266) [24]. As shown in Fig. 5.8, raters prefer our model to the compared methods at multiple bit rates.

We also compare the compression quality using common distortion metrics (LPIPS, PSNR, and MS-SSIM). Tab. 5.4 compares at 0.0384 bpp, the bit rate of MAGVIT, with full curves in Fig. 5.9. The results show that our model outperforms MAGVIT on all metrics, and it outperforms all methods on LPIPS, a metric which correlates more closely with subjective quality assessments. At equal bit rates, standard codecs may render local details more accurately than neural models but also introduce block artifacts, detrimental to perceptual quality yet not cap-

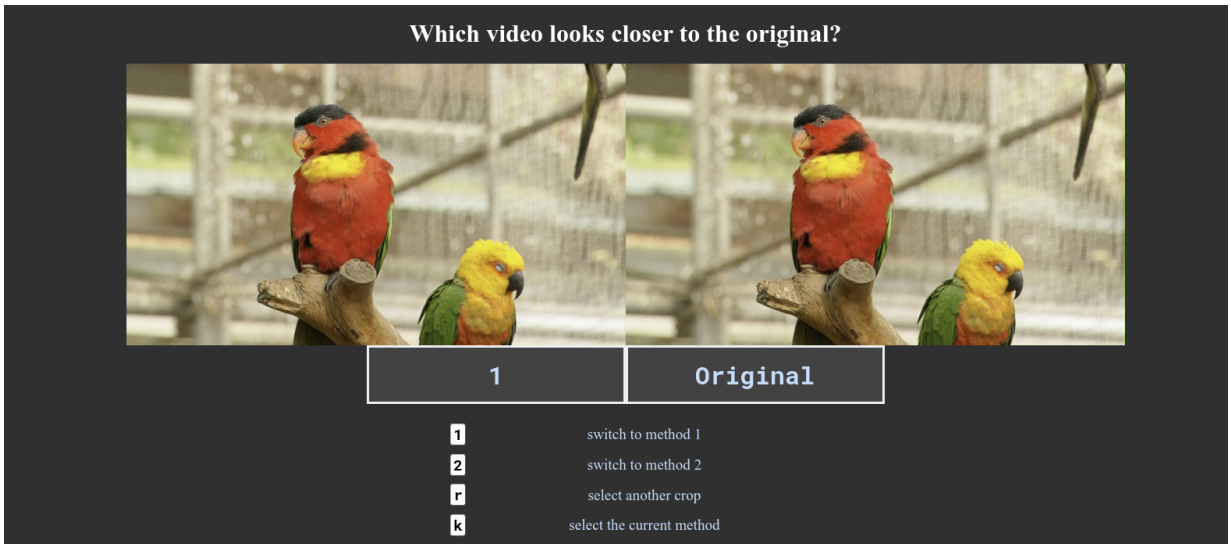


Figure 5.7: **Rating interface for subjective compression evaluation**.

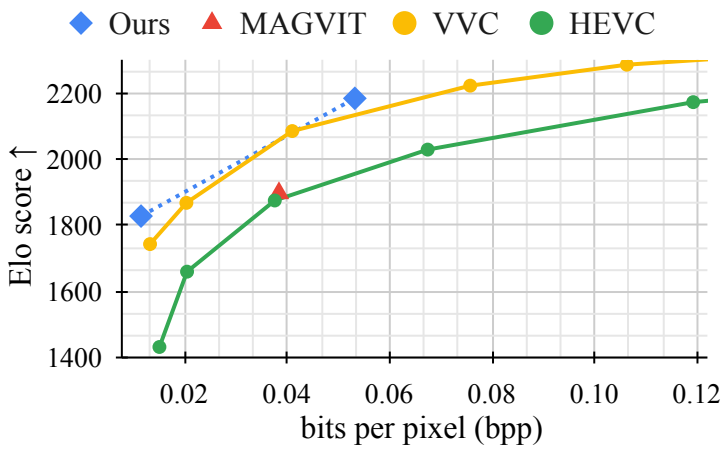
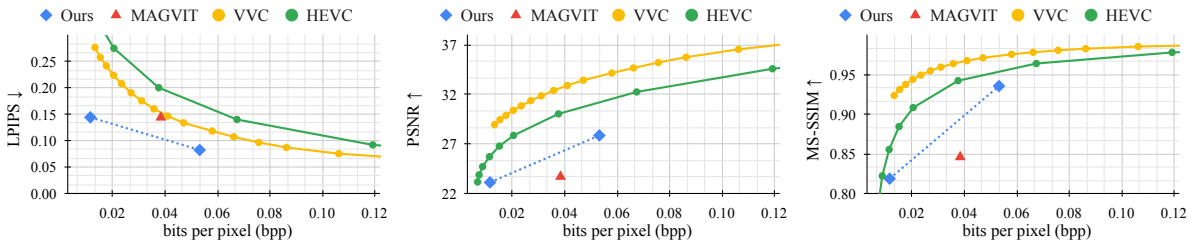


Figure 5.8: **Video compression rater study**.



(a) LPIPS↓ (b) PSNR↑ (c) MS-SSIM↑

Figure 5.9: **Video compression metrics**, supplementary to Tab. 5.4.

Method	LPIPS↓	PSNR↑	MS-SSIM↑
HEVC [186]	0.199	30.10	0.943
VVC [24]	0.153	**32.65**	**0.966**
MAGVIT [242]	0.144	23.70	0.846
MAGVIT-v2 (ours)	**0.104**	26.18	0.894

Table 5.4: **Video compression metrics**.

Token as transformer's:	Output	Input		
Tokenizer	SSv2	SSv2	K400	K600
3D VQ-VAE	64.13	41.27	44.44	45.67
MAGVIT [242]	67.22	57.34	72.29	74.65
MAGVIT-v2 (ours)	**67.38**	**62.40**	**75.34**	**77.93**
Raw pixel	64.83	63.08	76.13	78.92
HoG descriptor [220]	65.86	n/a	n/a	n/a

Table 5.5: **Video action recognition performance** (classification accuracy↑ ×100).

tured by PSNR and MS-SSIM [4]. Despite promising results with TPUs, further research is needed to adapt our model to run efficiently on CPUs like standard codecs.

5.4.4 Video Understanding

In this subsection, we assess the tokenizer's capability to learn a video understanding model for action recognition. Two setups are examined: (1) using tokens as prediction targets for the transformer's output, and (2) using tokens as the input to the transformer. For the former setup, we use a similar architecture following the BEVT [211] pre-training. For the tokens as inputs, to work with the ViViT backbone [9], we detokenize the tokens to pixels before feeding them to frozen ViViT transformers trained on raw pixels.

Tokens as prediction targets. BEiT [14] and BEVT [211] class of models pretrain visual encoders on pixel inputs by predicting tokens as targets in a masked-modeling framework, and demonstrate state-of-the-art downstream results. We use a simplified BEVT pre-training setup to test the effectiveness of our video tokens as targets for masked modeling. The main difference is that we drop the image-stream from pre-training and only use the video stream and for this reason, we also drop the multiple decoders completely and adopt an encoder-only architecture similar to BEiT. Detailed pre-training and fine-tuning setup is presented in Tab. 5.6.

Tokens as inputs. We show that we can re-use video understanding models trained on pixels using our video tokens as input, with very minimal performance drop. For this experiment, we

Config	SSv2 Pre-Training	SSv2 Fine-tuning
inputs	pixels	pixels
input size	$16 \times 224 \times 224 \times 3$	$16 \times 224 \times 224 \times 3$
targets	tokens	classes
encoder	ViT-B	ViT-B
decoder	linear	linear
masking	block-tube [211]	none
masking ratio	0.75	0.0
mask temporal length	16	0
batch size	1024	512
training epochs	800	50
ViT sequence length	$8 \times 16 \times 16$	$8 \times 16 \times 16$
optimization		
optimizer	AdamW	AdamW
optimizer momentum	0.9	0.9
layer decay	0.75	0.75
weight decay	0.05	0.05
learning rate schedule	cosine decay	cosine decay
warmup epochs	40	5
data augmentations		
random horizontal flip	true	false
label smoothing	0.1	0.1
mixup	none	0.8
cutmix	none	1.0
droppath	0.0	0.1
dropout	0.1	0.0
random color augmentation	false	false

Table 5.6: **Experimental configurations with tokens as targets**.

train a factorized variant of the ViViT model [9] on pixels, and evaluate it on de-tokenized pixels from our model. We use the same hyper-parameters as used in [9] with a Base sized model operating on 32 frames of inputs at 224p resolution. For the Kinetics-600 experiment, we use the same hyper-parameters as the Kinetics-400 experiments.

Tab. 5.5 shows that MAGVIT-v2 outperforms the previous best MAGVIT in these evaluations. Specifically, when using the decoded tokens as input, the performance approaches that of the model trained with ground-truth pixels using the same ViViT backbone. While these numbers are still worse than the state-of-the-art in action recognition, they represent solid improvements credited to the new tokenizer.

5.4.5 Ablation Study

In Fig. 5.1, we have ablated LFQ *vs.* VQ and the vocabulary size. In Tab. 5.7, we validate the key designs proposed in Section 5.3.2. Specifically, Tab. 5.7a compares the architecture illustrated in Fig. 5.2; Tab. 5.7b and Tab. 5.7c verify the LFQ and other improvements on ImageNet and UCF-101, respectively.

Tab. 5.7d verifies the entropy penalty $\mathcal{L}_{entropy}$ introduced in Eq. (5.5), which plays an important role under the LFQ setup to achieve better reconstruction and better codebook utilization. When using binary latents, certain LFQ dimensions may become dead if the dataset is not complexed enough for the large vocabulary size. In such cases, $\mathcal{L}_{entropy}$ is beneficial to ensure the activation of all bits, which enforces a discrete uniform prior in the latent space. Multi-variate latents are less likely to have dead dimensions, since fewer dimensions are used at the same vocabulary size. A potential alternative to $\mathcal{L}_{entropy}$ is applying normalization before quantization, such that the continuous latents are centered around 0. The specific choice of normalization is left as future work.

5.5 Summary

We introduce MAGVIT-v2, a novel video tokenizer that exploits lookup-free quantization along with architectural advancements to tokenize images and video with a shared vocabulary. The experiments show that our tokenizer outperforms the previously leading video tokenizer across three areas: visual generation, video compression, and action recognition in videos. Our results suggest that a good visual tokenizer is key for enabling language models to excel in image and video generation. These results demonstrate the great capabilities of LMs in visual generation, and advocate for further exploration of advanced visual tokenization methods as *multi-modal latent representation* designed for LLMs.

	#Params	FID↓	FVD↓
MAGVIT	39M	n/a	107.15
C-ViViT	90M	28.02	437.54
C-ViViT + MAGVIT	67M	13.52	316.70
MAGVIT-v2: Causal 3D CNN	58M	**7.06**	**96.33**

(a) Causal architectures on UCF-101. FID is calculated on the first frame.

	FID↓	LPIPS↓
MAGVIT	2.65	0.1292
+ LFQ	2.48	0.1182
+ large vocabulary	1.34	0.0821
+ up/downsampler	1.21	0.0790
+ deeper model	1.20	0.0686
+ adaptive normalization	**1.15**	**0.0685**

(b) Image tokenization on ImageNet 128×128.

	FVD↓	LPIPS↓
MAGVIT	24.55	0.0988
+ LFQ & large vocabulary	16.12	0.0694
+ up/downsampler	15.37	0.0678
+ late temporal downsample	11.11	0.0653
+ deeper model	8.90	0.0542
+ 3D blur pooling	**8.62**	**0.0537**

(c) Video tokenization on UCF-101.

$\mathcal{L}_{entropy}$	L1↓	L2↓	Utilization↑	Entropy↑
	0.174	0.052	0.219	3.12
✓	**0.147**	**0.039**	**1.000**	**7.18**

(d) Entropy penalty on CIFAR10 with a 2^8 LFQ vocabulary.

Table 5.7: **Ablation study verifying key design choices**.

Part III

Multi-Task Generative Models

Part III Overview. Harnessing the acquired high-fidelity representations detailed in Part II, we possess the capacity to construct latent generative models that adeptly perceive, comprehend, and replicate the intricacies of the world. Within this section, our concentration is directed toward formulating techniques for data modeling and shaping task structures. Notably, we present methodologies tailored to facilitate multi-task learning using a solitary model.

In Chapter 6, we unveil a multi-task video generation model, leveraging the capabilities of *masked generative transformers*. By utilizing the spatial-temporal vector-quantized representation detailed in Chapter 3, videos are conceptualized as sequences of visual tokens within the latent space. To enrich the landscape of multi-task learning, an effective embedding technique for masked video token modeling is introduced. Remarkably, a single model, with no alterations, supports an array of conditional *video generation* tasks, encompassing scenarios where input involves a subset of pixels or an embedding. This model not only exhibits an adaptability spectrum across diverse tasks but also attains a favorable level of video generation quality, alongside an efficient sampling process.

In Chapter 7, we delve into the realm of generating *video, image, and text* through a frozen Large Language Model (LLM), fortified by the visual lexical representation introduced in Chapter 4. Our approach introduces a progressive in-context learning methodology, empowering static LLMs to proficiently undertake both *generation and understanding* tasks spanning non-linguistic domains, including images and videos. Remarkably, even without any updates to the LLM's parameters, it showcases prowess in image and video tasks such as classification, captioning, visual question answering, text-to-image, and frame prediction.

In Chapter 8, our exploration advances as we develop *scalable generative multi-modal transformers* from the ground up, utilizing the scalable representation conceptualized in Chapter 5. This development employs modality-specific discrete tokenization to cohesively integrate text, images, videos, and audio within a decoder-only, transformer-based framework akin to LLMs. By pretraining this model on a broad array of multi-modal generative tasks using the established LLM training methodologies, we endow the model with robust capabilities for multi-task video generation. Notably, this model represents a pioneering achievement in its ability to generate high-quality videos, complete with corresponding audio, based on a wide range of input signals.

Chapter 6

Masked Generative Video Transformer

Overview. In this chapter, we introduce the MAsked Generative VIdeo Transformer (MAGVIT) to tackle various video synthesis tasks with a single model. With the 3D tokenizer introduced in Chapter 3, we propose an embedding method for masked video token modeling to facilitate *multi-task* learning. We conduct extensive experiments to demonstrate the quality, efficiency, and flexibility of MAGVIT. Our experiments show that (i) MAGVIT performs favorably against state-of-the-art approaches and establishes the best-published FVD on three video generation benchmarks, including the challenging Kinetics-600. (ii) MAGVIT outperforms existing methods in inference time by two orders of magnitude against diffusion models and by 60× against autoregressive models. (iii) A single MAGVIT model supports ten diverse generation tasks and generalizes across videos from different visual domains. The source code and trained models are released to the public at `https://magvit.cs.cmu.edu`.

6.1 Motivation

Recent years have witnessed significant advances in image and video content creation based on learning frameworks ranging from generative adversarial networks (GANs) [42, 136, 170, 196, 206], diffusion models [71, 90, 95, 165, 205], to vision transformers [143, 158, 222]. Inspired by the recent success of generative image transformers such as DALL·E [160] and other approaches [33, 51, 57, 235], we propose an efficient and effective video generation model by leveraging masked token modeling and multi-task learning.

We introduce the MAsked Generative VIdeo Transformer (*MAGVIT*) for multi-task video generation. Specifically, we build and train a single MAGVIT model to perform a variety of diverse video generation tasks and demonstrate the model's efficiency, effectiveness, and flexibility against state-of-the-art approaches. Fig. 6.1(a) shows the quality metrics of MAGVIT on a few benchmarks with efficiency comparisons in (b), and generated examples under different task setups such as frame prediction/interpolation, out/in-painting, and class conditional generation in (c).

MAGVIT models a video as a sequence of visual tokens in the latent space and learns to predict masked tokens with BERT [49]. We adopt the 3D quantization model introduced in Chapter 3 and propose an effective *masked token modeling* (MTM) scheme for multi-task video generation. Unlike conventional MTM in image understanding [213] or image/video synthesis [33, 74, 77], we present an embedding method to model a video condition using a multivariate mask and show its efficacy in training.

We conduct extensive experiments to demonstrate the quality, efficiency, and flexibility of MAGVIT against state-of-the-art approaches. Specifically, we show that MAGVIT performs favorably on two video generation tasks across three benchmark datasets, including UCF-101 [184], BAIR Robot Pushing [55, 198], and Kinetics-600 [30]. For the class-conditional generation task on UCF-101, MAGVIT reduces state-of-the-art FVD [198] from 332 [63] to 76 ($\downarrow$77%). For the frame prediction task, MAGVIT performs best in terms of FVD on BAIR (84 [95] $\rightarrow$ 62, $\downarrow$26%) and Kinetics-600 (16 [90] $\rightarrow$ 9.9, $\downarrow$38%).

Aside from the visual quality, MAGVIT's video synthesis is highly efficient. For instance, MAGVIT generates a 16-frame 128×128 video clip in 12 steps, which takes 0.25 seconds on a single TPUv4i [106] device. On a V100 GPU, a base variant of MAGVIT runs at 37 frame-per-second (fps) at 128×128 resolution. When compared at the same resolution, MAGVIT is two orders of magnitude faster than the video diffusion model [90]. In addition, MAGVIT is 60 times faster than the autoregressive video transformer [63] and 4-16 times more efficient than the contemporary non-autoregressive video transformer [74].

We show that MAGVIT is flexible and robust for multiple video generation tasks with a single trained model, including frame interpolation, class-conditional frame prediction, inpainting, and outpainting, etc. In addition, MAGVIT learns to synthesize videos with complex scenes and motion contents from diverse and distinct visual domains, including actions with objects [68], autonomous driving [27], and object-centric videos from multiple views [5].

The main contributions of this chapter are:

- To the best of our knowledge, we present the first masked multi-task transformer for efficient video generation and manipulation. We show that a trained model can perform ten different tasks at inference time.

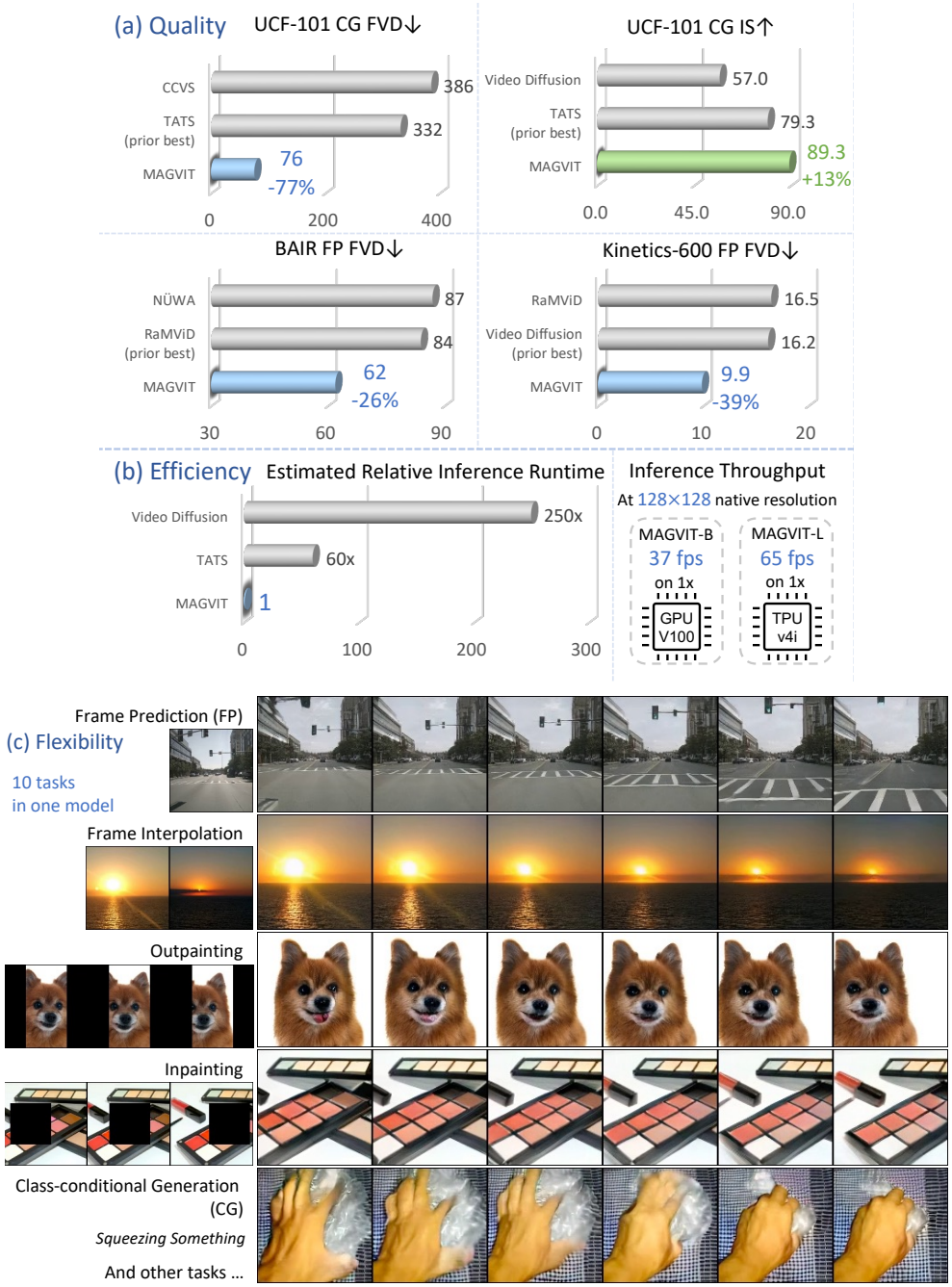


Figure 6.1: **Overview of the video generation *quality*, *efficiency*, and *flexibility* of the proposed MAGVIT model**. (a) MAGVIT achieves the state-of-the-art FVD [198] and Inception Score (IS) [171] on two video generation tasks and three benchmarks, in comparison with prior best diffusion models (RaMViD [95], Video Diffusion [90]) and autoregressive models (CCVS [116], TATS [63], NÜWA [225]). (b) It is two orders of magnitude faster than diffusion models and 60× faster than autoregressive models. (c) A single MAGVIT model accommodates different generation tasks, ranging from class-conditional generation to dynamic inpainting of a moving object.

- We propose an effective embedding method with diverse masks for numerous video generation tasks.
- We show that MAGVIT achieves the best-published fidelity on three widely-used benchmarks, including UCF-101, BAIR Robot Pushing, and Kinetics-600 datasets.

6.2 Prior Work

GAN-based approaches. Early success in video synthesis has been made by GAN models [2, 21, 22, 42, 76, 107, 170, 180, 189, 196, 206, 247]. Training instability and lack of generation diversity [33] are known issues of GAN models.

Autoregressive transformers. Inspired by the success of GPT [25], autoregressive transformers have been adapted for image [37, 51, 57, 160, 235] and video generation [13, 93, 222, 225]. A focus for video is autoregressive modeling of visual dynamics. Studies have switched from modeling the raw pixels [37, 199] to the discrete codes in a latent space [158, 232]. The state-of-the-art model TATS [63] uses two hierarchical transformers to reduce the computation for long video generation, with tokens learned by a 3D-VQGAN [57]. Unlike prior works, we introduce a non-autoregressive transformer with higher efficiency and flexibility.

Non-autoregressive transformers. Concurrently, a few methods use non-autoregressive transformers for image synthesis [33, 125, 182, 187, 256]. Section 6.3 reviews a state-of-the-art model called MaskGIT [33]. Compared with these approaches [74, 77], we present an embedding mask to model multi-task video conditions with better quality.

Diffusion models. Diffusion models have recently received much attention for image synthesis. For example, the state-of-the-art video diffusion model [90] extends the image denoising diffusion model [10, 87, 181, 183, 197] by incorporating 3D U-Net [41] architectures and joint training on both images and videos. Despite its high-quality, sampling speed is a bottleneck hindering the application of diffusion models in video synthesis. We show a different solution to train a highly-efficient model that offers compelling quality.

Multi-task video synthesis. Multi-task video synthesis [77, 143, 225] is yet to be well-studied. Transframer [143] is the closest to our work, which adopts an image-level representation for autoregressive modeling of tasks based on frame prediction. We present an efficient non-autoregressive multi-task transformer, and verify the quality and efficiency on ten video generation tasks.

Text-to-video. All of our models are trained only on public benchmarks, except the Web video model. We leave the text-to-video task as future work. As shown in recent works [88, 177, 203], training such models requires large, and sometimes non-public, datasets of paired texts and images.

6.3 Preliminaries: Masked Image Synthesis

The proposed video generation framework is based on a two-stage image synthesis process [57, 160] with non-autoregressive transformers [33, 125]. In the first stage, an image is quantized and flattened into a sequence of discrete tokens by a Vector-Quantized (VQ) auto-encoder [57, 200, 234]. In the second stage, masked token modeling (MTM) is used to train a transformer model [33, 71] on the tokens. Let $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ be an image and $\mathbf{z} \in \mathbb{Z}^N$ denote the corresponding token sequence of length N.

We take MaskGIT [33] as an example. In the second stage, it applies a binary mask $\mathrm{m}_i \in \{x \to x, x \to [\mathrm{MASK}]\}$ to each token to build a corrupted sequence $\bar{\mathbf{z}} = \mathbf{m}(\mathbf{z})$. Condition inputs, such as class labels, are incorporated as the prefix tokens $\mathbf{c}$. A BERT [49] parameterized by θ is learned to predict the masked tokens in the input sequence $[\mathbf{c}, \bar{\mathbf{z}}]$, where $[\cdot, \cdot]$ concatenates the sequences. The objective is to minimize the cross-entropy between the predicted and the ground-truth token at each masked position:

$$\mathcal{L}_{\text{mask}}(\mathbf{z}; \theta) = \mathop{\mathbb{E}}_{\mathbf{m} \sim p_{\mathcal{U}}} \Big[\sum_{\bar{\mathrm{z}}_i = [\mathrm{MASK}]} -\log p_\theta(\mathrm{z}_i \mid [\mathbf{c}, \bar{\mathbf{z}}]) \Big]. \tag{6.1}$$

During training, MaskGIT randomly samples $\mathbf{m}$ from a prior distribution $p_{\mathcal{U}}$ where the mask ratio follows a cosine scheduling function $\gamma(\cdot)$ [33]. Specifically, it first uniformly samples a per-token mask score $\mathrm{s}_i \sim \mathcal{U}(0, 1)$ to form a sequence denoted as $\mathbf{s}$. Then it samples $r \sim \mathcal{U}(0, 1)$ and computes a cut-off threshold s^* as the $\lceil \gamma(r)N \rceil$-th smallest element in $\mathbf{s}$. Finally, a mask $\mathbf{m}$ is created such that $\mathrm{m}_i(x) = [\mathrm{MASK}]$ if $\mathrm{s}_i \leq s^*$ and $\mathrm{m}_i(x) = x$ otherwise.

For inference, the non-autoregressive decoding method [67, 69, 114] is used to synthesize an image [33, 125, 256]. For example, MaskGIT generates an image in K=12 steps [33] from a blank canvas with all visual tokens masked out. At each step, it predicts all tokens in parallel while retaining tokens with the highest prediction scores. The remaining tokens are masked and predicted in the next iteration until all tokens are generated. Similar to the training stage, the mask ratio is computed by the schedule function γ, but with a deterministic input as $\gamma(\frac{t}{K})$, where t is the current step.

6.4 MAGVIT: Masked Generative Video Transformer

Our goal is to design a multi-task video generation model with high quality and inference efficiency. We propose MAsked Generative VIdeo Transformer (MAGVIT), a vision transformer framework that leverages masked token modeling and multi-task learning. MAGVIT generates a video from task-specific condition inputs, such as a frame, a partially-observed video volume, or a class identifier.

The framework consists of two stages. First, we learn a 3D vector-quantized (VQ) autoencoder to quantize a video into discrete tokens, as introduced in Chapter 3. In the second stage, we learn a video transformer by multi-task masked token modeling.

Fig. 6.2 illustrates the training in the second stage. At each training step, we sample one of the tasks with its prompt token, obtain a task-specific conditional mask, and optimize the transformer to predict all target tokens given masked inputs. During inference, we adapt the

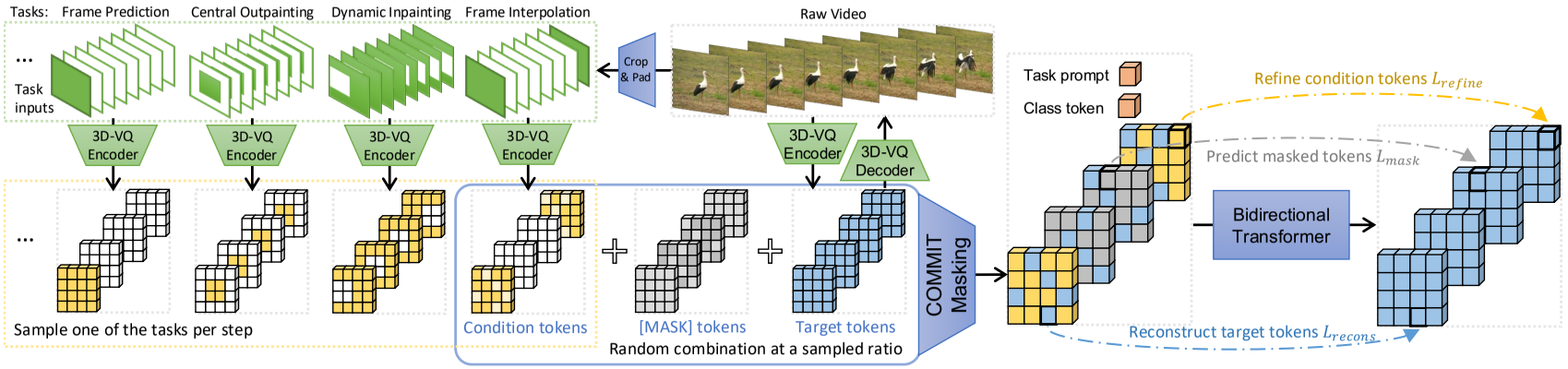


Figure 6.2: **MAGVIT pipeline overview**. The 3D-VQ encoder quantizes a video into discrete tokens, while the 3D-VQ decoder maps them back to the pixel space. We sample one of the tasks at each training step and build its condition inputs by cropping and padding the raw video, where green denotes valid pixels and white is padding. We quantize the condition inputs with the 3D-VQ encoder and select the non-padding part as condition tokens. The masked token sequence combines condition tokens, [MASK] tokens, and the target tokens, with a task prompt and a class token as the prefix. The bidirectional transformer learns to predict the target tokens through three objectives: refining condition tokens, predicting masked tokens, and reconstructing target tokens.

non-autoregressive decoding method to generate tokens conditionally on the task-specific inputs, which will be detailed in Algorithm 1.

6.4.1 Multi-Task Masked Token Modeling

In MAGVIT, we adopt various masking schemes to facilitate learning for video generation tasks with different conditions. The conditions can be a spatial region for inpainting/outpainting or a few frames for frame prediction/interpolation. We refer to these partially-observed video conditions as *interior conditions*.

We argue that it is suboptimal to directly unmask the tokens corresponding to the region of the interior condition [33]. As discussed in Section 3.2.1, the non-local receptive field of the tokenizer can leak the ground-truth information into the unmasked tokens, leading to problematic non-causal masking and poor generalization.

We propose a method, COnditional Masked Modeling by Interior Tokens (or *COMMIT* for short), to embed interior conditions inside the corrupted visual tokens.

Training. Each training example includes a video $\mathbf{V}$ and the optional class annotation $\mathbf{c}$. The target visual tokens come from the 3D-VQ as $\mathbf{z} = f_{\mathcal{T}}(\mathbf{V})$. At each step, we sample a task prompt ρ, obtain the task-specific interior condition pixels, pad it into $\tilde{\mathbf{V}}$ with the same shape as $\mathbf{V}$, and get the condition tokens $\tilde{\mathbf{z}} = f_{\mathcal{T}}(\tilde{\mathbf{V}})$. Section 6.4.2 lists the padding functions for each task.

At a sampled mark ratio, we randomly replace target tokens z_i, with either 1) the condition token $\tilde{\mathrm{z}}_i$, if the corresponding supervoxel of z_i contains condition pixels; or 2) the special [MASK] token, otherwise. Formally, we compute the *multivariate* conditional mask $\mathbf{m}(\cdot \mid \tilde{\mathbf{z}})$ as

$$\mathrm{m}(\mathrm{z}_i \mid \tilde{\mathrm{z}}_i) = \begin{cases} \tilde{\mathrm{z}}_i & \text{if } \mathrm{s}_i \le s^* \wedge \neg\, \mathrm{ispad}(\tilde{\mathrm{z}}_i) \\ \texttt{[MASK]} & \text{if } \mathrm{s}_i \le s^* \wedge \mathrm{ispad}(\tilde{\mathrm{z}}_i) \\ \mathrm{z}_i & \text{if } \mathrm{s}_i > s^* \end{cases}, \tag{6.2}$$

where s_i and s^* are the per-token mask score and the cut-off score introduced in Section 6.3. $\mathrm{ispad}(\tilde{\mathrm{z}}_i)$ returns whether the corresponding supervoxel of $\tilde{\mathrm{z}}_i$ in $\tilde{\mathbf{V}}$ only contains padding.

Eq. (6.2) indicates that COMMIT embeds interior conditions as corrupted visual tokens into the multivariate mask $\mathbf{m}$, which follows a new distribution $p_{\mathcal{M}}$ instead of the prior $p_{\mathcal{U}}$ for binary masks. With the corrupted token sequence $\bar{\mathbf{z}} = \mathbf{m}(\mathbf{z} \mid \tilde{\mathbf{z}})$ as input, the *multi-task* training objective is

$$\mathcal{L}(\mathbf{V}; \theta) = \mathop{\mathbb{E}}_{\rho, \tilde{\mathbf{V}}} \mathop{\mathbb{E}}_{\mathbf{m} \sim p_{\mathcal{M}}} \Big[\sum_i -\log p_\theta(\mathrm{z}_i \mid [\rho, \mathbf{c}, \bar{\mathbf{z}}]) \Big]. \tag{6.3}$$

We can decompose the loss in Eq. (6.3) into three parts according to Eq. (6.2): $\mathcal{L}_{\text{refine}}$ refines the task-specific condition tokens, $\mathcal{L}_{\text{mask}}$ predicts masked tokens , and $\mathcal{L}_{\text{recons}}$ reconstructs target

Algorithm 1 Non-autoregressive Decoding by COMMIT

Input: prefix ρ and $\mathbf{c}$, condition $\tilde{\mathbf{z}}$, steps K, temperature T
Output: predicted visual tokens $\hat{\mathbf{z}}$

$\mathbf{s} = \mathbf{0}$, $s^* = 1$, $\hat{\mathbf{z}} = \mathbf{0}^N$
for $t \leftarrow 0, 1, \cdots, K-1$ **do**
  $\overline{\mathbf{z}} \leftarrow \mathbf{m}(\hat{\mathbf{z}} \mid \tilde{\mathbf{z}}; \mathbf{s}, s^*)$
  $\hat{\mathrm{z}}_i \sim p_\theta(\mathrm{z}_i \mid [\rho, \mathbf{c}, \overline{\mathbf{z}}])$, $\forall i$ where $\mathrm{s}_i \leq s^*$
  $\mathrm{s}_i \leftarrow p_\theta(\hat{\mathrm{z}}_i \mid [\rho, \mathbf{c}, \overline{\mathbf{z}}])$, $\forall i$ where $\mathrm{s}_i \leq s^*$
  $\mathrm{s}_i \leftarrow \mathrm{s}_i + T(1 - \frac{t+1}{K})\,\mathrm{Gumbel}(0, 1)$, $\forall i$ where $\mathrm{s}_i < 1$
  $s^* \leftarrow$ The $\lceil \gamma(\frac{t+1}{K})N \rceil$-$th$ smallest value of $\mathbf{s}$
  $\mathrm{s}_i \leftarrow 1$, $\forall i$ where $\mathrm{s}_i > s^*$
end for
return $\hat{\mathbf{z}} = [\hat{\mathrm{z}}_1, \hat{\mathrm{z}}_2, \cdots, \hat{\mathrm{z}}_N]$

tokens. Let $\overline{\mathbf{c}} = [\rho, \mathbf{c}, \overline{\mathbf{z}}]$ for simplicity,

$$\sum_{i=1}^{N} -\log p_\theta(\mathrm{z}_i \mid [\rho, \mathbf{c}, \overline{\mathbf{z}}]) = \underbrace{\sum_{\overline{\mathrm{z}}_i = \tilde{\mathrm{z}}_i} -\log p_\theta(\mathrm{z}_i \mid \overline{\mathbf{c}})}_{\text{Refine condition tokens } \mathcal{L}_{\text{refine}}} + \underbrace{\sum_{\overline{\mathrm{z}}_i = [\mathtt{MASK}]} -\log p_\theta(\mathrm{z}_i \mid \overline{\mathbf{c}})}_{\text{Predict masked tokens } \mathcal{L}_{\text{mask}}} + \underbrace{\sum_{\overline{\mathrm{z}}_i = \mathrm{z}_i} -\log p_\theta(\mathrm{z}_i \mid \overline{\mathbf{c}})}_{\text{Reconstruct target tokens } \mathcal{L}_{\text{recons}}}. \tag{6.4}$$

While $\mathcal{L}_{\text{mask}}$ is the same as the MTM loss in Eq. (6.1) and $\mathcal{L}_{\text{recons}}$ sometimes is used as a regularizer (*e.g.*, in NLP tasks), $\mathcal{L}_{\text{refine}}$ is a new component introduced by COMMIT.

The COMMIT method facilitates multi-task video generation in three aspects. First, it provides a correct causal masking for all interior conditions. Second, it produces a fixed-length sequence for different conditions of arbitrary regional volume, improving training and memory efficiency since no padding tokens are needed. Third, it achieves state-of-the-art multi-task video generation results (see Tab. 6.8).

Inference. We use a non-autoregressive decoding method to generate video tokens from input conditions in K steps (*e.g.*, 12). Each decoding step follows the COMMIT masking in Eq. (6.2) with a gradually reduced mask ratio. Algorithm 1 outlines the inference procedure.

Fig. 6.3 compares the non-autoregressive image decoding [33] and our video decoding procedure. Different from the MTM decoding in [33] which performs denoising from all [MASK], COMMIT decoding starts from a *multivariate* mask that embeds the interior conditions. Guided by this mask, Algorithm 1 performs a conditional transition process toward the output tokens by replacing a portion of newly generated tokens at each step. In the end, all tokens are predicted where the interior condition tokens get refined.

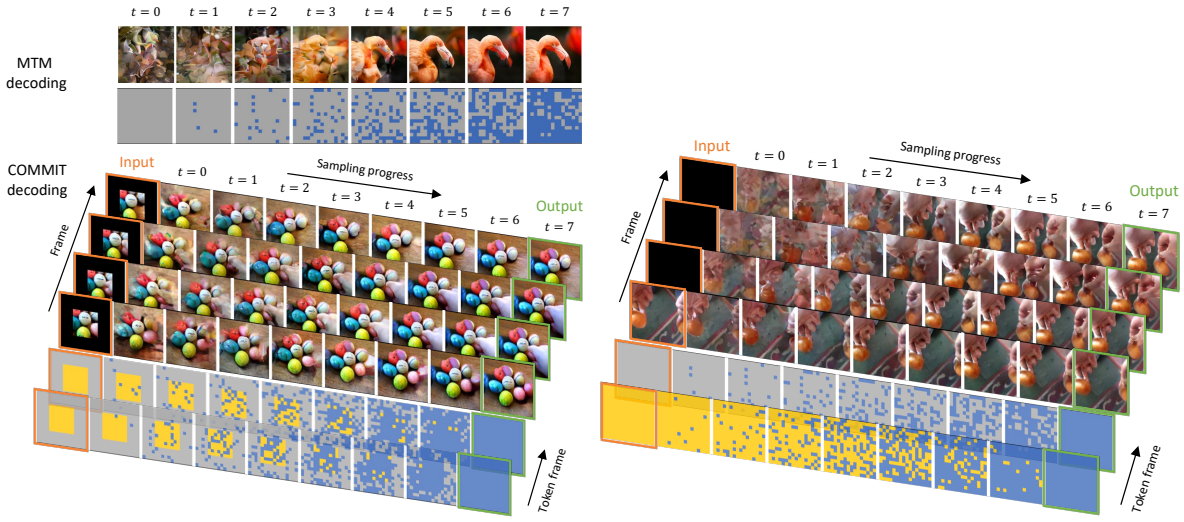
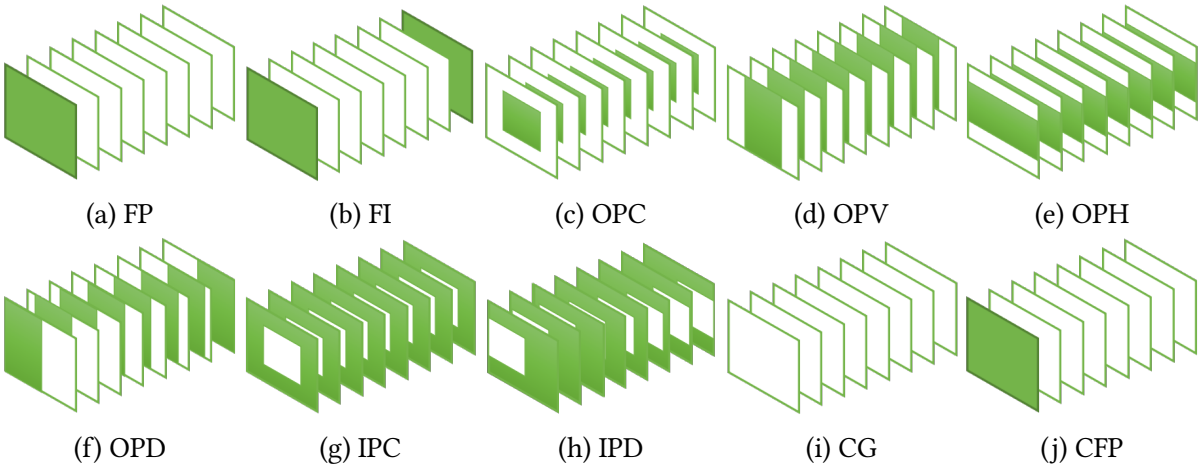


Figure 6.3: **Comparison between MTM decoding for image [33] and COMMIT decoding for video**. We show the output tokens and image/video at each decoding step t, with a central outpainting example for COMMIT on the left and a frame prediction example on the right. Unlike the MTM denoising decoding from all [MASK], COMMIT performs a conditional generation process toward the output tokens while gradually replacing the interior condition tokens. Videos and tokens are temporally down-sampled and stacked for visualization.

6.4.2 Video Generation Tasks

We employ a total of ten tasks for multi-task video generation. Each task is characterized by a few adjustable settings such as interior condition shape, padding function, and optionally prefix condition. Fig. 6.4 illustrates the interior condition regions for each task under the above setup. Given a video of shape $T \times H \times W$, we define the tasks as following:

- Frame Prediction (FP)
  - Interior condition: t frames at the beginning; $t = 1$.
  - Padding: replicate the last given frame.
- Frame Interpolation (FI)
  - Interior condition: t_1 frames at the beginning and t_2 frames at the end; $t_1 = 1, t_2 = 1$.
  - Padding: linear interpolate between the last given frame at the beginning and the first given frame at the end.
- Central Outpainting (OPC)
  - Interior condition: a rectangle at the center with height h and width w; $h = 0.5H$, $w = 0.5W$.
  - Padding: pad the nearest pixel for each location (`edge` padding).
- Vertical Outpainting (OPV)
  - Interior condition: a centered vertical strip with width w; $w = 0.5W$.
  - Padding: `edge` padding.
- Horizontal Outpainting (OPH)



(a) FP (b) FI (c) OPC (d) OPV (e) OPH

(f) OPD (g) IPC (h) IPD (i) CG (j) CFP

Figure 6.4: **Interior condition regions for each task**, where green denotes valid pixels and white pixels denote the task-specific paddings discussed in Section 6.4.2. The tasks are Frame Prediction (FP), Frame Interpolation (FI), Central Outpainting (OPC), Vertical Outpainting (OPV), Horizontal Outpainting (OPH), Dynamic Outpainting (OPD), Central Inpainting (IPC), Dynamic Inpainting (IPD), Class-conditional Generation (CG), and Class-conditional Frame Prediction (CFP).

    - Interior condition: a centered horizontal strip with height h; $h = 0.5H$.
    - Padding: `edge` padding.
- Dynamic Outpainting (OPD)
    - Interior condition: a moving vertical strip with width w; $w = 0.5W$.
    - Direction of movement: left to right.
    - Padding: zero padding.
- Central Inpainting (IPC)
    - Interior condition: everything but a rectangle at the center with height h and width w; $h = 0.5H$, $w = 0.5W$.
    - Padding: zero padding.
- Dynamic Inpainting (IPD)
    - Interior condition: everything but a vertically centered moving rectangle with height h and width w; $h = 0.5H$, $w = 0.5W$.
    - Direction of movement: left to right.
    - Padding: zero padding.
- Class-conditional Generation (CG)
    - Prefix condition: class label.
- Class-conditional Frame Prediction (CFP)
    - Prefix condition: class label.
    - Interior condition: t frames at the beginning; $t = 1$.
    - Padding: replicate the last given frame.

Model	Param.	# heads	# layers	Hidden size	MLP dim
MAGVIT-**B**	87 M	12	12	768	3072
MAGVIT-**L**	305 M	16	24	1024	4096
MAGVIT-**H**	634 M	16	32	1280	5120

Table 6.1: **Transformer architecture configurations used in MAGVIT**.

6.5 Experimental Results

We conduct extensive experiments to demonstrate the video generation quality (Section 6.5.2), efficiency (Section 6.5.3), and flexibility for multi-task generation (Section 6.5.4). We show a few generation results here, and refer to the web page https://magvit.cs.cmu.edu for more examples.

6.5.1 Experimental Setups

Datasets. We evaluate the single-task video generation performance of MAGVIT on three standard benchmarks, *i.e.*, class-conditional generation on UCF-101 [184] and frame prediction on BAIR Robot Pushing [55, 198] (1-frame condition) and Kinetics-600 [30] (5-frame condition). For multi-task video generation, we quantitatively evaluate MAGVIT on BAIR and SSv2 [68] on 8-10 tasks. Furthermore, to evaluate model generalizability, we train models with the same learning recipe on three additional video datasets: nuScenes [27], Objectron [5], and 12M Web videos.

Evaluation metrics. The FVD [198] is used as the primary evaluation metric. We follow the official implementation in extracting video features with an I3D model trained on Kinetics-400 [29] (https://github.com/google-research/google-research/tree/master/frechet_video_distance). We report Inception Score (IS) [171] on the UCF-101 dataset which is calculated with a C3D [192] model trained on UCF-101 (https://github.com/pfnet-research/tgan2). We further include image quality metrics: PSNR, SSIM [218] and LPIPS [254] (computed by the VGG features) on the BAIR dataset.

Model configurations. We train MAGVIT to generate 16-frame videos at 128×128 resolution, except for BAIR at 64×64. The proposed 3D-VQ model quantizes a video into 4×16×16 visual tokens, where the visual codebook size is 1024. We use the BERT transformer [49] adapted from the Flaxformer implementation (https://github.com/google/flaxformer) to model the token sequence, which includes 1 task prompt, 1 class token, and 1024 visual tokens. Following the transformer configurations in ViT [52], we use two variants of transformers, *i.e.*, base (B) with 87M parameters and large (L) with 306M in all our experiments. Tab. 6.1 lists the detailed configurations for each variant. A huge (H) transformer is only used to train on the large Web video dataset and generate demo videos. We train both stages with the Adam optimizer [110] in JAX/Flax [20, 82] on TPUs. Section 6.5.6 details training configurations.

6.5.2 Single-Task Video Generation

Class-conditional generation. The model is given a class identifier in this task to generate the full video. Tab. 6.2 shows a detailed comparison with the previously published results on the UCF-101 [184] class-conditional video generation benchmark, where the numbers are quoted from the cited papers. Note that CogVideo [93] and Make-A-Video [177] are pretrained on additional 5-10M videos before finetuning on UCF-101, where Make-A-Video further uses a text-image prior trained on a billion text-image pairs. The remaining models, including MAGVIT, are only trained on 9.5K training videos of UCF-101, or 13.3K training and testing videos of UCF-101 for those marked with $^{\#}$. Fig. 6.5 compares the generated videos to baseline models. We can see that CCVS+StyleGAN [116] gets a decent single-frame quality, but yields little or no motion. TATS [63] generates some motion but with artifacts. In contrast, our model produces higher-quality frames with substantial motion.

Frame prediction. The model is given a single or a few frames to generate future frames. In Tab. 6.3, we compare MAGVIT against highly-competitive baselines. MAGVIT surpasses the previous state-of-the-art FVD on BAIR by a large margin (84 → 62). Inspired by [198], a "debiased" FVD is also reported in the parentheses to overcome the small validation set. See more discussion in Section 6.5.6. In Tab. 6.4, it demonstrates better image quality.

On the large dataset of Kinetics-600, it establishes a new state-of-the-art result, improving the previous best FVD in [90] from 16.2 to 9.9 by a relative 39% improvement. The above results verify MAGVIT's compelling generation quality, including on the large Kinetics dataset. Fig. 6.6 and Fig. 6.7 below provide visual comparisons to the baseline methods on BAIR and Kinetics-600, respectively.

6.5.3 Inference-Time Generation Efficiency

Video generation efficiency is an important metric in many applications. We conduct experiments to validate that MAGVIT offers top speed in video generation. Fig. 6.8 shows the processing time for each frame on a single V100 GPU at different resolutions. We compare MAGVIT-B with an autoregressive transformer of the same size and a diffusion-based model [90]. At 128×128 resolution, MAGVIT-B runs at 37 frames-per-second (fps). When running on a single TPUv4i [106], MAGVIT-B runs at 190 fps and MAGVIT-L runs at 65 fps.

Fig. 6.8 compares the sequence lengths and inference steps of these models. Diffusion models [90] typically require 256-1000 diffusion steps with a 3D U-Net [41]. Autoregressive models, such as TATS [63], decode visual tokens sequentially, which runs 60 times slower than MAGVIT at 128×128. Compared to the recent non-autoregressive model MaskViT [74], MAGVIT is 4 to 16 times faster due to more efficient decoding on shorter sequences.

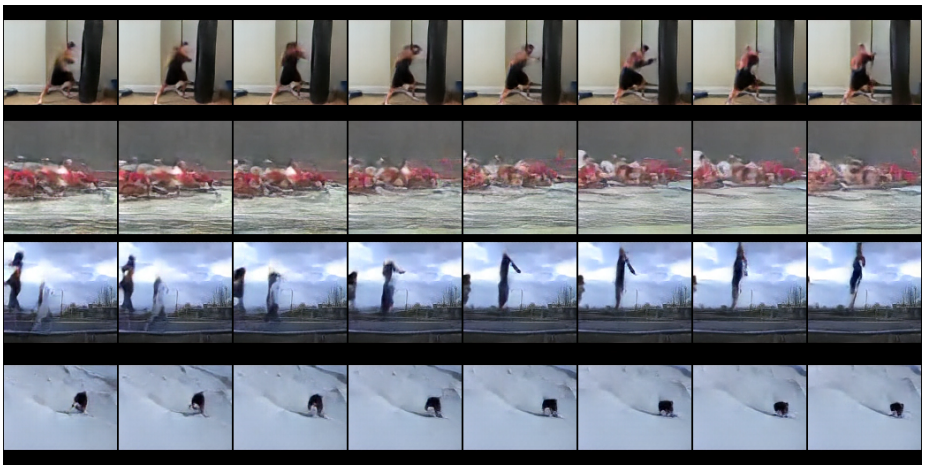
6.5.4 Multi-task Video Generation

To demonstrate the flexibility in multi-task video synthesis, we train a single MAGVIT model to perform eight tasks on BAIR or ten tasks on SSv2. We do not intend to compare with dedicated models trained on these tasks but to demonstrate a generic model for video synthesis.

Method	Extra Video	Class	FVD↓	IS↑
VGAN[206]		✓	-	8.31±0.09
TGAN[170]			-	11.85±0.07
MoCoGAN*[196]		✓	-	12.42±0.07
ProgressiveVGAN[2]		✓	-	14.56±0.05
TGAN[170]		✓	-	15.83±0.18
RaMViD[95]			-	21.71±0.21
LDVD-GAN[107]			-	22.91±0.19
StyleGAN-V*#[180]			-	23.94±0.73
VideoGPT[232]			-	24.69±0.30
TGANv2[171]		✓	1209±28	28.87±0.67
MoCoGAN-HD#[189]			838	32.36
DIGAN[247]			655±22	29.71±0.53
DIGAN#[247]			577±21	32.70±0.35
DVD-GAN#[42]		✓	-	32.97±1.70
Video Diffusion*#[90]			-	57.00±0.62
TATS[63]			420±18	57.63±0.24
CCVS+StyleGAN#[116]			386±15	24.47±0.13
Make-A-Video*[177]		✓	367	33.00
TATS[63]		✓	332±18	79.28±0.38
CogVideo*[93]	✓	✓	626	50.46
Make-A-Video*[177]	✓	✓	81	82.55
MAGVIT-B-CG (ours)		✓	159±2	83.55±0.14
MAGVIT-L-CG (ours)		✓	**76**±2	**89.27**±0.15

Table 6.2: **Generation performance on the UCF-101 dataset**. Methods in gray are pretrained on additional large video data. Methods with ✓ in the Class column are class-conditional, while the others are unconditional. Methods marked with * use custom resolutions, while the others are at 128×128. Methods marked with # additionally used the test set in training.

(a) CCVS+StyleGAN [116]



(b) TATS [63]

(c) MAGVIT-L-CG (ours)

Figure 6.5: **Comparison of class-conditional generation samples on UCF-101**. 16-frame videos are generated at 128×128 resolution 25 fps and shown at 12.5 fps. Samples for [63, 116] are obtained from their official release (https://songweige.github.io/projects/tats/).

Method	K600 FVD↓	BAIR FVD↓
LVT[158]	224.7	126±3
Video Transformer[222]	170.0±5.0	94±2
CogVideo*[93]	109.2	-
DVD-GAN-FP[42]	69.1±1.2	110
CCVS[116]	55.0±1.0	99±2
Phenaki[203]	36.4±0.2	97
VideoGPT[232]	-	103
TrIVD-GAN-FP[136]	25.7±0.7	103
Transframer[143]	25.4	100
MaskViT[74]	-	94
FitVid[13]	-	94
MCVD[205]	-	90
NÜWA[225]	-	87
RaMViD[95]	16.5	84
Video Diffusion[90]	16.2±0.3	-
MAGVIT-B-FP (ours)	24.5±0.9	76±0.1 (47±0.1)
MAGVIT-L-FP (ours)	**9.9**±0.3	**62**±0.1 (31±0.2)

Table 6.3: **Frame prediction performance on the BAIR and Kinetics-600 datasets**. - marks that the value is unavailable in their paper or incomparable to others. The FVD in parentheses uses a debiased evaluation protocol on BAIR detailed in Section 6.5.6. Methods marked with * is pretrained on additional large video data.

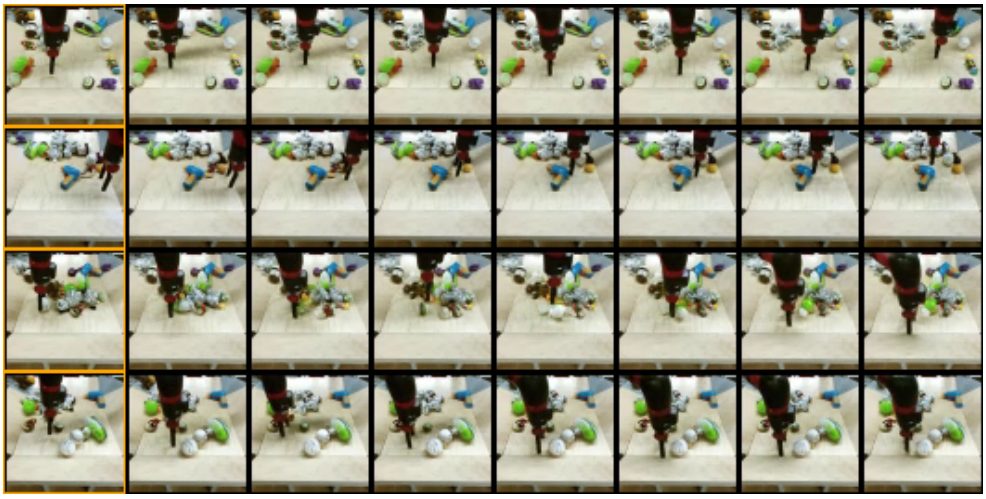
Method	FVD↓	PSNR↑	SSIM↑	LPIPS↓
CCVS[116]	99	-	0.729	-
MCVD[205]	90	16.9	0.780	-
MAGVIT-L-FP (ours)	**62**	**19.3**	**0.787**	0.123

Table 6.4: **Image quality metrics on BAIR frame prediction**.

Method	Task	**Avg↓**	FP	FI	OPC	OPV	OPH	OPD	IPC	IPD
MAGVIT-B-UNC	Single	150.6	74.0	71.4	119.0	46.7	55.9	389.3	145.0	303.2
MAGVIT-B-FP	Single	201.1	47.7	56.2	247.1	118.5	142.7	366.3	357.3	272.7
MAGVIT-B-*MT*	Multi	32.8	47.2	36.0	28.1	29.0	27.8	32.1	31.1	31.0
MAGVIT-L-*MT*	Multi	**22.8**	31.4	26.4	21.3	21.2	19.5	20.9	21.3	20.3

Table 6.5: **Multi-task generation performance on BAIR evaluated by FVD**. Gray values denote unseen tasks during training.

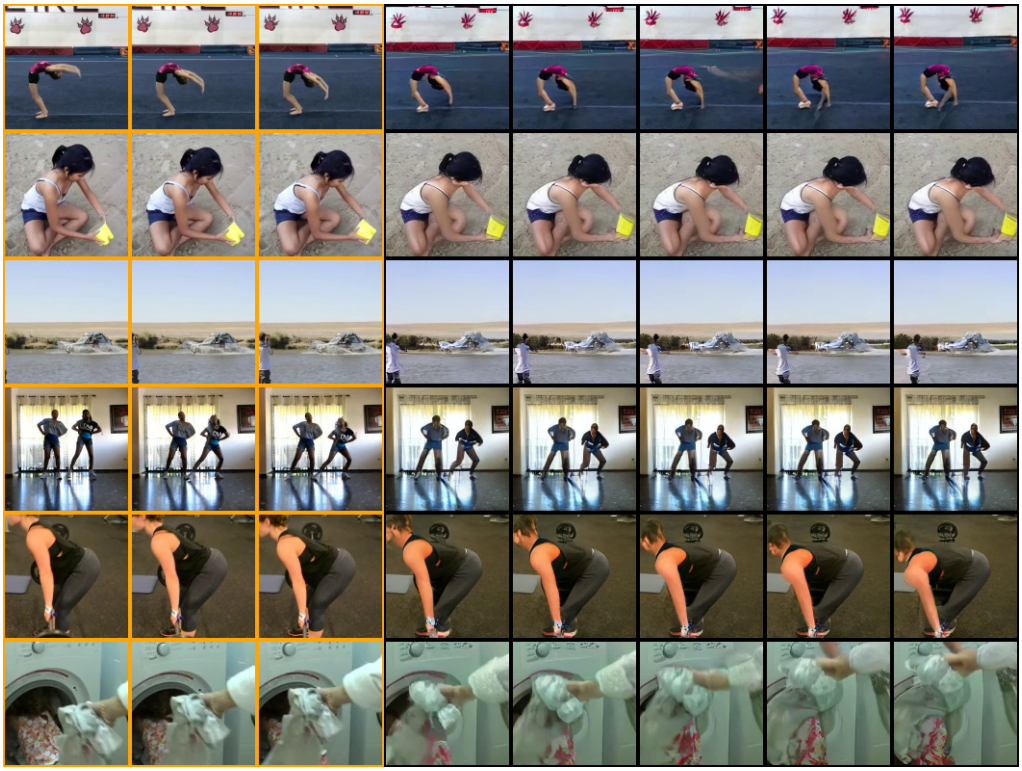
(a) RaMViD [95]



(b) MAGVIT-L-FP (ours)

Figure 6.6: **Comparison of frame prediction samples on BAIR unseen evaluation set**. 16-frame videos are generated at 64×64 resolution 10 fps given the first frame as condition and shown at 5 fps where condition frames are marked in orange. Samples for [95] are obtained from their official release (https://sites.google.com/view/video-diffusion-prediction). As shown, the clips produced by MAGVIT maintaining a better visual consistency and spatial-temporal dynamics.

(a) RaMViD [95] at 64×64 resolution, condition information is unavailable.



(b) MAGVIT-L-FP (ours) at 128×128 resolution, condition frames are marked in orange.

Figure 6.7: **Comparison of frame prediction samples on Kinetics-600 unseen evaluation set**. 16-frame videos are generated at 25 fps given 5-frame condition. Samples for [95] are obtained from their official release (https://sites.google.com/view/video-diffusion-prediction). As shown, given the conditioned frames, MAGVIT generates plausible actions with greater details.

Method	Task	SSv2-**Avg**↓	FP	FI	OPC	OPV	OPH	OPD	IPC	IPD	CG	CFP
MAGVIT-B-UNC	Single	258.8	278.8	91.0	67.5	27.3	36.2	711.5	319.3	669.8	107.7	279.0
MAGVIT-B-FP	Single	402.9	59.3	76.2	213.2	81.2	86.3	632.7	343.1	697.9	1780.0	59.3
MAGVIT-B-*MT*	Multi	43.4	71.5	38.0	38.8	23.3	26.1	33.4	23.3	25.3	94.7	59.3
MAGVIT-L-*MT*	Multi	**27.3**	33.8	25.0	21.1	16.8	17.0	23.5	13.5	15.0	79.1	28.5
Masked pixel	-	-	94%	87%	75%	50%	50%	50%	25%	25%	100%	94%
Masked token	-	-	75%	50%	75%	50%	50%	50%	25%	25%	100%	75%

Table 6.6: **Multi-task generation performance on Something-Something-V2 evaluated by FVD**. Gray values denote unseen tasks during training. The bottom two rows list the proportions of masked pixels and tokens for each task.

Method	nuScenes-FP	Objectron-FI	Web12M-**MT8**	FP	FI	OPC	OPV	OPH	OPD	IPC	IPD
MAGVIT-B	29.3	-	33.0	84.9	33.9	34.4	21.5	22.1	26.0	20.7	20.4
MAGVIT-L	20.6	26.7	**21.6**	45.5	30.9	19.9	15.3	14.5	20.2	12.0	14.7

Table 6.7: **Multi-task generation performance on NuScenes, Objectron, and Web videos evaluated by FVD**.

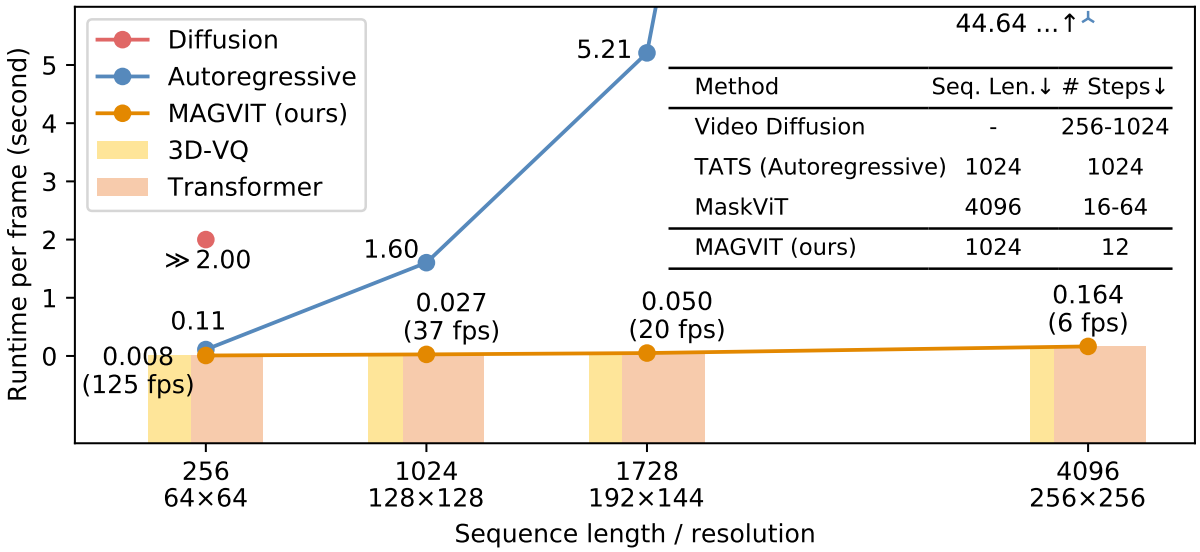


Figure 6.8: **Inference-time generation efficiency comparison**. The average runtime for generating one frame is measured at different resolutions. The colored bars show the time breakdown between the 3D-VQ and the transformer. The embedded table compares the critical factors of inference efficiency for different methods at 16-frame 128×128, except for Video Diffusion [90] at 64×64.

Eight tasks on BAIR. We perform a multi-task evaluation on BAIR with eight self-supervised tasks. Tab. 6.5 lists the "debiased" FVD for each task, where the third column computes the average. We compare the multi-task models (MT) with two single-task baselines trained on unconditional generation (UNC) and frame prediction (FP).

As shown in Tab. 6.5, the multi-task models achieve better fidelity across all tasks. Single-task models perform considerably worse on the tasks unseen in training (gray values in Tab. 6.5), especially on the tasks that differ more from the training task. Compared to the single-task models in their training task, MT performs better with a small gain on FP with the same model size.

Ten tasks on SSv2. We evaluate on the large-scale SSv2 dataset, where MAGVIT needs to synthesize 174 basic actions with everyday objects. We evaluate a total of ten tasks, with two of them using class labels (CG and CFP), as shown in Tab. 6.6. We observe a pattern consistent with BAIR: multi-task models achieve better average FVD across all tasks. Fig. 6.9 shows examples of generated videos for each task. The above results substantiate model generalization trained with the proposed multi-task objective.

Results on nuScenes, Objectron, and 12M Web Videos. Tab. 6.7 shows the generation performance on three additional datasets, *i.e.*, nuScenes [27], Objectron [5], and Web12M which contains 12 million videos we collected from the web. We evaluate our model on the frame prediction task on nuScenes, the frame interpolation task on Objectron, and the 8-task suite on the Web videos. Fig. 6.10 shows examples of generated videos for each task. The results

Method		Seq. Length	FP FVD↓	MT8 FVD↓
Latent masking in MaskGIT [33]		1024	74	151
Prefix condition		1024-1792	55	-
COMMIT (ours)	$\mathcal{L}_{\text{mask}}$	1024	388	143
	$\mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{recons}}$		51	53
	$\mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{refine}}$		**48**	**33**

Table 6.8: **Comparison of conditional masked token modeling** on BAIR frame prediction (FP) and eight-task (MT8) benchmarks. - indicates we were not able to train to convergence.

Decoding Method	Tokenizer	Type	Param.	Seq. Len.↓	# Steps↓	FVD↓
MaskGIT [33]	2D-VQ	NAR	53M+87M	4096	12	222 (177)
	3D-VQ	NAR	41M+87M	1024	12	122 (74)
MaskViT [74]	2D-VQ	NAR	53M+189M	4096	18	94*
AR	3D-VQ	AR	41M+87M	1024	1024	91 (56)
MAGVIT (ours)	3D-VQ	NAR	41M+87M	1024	12	**76** (**48**)

Table 6.9: **Comparison of decoding methods** on BAIR frame prediction benchmark. The number of parameters is broken down as VQ + Transformer. NAR is non-autoregressive and AR is autoregressive. FVD and debiased FVD (in parentheses) are reported. * marks the quoted number from their paper.

substantiate the generalization performance of MAGVIT on videos from distinct visual domains and the multi-task learning recipe on large-scale data.

6.5.5 Ablation Study

Conditional MTM. We demonstrate the efficacy of COMMIT by comparing it with conventional MTM methods, including the latent masking in MaskGIT for image synthesis [33] and the commonly-used prefix condition that prepends cropped condition tokens to the input sequence.

Tab. 6.8 compares these methods on the BAIR dataset where the same 3D-VQ tokenizer is used in all approaches. As discussed in Section 6.4.1, latent masking in [33], which directly unmasks tokens of the condition region at inference time, leads to poor generalization, especially for the multi-task setup. Prefix condition produces a long sequence of variable length, making it less tractable for multi-task learning. In contrast, COMMIT yields a fixed-length sequence and better generalizability for both single- and multi-task setups.

Training losses. The bottom section of Tab. 6.8 shows the contribution of the training loss components in Eq. (6.4).

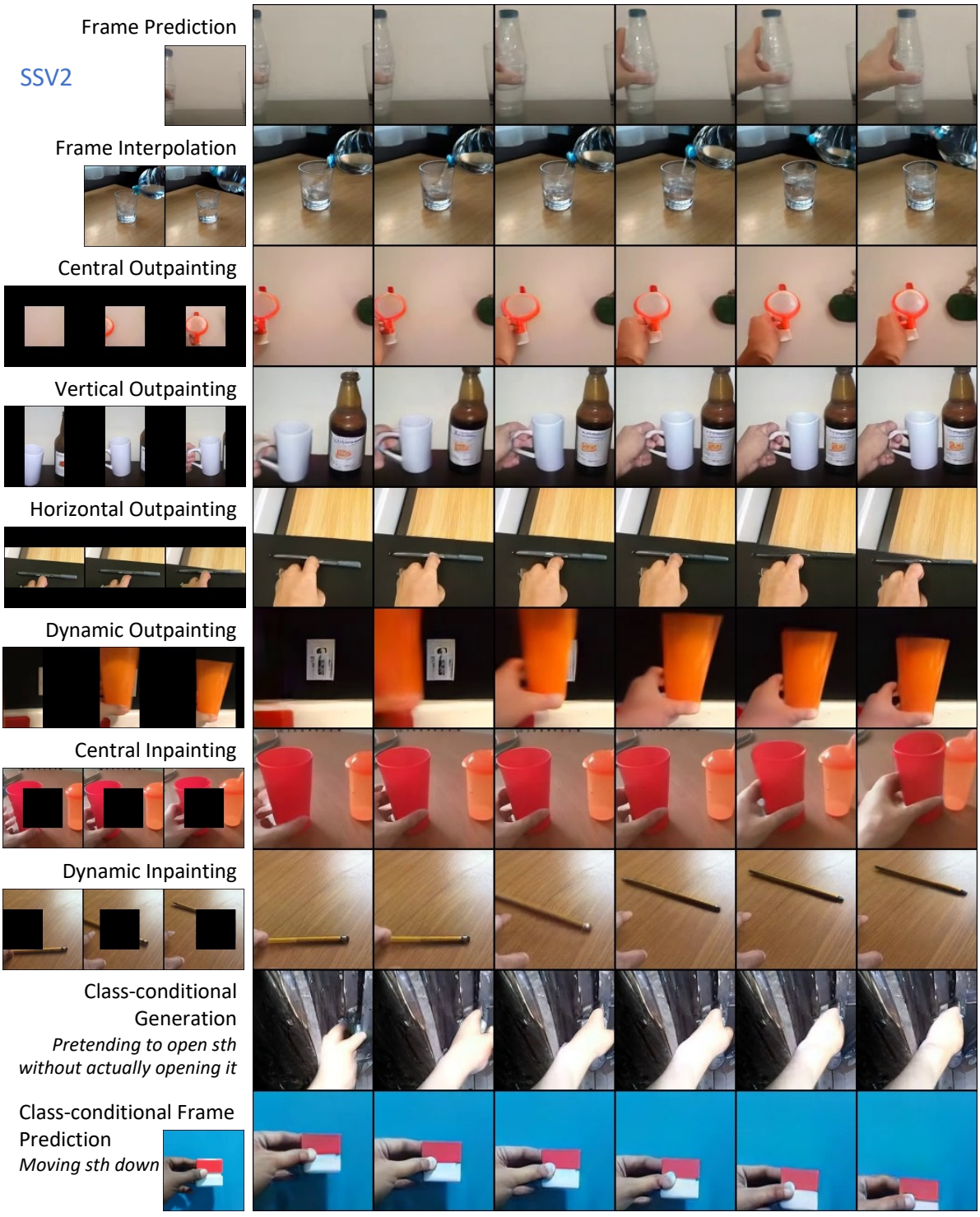


Figure 6.9: **Multi-task generation results** for the model only trained on the Something-Something-V2 dataset [68]. The conditions used to generate the shown videos are taken from the Something-Something-V2 evaluation videos.

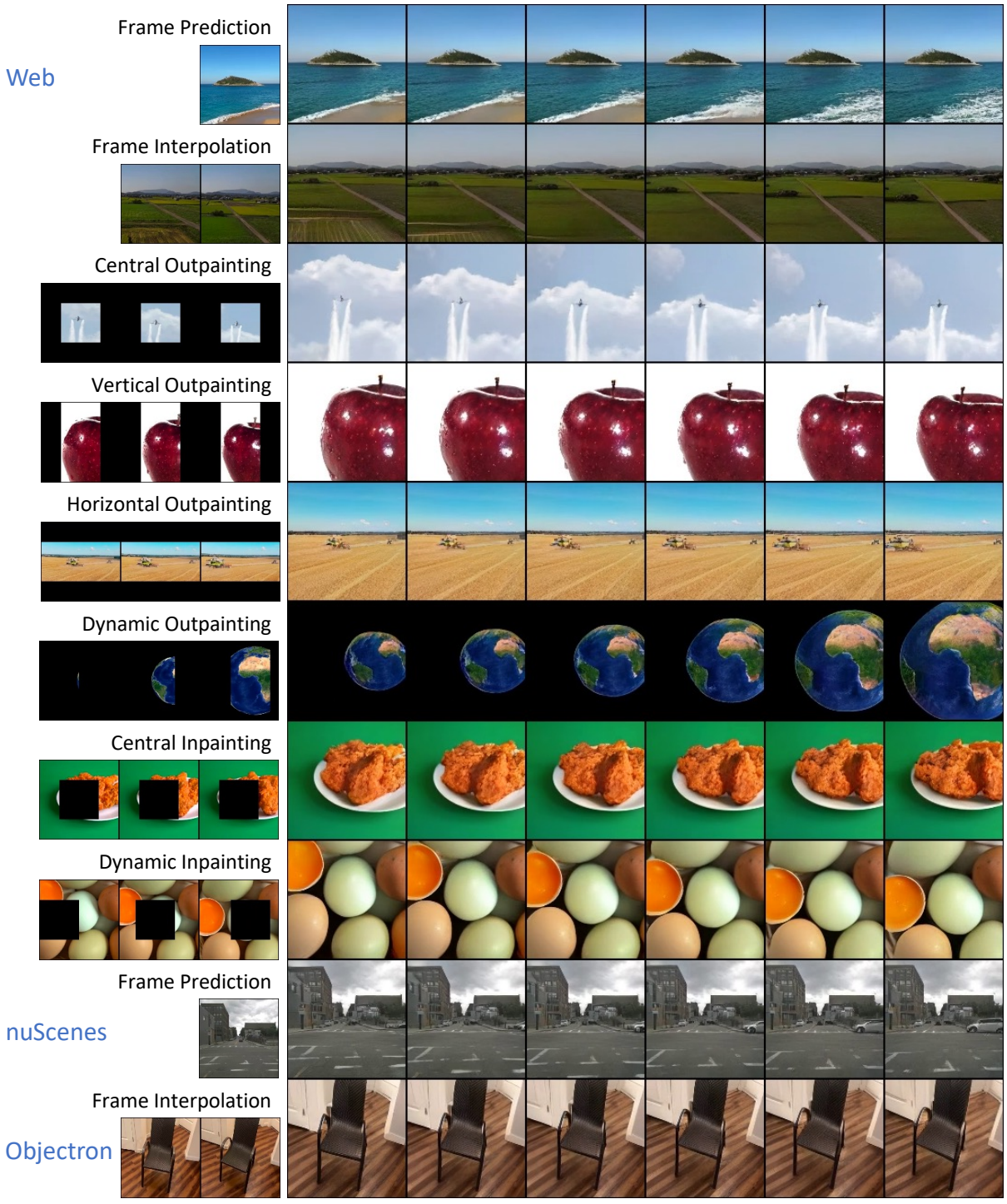


Figure 6.10: **Multi-task generation results** for three models trained on nuScenes [27], Objectron [5], and 12M Web videos, respectively. The conditions used to generate the shown videos are taken from the evaluation set.

Dataset	B	L
UCF-101	2000	2000
BAIR	400	800
BAIR-MT	1200	1600
Kinetics-600	180	360
SSv2	720	1440
nuScenes	2560	10240
Objectron	1000	2000
Web12M	10	20

Table 6.10: **Training epochs of MAGVIT transformer for each dataset**.

Decoding methods. Tab. 6.9 compares Algorithm 1 with existing autoregressive (AR) and non-autoregressive (NAR) decoding methods. We consider two NAR baselines, *i.e.*, MaskGIT [33] for image and MaskViT [74] for video synthesis. We use the same 3D-VQ tokenizer for MaskGIT, AR, and MAGVIT. As shown, the proposed decoding algorithm produces the best quality with the 3D-VQ and has a 4× shorter sequence than the 2D-VQ. While the AR transformer obtains a reasonable FVD, it takes over 85× more steps at inference time.

6.5.6 Implementation Details

Training. MAGVIT is trained in two stages where we first train the 3D-VQ tokenizer as detailed in Chapter 3 and then train the transformer with a frozen tokenizer. We follow the same learning recipe across all datasets, with the only variation in the number of training epochs. Here are the training details for the second stage:

- Sequence length: 1026.
- Hidden dropout rate: 0.1.
- Attention dropout rate: 0.1.
- Mask rate schedule: cosine.
- Peak learning rate: 10^{-4}.
- Learning rate schedule: linear warm up and cosine decay.
- Optimizer: Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.96$.
- Weight decay 0.045.
- Label smoothing: 10^{-4}.
- Max gradient norm: 1.
- Batch size: 256.
- Speed:
  1.24 steps/sec on 16 TPU-v2 chips for B,
  2.70 steps/sec on 32 TPU-v4 chips for L.

Using more hardware resources can speed up the training. We train MAGVIT models for each dataset separately. The training epochs for each dataset are listed in Tab. 6.10.

Sampling protocols. We follow the sampling protocols from previous works [42, 63] when eveluating on the standard benchmarks, *i.e.* UCF-101, BAIR, and Kinetics-600. We sample 16-frame clips from each dataset without replacement to form the real distribution in FVD and

extract condition inputs from them to feed to the model. We continuously run through all the samples required (*e.g.*, 40,000 for UCF-101) with a single data loader and compute the mean and standard deviation for 4 folds. When evaluating on other datasets, due to the lack of prior works, we adapt the above protocol based on the dataset size to ensure sample diversity.

For our MAGVIT model, we use the following COMMIT decoding hyperparameters by default: cosine schedule, 12 steps, temperature 4.5. Below are detailed setups for each dataset:

- UCF-101:
  - Dataset: 9.5K videos for training, 101 classes.
  - Number of samples: 10,000×4.
  - Resolution: 128×128.
  - Real distribution: random clips from the training videos.
- BAIR:
  - Dataset: 43K videos for training and 256 videos for evaluation.
  - Number of samples: 25,600×4.
  - Resolution: 64×64.
  - Real distribution: the first 16-frame clip from each evaluation video.
  - COMMIT decoding: exponential schedule, temperature 400.
- Kinetics-600:
  - Dataset: 384K videos for training and 29K videos for evaluation.
  - Number of samples: 50,000×4.
  - Generation resolution: 128×128.
  - Evaluation resolution: 64×64, via central crop and bilinear resize.
  - Real distribution: 6 sampled clips (2 temporal windows and 3 spatial crops) from each evaluation video.
  - COMMIT decoding: uniform schedule, temperature 7.5.
- SSv2:
  - Dataset: 169K videos for training and 24K videos for evaluation, 174 classes.
  - Number of samples: 50,000×4.
  - Resolution: 128×128.
  - Real distribution for the CG task: random clips from the training videos.
  - Real distribution for the other tasks: 2 sampled clips (2 temporal windows and central crop) from each evaluation video.
- nuScenes:
  - Dataset: 5.4K videos for training and 0.6K videos for evaluation, front camera only, 32 frames per video.
  - Number of samples: 50,000×4.
  - Resolution: 128×128.
  - Real distribution: 48 sampled clips (16 temporal windows and 3 spatial crops) from each evaluation video.
- Objectron:
  - Dataset: 14.4K videos for training and 3.6K videos for evaluation.
  - Number of samples: 50,000×4.
  - Resolution: 128×128.
  - Real distribution: 5 sampled clips (5 temporal windows and central crop) from each evalu-

ation video.
- Web12M:
  - Dataset: ~12M videos for training and 26K videos for evaluation.
  - Number of samples: 50,000×4.
  - Resolution: 128×128.
  - Real distribution: randomly sampled clips from evaluation videos.

6.6 Summary

In this chapter, we propose MAGVIT, a generic and efficient mask-based video generation model. We use the high-quality 3D-VQ tokenizer from Chapter 3 to quantize a video and design *COMMIT* for *multi-task* conditional masked token modeling. We conduct extensive experiments to demonstrate the video generation quality, efficiency, and flexibility for multi-task generation. Notably, MAGVIT establishes a new state-of-the-art quality for class conditional generation on UCF-101 and frame prediction on BAIR Robot Pushing and Kinetics-600 datasets.

Chapter 7

Generative Modality Infusion Into Frozen Large Language Models

Overview. Large Language Models (LLMs) have made significant advances in solving a wide range of NLP tasks, while expanding their capabilities beyond language into other modalities. However, little success has been made in the generation of another modality with an LLM itself. In this chapter, we adopt the SPAE representation proposed in Chapter 4 for enabling frozen LLMs to perform *multi-modal multi-task* understanding and generation involving non-linguistic modalities such as images or videos. Our approach is validated through in-context learning experiments with frozen PaLM 2 and GPT 3.5 on a diverse set of image understanding and generation tasks. Our method marks the first successful attempt to enable a frozen LLM to generate image content while surpassing state-of-the-art performance in image understanding tasks, under the same setting, by over 25%.

7.1 Motivation

Large language models (LLMs) empowered by Transformers [201] have achieved remarkable progress in addressing a broad spectrum of Natural Language Processing (NLP) tasks [7, 25, 40, 145]. With the continuous increases in model size and training data, LLMs are gradually becoming more versatile and agnostic to specific tasks, unlocking new capabilities in solving complex AI tasks [221], like question answering, code generation, reasoning, mathematics problem-solving, and understanding humor, among various other applications [7, 145].

LLMs capture rich conceptual knowledge about the world in their lexical embeddings. This raises a question: if provided with the appropriate visual representations as input, *are frozen LLMs capable of solving tasks in visual modalities?* Very recently, there have been notable advancements in extending the capabilities of frozen LLMs to tackle image understanding and retrieval tasks [112, 132]. However, generating a different modality using a frozen LLM that has not been explicitly trained on that modality has proven to be challenging and has had little success.

To facilitate LLMs for such cross-modal tasks, we propose to learn a vector quantizer to map an image, or some other non-linguistic ("foreign") modality, to the token space of a frozen LLM. This effectively translates the image into a language that the LLM can comprehend, enabling us to leverage the generative abilities of the LLM to perform image understanding and generation tasks without having to train on image-text pairs. Specifically, our new approach is that, given an image prompt, convert it to a token space with our learned encoder, use the LLM to generate suitable lexical tokens, and convert back to pixel space with our learned decoder.

We verify the plausibility of our approach in an extreme setting of in-context learning [25], without any parameter updates to the LLM. Our SPAE model is trained standalone, without backpropagating through any language model. We evaluate our approach on image understanding tasks including image classification, image captioning, and visual question answering. We showcase a promising direction to image generation with LLMs by utilizing in-context denoising techniques. Our method is LLM-agnostic and has been tested with PaLM 2 [7] and GPT-3.5 [145], suggesting compatibility with arbitrary LLMs.

The main contributions of this work are summarized as follows:

- This is the first successful method, to the best of our knowledge, that uses a frozen language model, trained solely on language tokens, to directly generate image content through in-context learning.
- We propose a new progressive prompting method that facilitates in-context generation of long cross-modal sequences.
- We evaluate our method on visual understanding and generation tasks, and notably, our approach outperforms the best-published few-shot image classification accuracy [132] by an absolute 25% under the same in-context setting.

7.2 Prior Work

Multimodal generation with LLMs. Advances have been made to expand the capabilities of LLMs beyond language. For example, Visual ChatGPT [226] uses ChatGPT to generate prompts

and executes multimodal tasks through another model, *e.g.*, generating image from text prompts by Stable Diffusion [165]. FROMAGe [112] feeds CLIP [160] embeddings to OPT [255] for image understanding and retrieval. However, it requires backpropagation through the LLM and does not support image synthesis. This work enables a standalone frozen LLM to understand and generate other modalities which are unseen in training.

Few-shot learning with LLMs. In-context learning [7, 25, 40] facilitates LLMs for few-shot learning via the text interface without parameter updates. This approach is commonly employed to assess the performance of LLMs on numerous NLP benchmarks, *e.g.*, classification and question answering [207], mathematical reasoning [124], and code generation [233], which yields competitive results to their fine-tuned counterparts. However, existing few-shot vision-language understanding and generation frameworks [6, 112] still require LLM parameter updates. In contrast, our work inherits the in-context learning ability from frozen LLMs.

7.3 Progressive In-Context Decoding with LLMs

While our method is more effective when backpropagating through LLMs by prompt [122] or adapter tuning [96, 98], this chapter focuses on verifying the plausibility in an extreme setting of in-context learning [25]. We demonstrate that LLMs are capable of performing new tasks in foreign modalities without any parameter updates. Specifically, a set of K examples $\{(\mathbf{u}^i, \mathbf{v}^i)\}_{i=1}^K$ are fed to the LLM to learn a new task and answer a query $\hat{\mathbf{u}}$ with

$$\hat{\mathbf{v}} \sim \mathbb{P}_{\text{LLM}}(\cdot \mid \hat{\mathbf{u}}; \{(\mathbf{u}^i, \mathbf{v}^i)\}_{i=1}^K). \tag{7.1}$$

Sampling $\hat{\mathbf{v}}$ by a single-pass autoregressive decoding is suboptimal due to the distributional shift in the representation and the presence of exceptionally long sequences, *e.g.*, an image is quantized into over 500 tokens. To this end, we use a progressive decoding method.

7.3.1 Method Formulation

Progressive generation. We generalize Eq. (7.1) into a multi-pass process, where the LLM learns to generate one segment of the target sequence at a time. The segment generated from the t-th pass is

$$\hat{\mathbf{v}}_t \sim \mathbb{P}_{\text{LLM}}(\cdot \mid [\hat{\mathbf{u}}, \hat{\mathbf{v}}_{<t'}]; \{([\mathbf{u}^i, \mathbf{v}^i_{<t'}], \mathbf{v}^i_t)\}_{i=1}^K), \tag{7.2}$$

where $[\cdot, \cdot]$ indicates concatenation. t' controls the length of previous segments to condition on, with two common cases: (1) a progressive autoregressive (PAR) process with $t' = t$, where each decoded segment conditions on all previously decoded ones; (2) a progressive non-autoregressive (PNAR) process with $t' = 0$ to sample each segment independently, which greatly reduces the sequence length requirement for the LLM. In practice, we use PAR to generate the first few token layers given task-specific conditions, followed by PNAR to generate the remaining token layers conditioned on the previous layers in an unconditional latent refinement process.

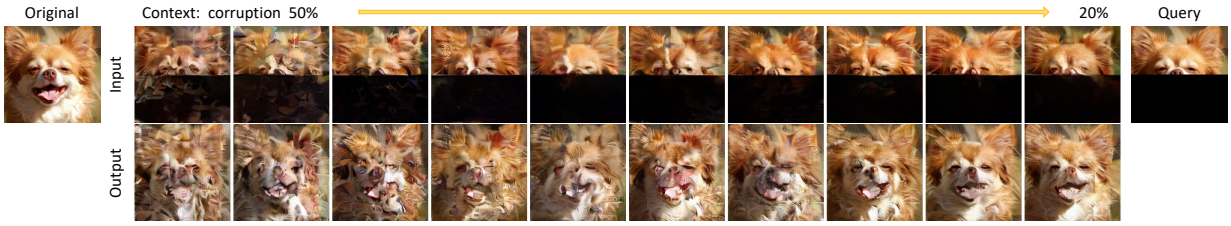


Figure 7.1: **An example of in-context denoising**. The context comprises images randomly corrupted in the token space, gradually ranging from 50% to 20%.

In-context denoising. The learning capacity of an in-context setup is far from sufficient for the entirety of a foreign modality $\mathcal{M}$. So far, there have been no successful attempts in the literature demonstrating that a frozen LLM can generate image content. For low-resolution images, LLMs can produce images directly using in-context learning, as will be demonstrated with 32×32 MNIST images [48]. For higher resolutions, the context length restricts the number of examples. For instance, a context window of 8k tokens can only hold less than a dozen 128×128 images. Therefore, we operate in a denoising subspace to synthesis beyond 32×32 resolution. Take the image-to-image task in Fig. 7.1 as an example. The provided context are images randomly corrupted in the token space by $\epsilon(\cdot; r)$, where the corruption ratio r follows a cosine schedule [33].

$$(\mathbf{u}^i, \mathbf{v}^i) \sim \Big(\epsilon(\mathcal{T}(\texttt{mask}(\mathbf{I})); r_i), \epsilon(\mathcal{T}(\mathbf{I}); r_i)\Big), \mathbf{I} \in \mathcal{M} \tag{7.3}$$

where $\mathcal{T}(\cdot)$ represents the SPAE tokenizer and $\mathcal{M}$ is a small set of raw images. $\texttt{mask}(\cdot)$ zeros out pixels of the real image to create the condition image, such as masking out the bottom half for out-painting. The query $\hat{\mathbf{u}}$ is always sampled from $\mathcal{M}$ without noise ϵ. To ensure the generation is not simply copying the context, we enforce a minimal corruption rate of 20% such that no identical image from the context matches the real target image.

7.3.2 LLM Prompting

To generate prompts, we utilize SPAE from Chapter 4 to quantize an image, or another non-linguistic modality, into a pyramid of lexical tokens. Subsequently, we flatten the tokens by concatenating them layer-by-layer, following a raster scan, and resulting in a 1-D string. This string, representing the image, is referred to as the ***SPAE string*** in the following prompts.

We use task-specific prompt templates to facilitate answer generation with LLMs. The LLM output is always parsed by removing leading and trailing whitespace or newline characters.

Image classification with GPT 3.5. We use the same prompt template as LQAE [132] to interact with GPT 3.5. For a 2-way 1-shot classification between class *lion* and *vase*, the prompt is

```
For each of the following input output pairs, output is one of
['lion', 'vase']
###
```

```
Input: <SPAE string from a lion image>
Output: lion
###
Input: <SPAE string from a vase image>
Output: vase
###
Input: <SPAE string from the query image>
Output:
```

We use greedy decoding to get a maximum of 7 tokens from GPT 3.5.

Image classification with PaLM 2. We use the original miniImageNet [204] format with PaLM 2. The prompt looks like

```
Answer with "lion" or "vase".

<SPAE string from a lion image>
This is a lion


<SPAE string from a vase image>
This is a vase


<SPAE string from the query image>
What is this?  # Only used in 5-way 3/5-shot setups
This is a
```

We use greedy decoding to get a maximum of 4 tokens from PaLM 2.

Image captioning. We use greedy decoding to get a maximum of 20 tokens before the first newline character with the following prompt:

```
Generate a caption sentence based on words describing an image.

Q: <SPAE string from image 1>
A: <Caption for image 1>

Q: <SPAE string from image 2>
A: <Caption for image 2>

Q: <SPAE string from the query image>
A:
```

Visual question answering. We use greedy decoding to get a maximum of 4 tokens before the first newline character with the prompt template as

```
Answer with a single word.
```

```
C: <SPAE string from image 1>
Q: <Question for image 1>
A: <Answer for image 1>

C: <SPAE string from image 2>
Q: <Question for image 2>
A: <Answer for image 2>

C: <SPAE string from the query image>
Q: <Question for the query image>
A:
```

Image/video generation with AR decoding. For image or video generation tasks, the condition can be a text string or an SPAE string of a condition image. Suppose we use AR decoding with a stride of 4 tokens. At the 4th step, the prompt looks like

```
Learn a new language and predict the 4 tokens following the examples.

C:<condition for image 1>
Q:<SPAE string (token 1-12) for image 1>
A:<SPAE string (token 13-16) for image 1>

C:<condition for image 2>
Q:<SPAE string (token 1-12) for image 2>
A:<SPAE string (token 13-16) for image 2>

C:<condition for the query>
Q:<SPAE string (token 1-12) for the generated image from previous steps>
A:
```

We use PaLM 2 to generate 8 predicted sequences for the next 4 tokens, starting with a temperature $T_0 = 0$. We use the sentence piece [115] tokenizer to tokenize the output string. If all predictions are shorter than 4 tokens, we retry the LLM prediction with a higher temperature. At the i-th retry, the temperature is given by

$$T_i = \psi \sum_{j=1}^{i} 2^j \tag{7.4}$$

where $\psi = 0.01$ is used.

Image/video generation with NAR decoding. We use NAR decoding to generate SPAE layer 6 conditioned on layer 1-5. With a stride of 16, the prompt at the 3rd step looks like

```
Predict the outputs following the examples.

Q:<SPAE string from layer 1-5 for image 1>
A:<SPAE string from layer 6 (token 33-48) for image 1>
```

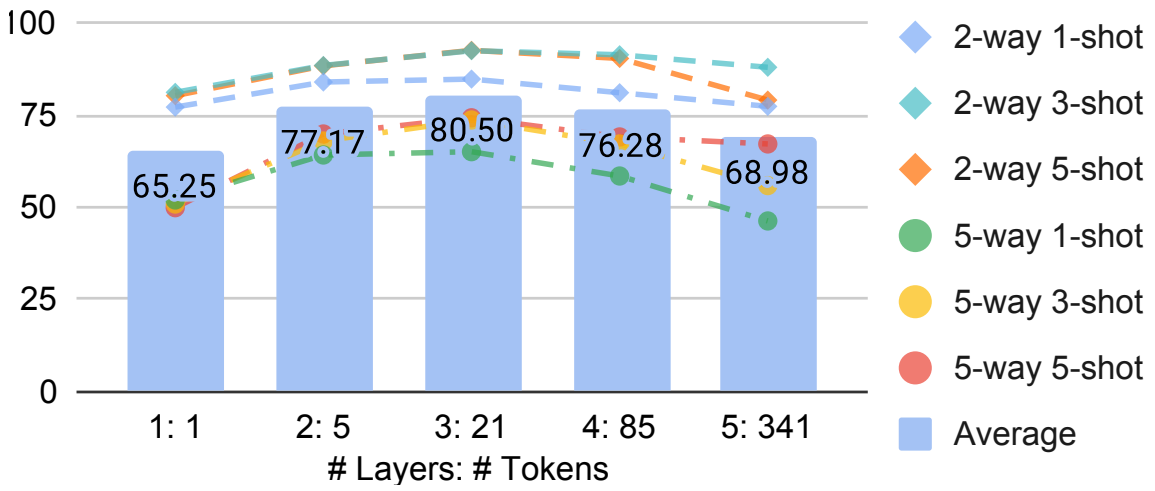


Figure 7.2: **Few-shot classification accuracy** on mini-ImageNet using different $\text{SPAE}_{\text{PaLM}}$ layers.

```
Q:<SPAE string from layer 1-5 for image 2>
A:<SPAE string from layer 6 (token 33-48) for image 2>

Q:<SPAE string from layer 1-5 for the generated image from AR decoding>
A:
```

We use PaLM 2 to generate 8 predicted sequences for the next 16 tokens. If the sentence piece parsing fails, we retry with the same temperature schedule as in AR decoding.

7.4 Experimental Results

LLM setups. The PaLM 2-L API [7] is used for in-context learning with $\text{SPAE}_{\text{PaLM}}$. For a fair comparison with prior works [132], we use SPAE_{GPT} with the GPT 3.5 `text-davinci-003` API (`https://platform.openai.com/docs/models/gpt-3-5`).

MNIST SPAE. In addition to the models introduced in Chapter 4, we train another SPAE on the MNIST [48] dataset with the same architecture setup. We pad the handwritten digit images from 28×28 to 32×32 pixels, which are then encoded into 4×4 embeddings. Each image is represented by 37 tokens organized in four layers, with sizes of 1×1, 2×2, 4×4, and 4×4. We replace the CLIP image embedding with the CLIP text embedding of the label for the semantic loss. The model is trained for 10k steps with a batch size of 256. For in-context generation, AR decoding with a stride of 4 is used to produce all 37 tokens.

7.4.1 Main Evaluation

Few-shot image classification. We evaluate the in-context image understanding capability with a frozen LLM on the mini-ImageNet [204] few-shot classification benchmark. A set of

Method	# Layers : # Tokens	Inner Shots	Task Induction		✓	✓	✓	✓	✓	✓	Avg
		Repeats	1	1	3	5	1	1	1	
			0	0	0	0	1	3	5	
2-Way Classification											
Frozen [195]	-		1.7	33.7	66	66	63	65	63.7	51.3
LQAE [132]	1: 256	GPT 3.5	1.5	35.2	68.2	69.8	68.5	68.7	65.9	53.97
SPAE$_{\text{GPT}}$ (ours)	2: 5	GPT 3.5	*5.3*	*77.2*	*84.4*	*86.0*	*79.4*	*77.2*	*77.1*	*69.51*
SPAE$_{\text{PaLM}}$ (ours)	1: 1	PaLM 2	**34.8**	77.2	81.2	80.3	74.0	73.2	71.5	70.31
SPAE$_{\text{PaLM}}$ (ours)	2: 5	PaLM 2	32.2	84.0	88.5	88.4	**85.1**	83.6	82.4	77.74
SPAE$_{\text{PaLM}}$ (ours)	3: 21	PaLM 2	27.9	**84.8**	**92.5**	**92.6**	84.8	**85.2**	**85.4**	**79.03**
SPAE$_{\text{PaLM}}$ (ours)	4: 85	PaLM 2	22.8	81.1	91.4	90.4	82.6	84.3	84.7	76.76
SPAE$_{\text{PaLM}}$ (ours)	5: 341	PaLM 2	21.2	77.4	88.0	79.1	84.8	74.0	76.1	71.51
SPAE$_{\text{PaLM}}$ (ours)	6: 597	PaLM 2	21.8	73.8	70.8	62.4	64.8	62.1	58.6	59.19
SPAE$^{\text{disjoint}}_{\text{PaLM}}$	2: 5	PaLM 2	24.8	79.8	84.5	83.7	80.8	78.5	78.4	72.93
SPAE$^{\text{disjoint}}_{\text{PaLM}}$	3: 21	PaLM 2	21.4	81.4	89.2	87.9	82.6	81.7	80.6	74.98
5-Way Classification											
Frozen [195]	-		0.9	14.5	34.7	33.8	33.8	33.3	32.8	26.26
LQAE [132]	1: 256	GPT 3.5	1	15.7	35.9	36.5	31.9	36.4	45.9	29.04
SPAE$_{\text{GPT}}$ (ours)	2: 5	GPT 3.5	*4.3*	*63.0*	*63.4*	*60.6*	*61.9*	*62.1*	*62.1*	*53.91*
SPAE$_{\text{PaLM}}$ (ours)	1: 1	PaLM 2	**26.8**	52.0	50.9	49.9	51.9	48.4	47.9	46.83
SPAE$_{\text{PaLM}}$ (ours)	2: 5	PaLM 2	23.6	64.2	68.0	69.9	63.4	62.0	60.2	58.76
SPAE$_{\text{PaLM}}$ (ours)	3: 21	PaLM 2	20.2	**65.1**	**73.7**	**74.3**	**66.4**	**67.0**	66.3	**61.86**
SPAE$_{\text{PaLM}}$ (ours)	4: 85	PaLM 2	16.1	58.5	67.2	69.1	64.0	66.4	**67.4**	58.39
SPAE$_{\text{PaLM}}$ (ours)	5: 341	PaLM 2	12.1	46.3	55.9	67.2	43.3	46.3	-	-
SPAE$_{\text{PaLM}}$ (ours)	6: 597	PaLM 2	12.1	35.7	-	-	-	-	-	-

Table 7.1: **Few-shot classification accuracy** on the mini-ImageNet benchmarks. SPAE$_{\text{GPT}}$ and SPAE$_{\text{PaLM}}$ are trained using different vocabularies and embedding sources, with different prompt templates for in-context learning. They show the broad compatibility of SPAE but are not for a comparison between the LLMs. The best performance with GPT is in italics while the overall best is in bold. - means value unavailable due to an infeasible sequence length.

Method	Inner Shots	1	3	5
Frozen [195]		7.8	10.1	10.5
SPAE$_{\text{PaLM}}$ (ours)	PaLM 2	**14.3**	**15.9**	**15.1**

Table 7.2: **Few-shot VQA performance on Real-Fast-VQA**.

tokenized images and class labels are fed to the language model as context for classification of a new image. Following [132, 195], we evaluate 14 settings controlled by four factors regarding the content of each test case: (1) task induction: whether including a preamble to specify the output space; (2) number of ways: the number of categories; (3) number of inner shots: the number of unique examples for each category; (4) number of repeats: the number of times that each unique example is repeated.

We compare SPAE with the state-of-the-art methods Frozen [195] and LQAE [132]. As shown in Tab. 7.1, SPAE_{GPT} consistently outperforms LQAE, both using the same GPT 3.5 model and in-context format, while using only 2% of its tokens. Fig. 7.2 shows the performance trend when using different number of $\text{SPAE}_{\text{PaLM}}$ layers across six settings with task induction and 0 repeats. $\text{SPAE}_{\text{PaLM}}$ with 3 layers achieves the best performance which balances between sufficient semantics and an image sequence length that is optimal for LLM in-context learning. Overall, $\text{SPAE}_{\text{PaLM}}$ yields +25% and +32% average accuracy improvement over the state-of-the-art on the 2-way and 5-way benchmarks in Tab. 7.1.

Few-shot visual question answering. Tab. 7.2 provides quantitative results on the visual question answering (VQA) task. We compare with the baseline Frozen [195] method on the Real-Fast-VQA [195] benchmark for few-shot learning. As shown, SPAE consistently outperforms Frozen. Unlike Frozen, SPAE training does not require backpropagation through the LLM.

7.4.2 Qualitative Studies

This section explores the capability of a frozen PaLM 2, trained solely on language tokens, in performing multimodal tasks using in-context learning. We adopt a two-stage decoding process for image generation. In stage one, we use AR decoding to produce the first 5 SPAE layers with task-specific conditions. Stage two is a task-agnostic NAR decoding process for layer 6 conditioned on the first 5 layers.

Image to text and VQA. We examine two tasks involving visual-text reasoning (1) image captioning on COCO [131] captions; and (2) visual question answering (VQA) on COCO-QA [162]. For both tasks, we provide 10 unique training examples as prompts. In the case of VQA, 10 different answers are presented to form a 10-way 1-shot setup.

We compare SPAE to a baseline model trained with the same frozen language codebook but without the proposed semantic guidance or pyramid SAQ. As shown in Fig. 7.3, when fed with baseline tokens, the LLM randomly hallucinates a caption or guesses an answer simply based on the question. Similar hallucination can happen if we only use the first two layers of SPAE or five words to represent an image, as it provides insufficient context for captioning. Reasonable captions start to appear with 4 layers or 85 words representing an image, while complex scenes may still need the full 6 layers of 597 words.

LLM generating MNIST images. Fig. 7.4 shows a few image generation examples on the MNIST [48] dataset. The frozen LLM learns about handwritten digit images through 50 context samples tokenized by SPAE trained on MNIST. Each sample consists of a preamble `"an`

Baseline: A man in a suit standing in front of a white wall.
SPAE L1: A man in a red jacket and black pants standing on a snowy mountain.
SPAE L2: A man in a red jacket skiing down a snowy mountain.
SPAE L3: A man skiing down a snowy mountain.
SPAE L4: A person skiing down a snowy mountain.
SPAE L5: A person skiing down a mountain.
SPAE L6: A person skiing down a mountain.

Baseline: A group of people are standing in a field.
SPAE L1: A group of people are standing in a room.
SPAE L2: A kitchen with a stove, sink, and refrigerator.
SPAE L3: A kitchen with a stove, sink, and refrigerator.
SPAE L4: A kitchen with a stove, sink, and refrigerator.
SPAE L5: A kitchen with a stove, sink, and cabinets.
SPAE L6: A kitchen with a sink, stove, and refrigerator.

Baseline: A man is standing on a rock in the middle of a river.
SPAE L1: A man is standing on a rock in the middle of a river.
SPAE L2: A man is wearing a coat and a hat.
SPAE L3: A man is holding a small dog.
SPAE L4: A teddy bear is sitting on a bed.
SPAE L5: A teddy bear is sitting on a bed.
SPAE L6: A teddy bear is sitting on a bed.

Baseline: A man and a woman are sitting on a bench in a park.
SPAE L1: A man is holding a baby in his arms.
SPAE L2: A group of people are standing in a line.
SPAE L3: A group of people in costumes at a Halloween party.
SPAE L4: A group of people are dressed up in costumes for Halloween.
SPAE L5: a group of people dressed in costumes at a party
SPAE L6: a table with a bowl of fruit and a vase of flowers

SPAE: A pizza with pepperoni and cheese on a white plate.

SPAE: A man in a suit and tie standing next to a woman in a wedding dress.

SPAE: A train is stopped at a station.

Q: what is the young boy riding in the empty parking lot
A: Baseline: bike
SPAE: skateboard

Q: what bear walking through tall grass
A: Baseline: siberian
SPAE: grizzly

Q: how many different wines are lined up in glasses on an outdoor table
A: SPAE: 5

Q: how many computer screens are displayed with one image
A: SPAE: 3

Figure 7.3: **Qualitative samples of image-to-text generation**: image captioning and VQA. We compare between different layers of SPAE (L1-L6) and a baseline model without semantic guidance or pyramid SAQ.

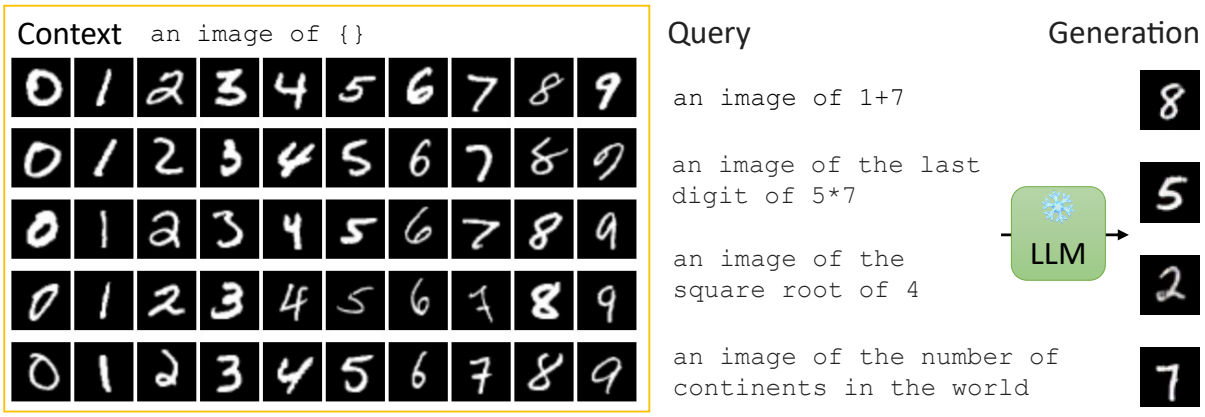


Figure 7.4: **Examples of text-to-image generation on MNIST using the frozen PaLM 2 model**. We use SPAE to tokenize 50 handwritten images as the context and ask PaLM 2, an LLM trained solely on text tokens, to answer complex queries that require generating digit images through SPAE as the output. To achieve this, we use SPAE to convert the image into lexical tokens and construct prompts. Then, we ask PaLM 2 to generate suitable answers for the prompts. Finally, we convert the answer tokens back into the pixel space and display them in the figure. Note that the generated digit images do not appear identical to any of the samples provided in the context.

`image of` k" and the lexical tokens representing an image of digit k. Then we can ask the LLM to answer questions with digit images. Specifically, with a query of "`an image of` `1+7`", we can use progressive AR decoding with the LLM to produce a token sequence that can be decoded into an image of 8 by SPAE. We test with complex questions requiring mathematical reasoning or common sense knowledge, and the LLM is able to respond correctly. In addition, the generated digit images appear different from all context samples. This demonstrates the cross-modal reasoning capability enabled by SPAE and a frozen LLM, with images generated over the text-only interface.

Conditional image interpolation. To the best of our knowledge, there have been no successful attempts that demonstrate generic image generation capability using a frozen LLM. To this end, we define a very simple setup to explore the interpolation capability of LLM, where the conditions are integers from 1 to 9. The target images are created with different pixel-space transformations:

- Brightness: $[\pm 0.8, \pm 0.6, \pm 0.4, \pm 0.2]$.
- Contrast: $[\pm 0.8, \pm 0.6, \pm 0.4, \pm 0.2]$.
- Saturation: $[\pm 0.4, \pm 0.3, \pm 0.2, \pm 0.1]$.
- Color (RGB): $[(0.6, 1.4, 1), (0.7, 1.3, 1), (0.8, 1.2, 1), (0.9, 1.1, 1), (1.1, 0.9, 1), (1.2, 0.8, 1), (1.3, 0.7, 1), (1.4, 0.6, 1)]$

Overflow pixels are clipped to $[0, 255]$. As shown in Fig. 7.6, images 1-4 and 6-9 are fed as context to produce image 5, where the model interpolates the variable property. Fig. 7.5 shows generated samples at 256×256 resolution under the same setup.

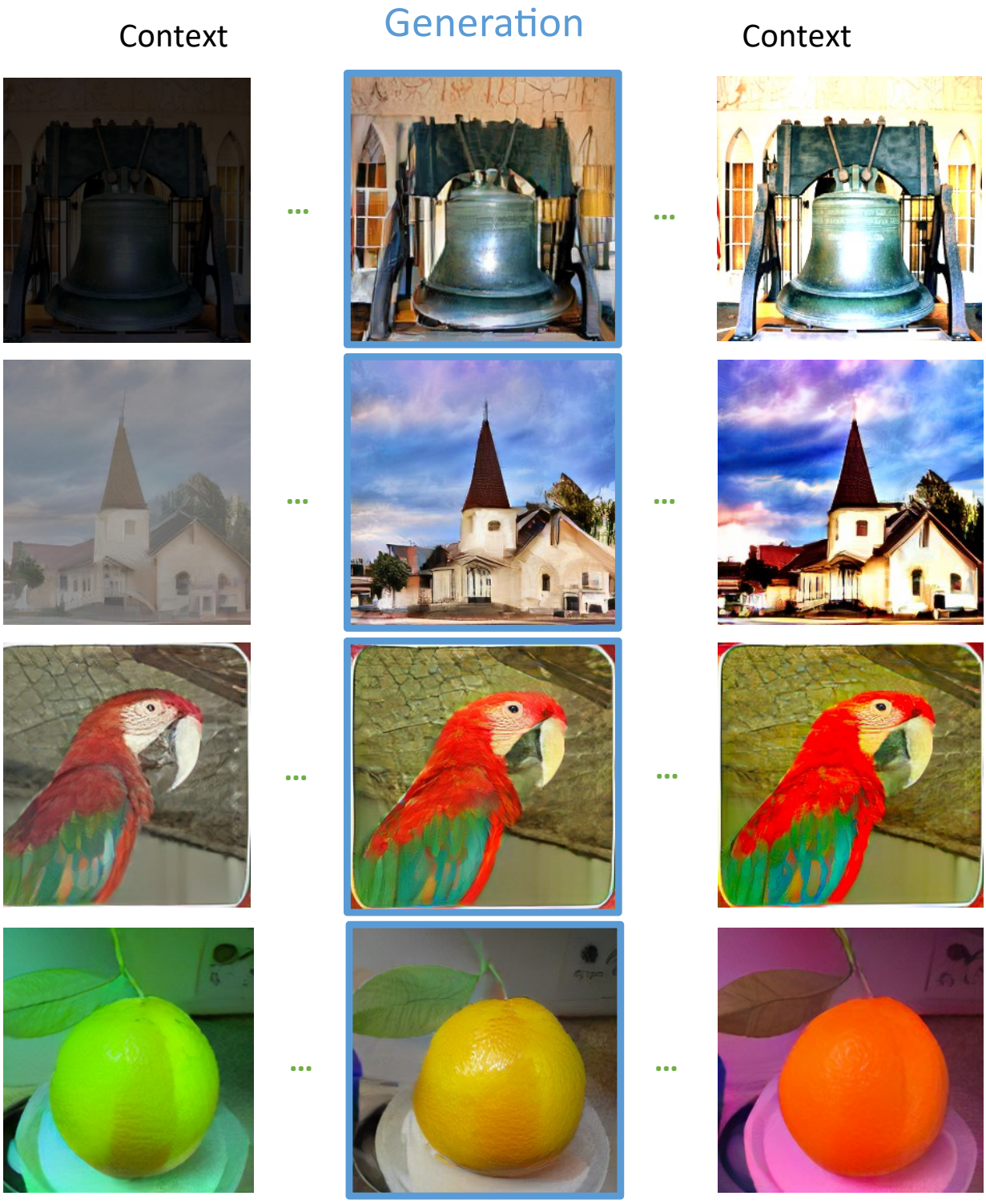


Figure 7.5: **Examples of conditional image interpolation** at 256x256 resolution. The LLM is provided with eight condition images for the interpolation following the setup in Fig. 7.6.

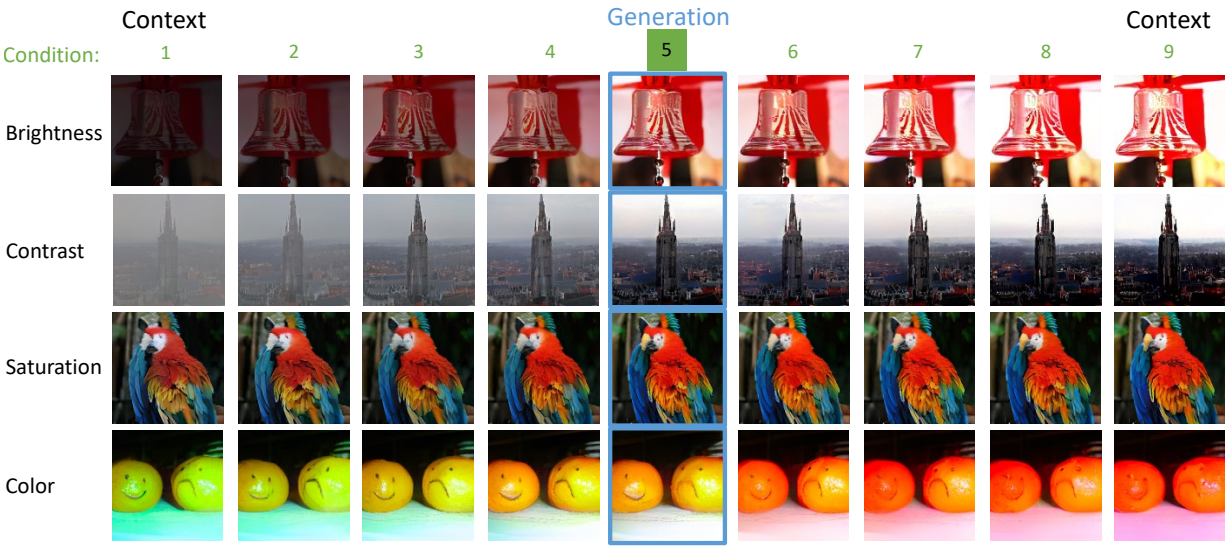


Figure 7.6: **Examples of conditional image interpolation** of different image transformations.

Conditional image denoising. We also define a denoising setup to explore the capability of LLMs. The conditional image denoising tasks include image outpainting, deblur, inpainting, location translation, rotation, *etc.* Note that, in order to generate images for each task, we utilize 10 pairs of noisy examples with corruption rates ranging from 50% to 20%, as discussed in Section 7.3.1. We use PAR decoding to produce the first 5 token layers with task-specific conditions, followed by task-agnostic PNAR decoding to fill in layer 6.

The bottom rows of Fig. 7.7 compare the output quality from different decoding strides with the same set of context examples. Single-step decoding with infinity stride fails to produce a reasonable image, which validates the proposed progressive generation approach. In Fig. 7.8, we qualitatively compare SPAE with baseline methods VQGAN and LQAE using the same in-context denoising procedure. As shown, VQGAN fails to produce reasonable images, in part because many words in the LLM output are out of its vocabulary. LQAE only produces vague object contours but cannot recover any visual details. On the contrary, SPAE can generate reasonable images. Fig. 7.9 visualizes the input pairs for the conditional image denoising samples in Fig. 7.8, with more examples in Fig. 7.10.

Figure 7.7: **Examples of conditional image denoising**. We compare different decoding strides for both stages. Yellow and blue boxes indicate the selected results. The LLM is provided with ten pairs of noisy examples shown at the top half. Multiple different outputs can be obtained from the same set of context samples.

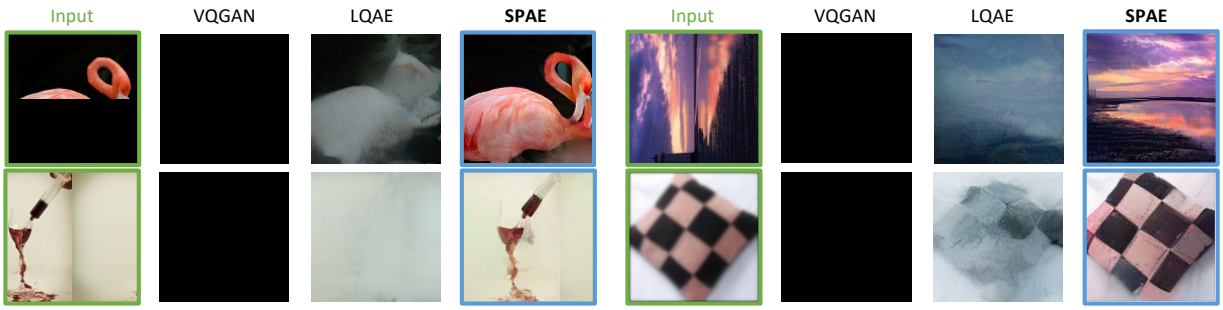
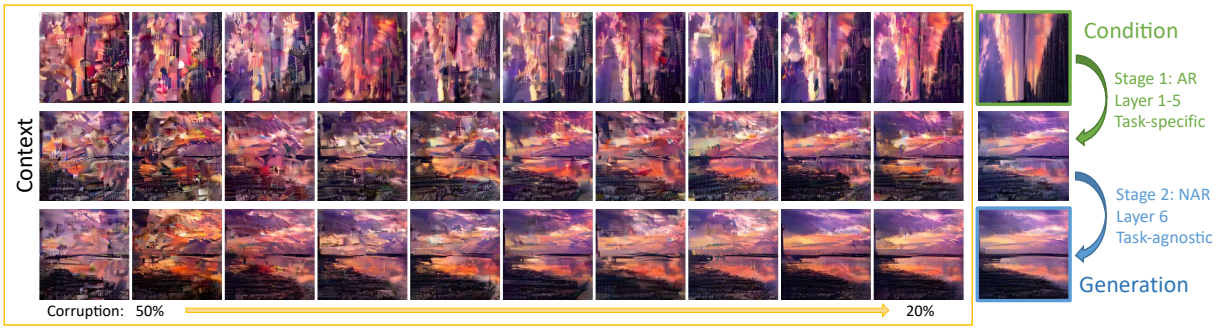
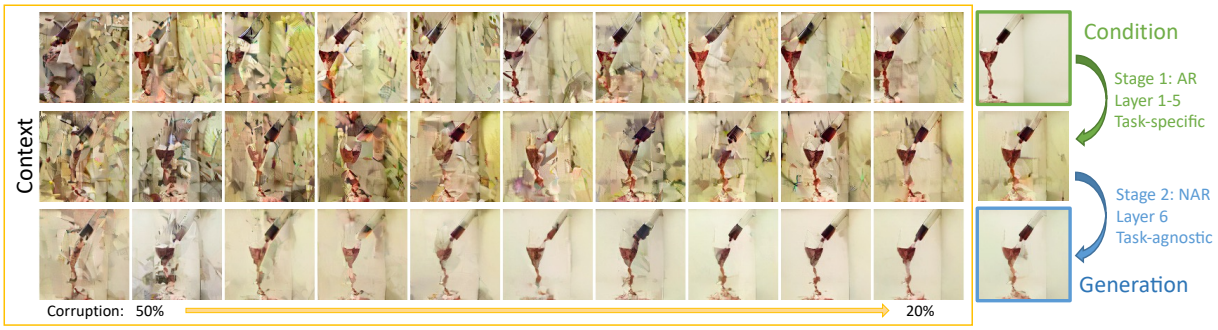


Figure 7.8: **Comparison on conditional image denoising with different tokenizers**. All models use the same decoding setup with the same ten pairs of prompt images from Figs. 7.7 and 7.9.



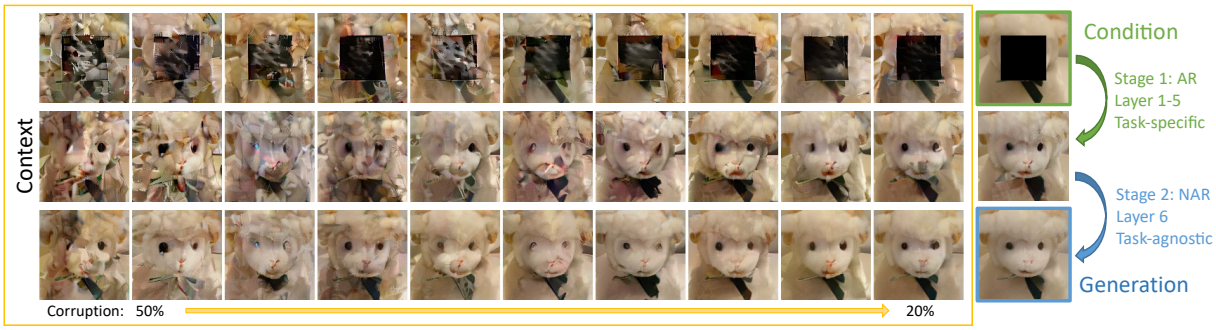
(a) Clockwise rotation by 90 degrees.



(b) Spatial translation to the right.

(c) Deblur from a gaussian filter.

Figure 7.9: **Examples of conditional image denoising**. All input samples for the in-context learning are presented for the examples in Fig. 7.8.



(a) Inpainting the center region.

(b) Inpainting the center region.

(c) Outpainting the bottom half.

Figure 7.10: **Examples of conditional image denoising**.

Figure 7.11: **Examples of multi-modal outputs** from the LLM.

Multimodal outputs. Fig. 7.11 shows a task requiring a single LLM to output both image and text, where it first inpaints the center region of an image using in-context denoising and then creates multiple captions for the output image.

Image-to-video denoising. Fig. 7.12 shows an image-to-video example with the frame prediction task using progressive in-context denoising. The input is one frame tokenized by the image SPAE, while the output is a 16-frame clip tokenized by the video SPAE. We follow the same two-stage procedure as conditional image denoising, with more steps in each stage to account for the longer sequence. Due to the sequence length limit, only four samples can be fit into the context, which limits LLM's performance for this task.

7.5 Summary

Our work unveils the untapped potential of frozen Large Language Models (LLMs) in tackling *multi-modal multi-task* understanding and generation involving images and videos, without requiring explicit training on these modalities. This is achieved with the new method, SPAE from Chapter 4, which converts between visual content and lexical tokens of variable length, imbued with rich semantic meaning. Our findings show the great potential of harnessing the vast knowledge and reasoning capabilities of LLMs in the field of computer vision, transcending the limitations of language-only tasks.

Limitations. The capability of in-context learning is significantly constrained by the acceptable sequence length. Although our results suggest the plausibility of image generation, the quality and diversity is still far from the recent text-to-image models trained on paired image and text data.

Broader impact. We showcase the untapped potential of frozen Large Language Models (LLMs) in multimodal understanding and generation tasks involving images and videos, with-

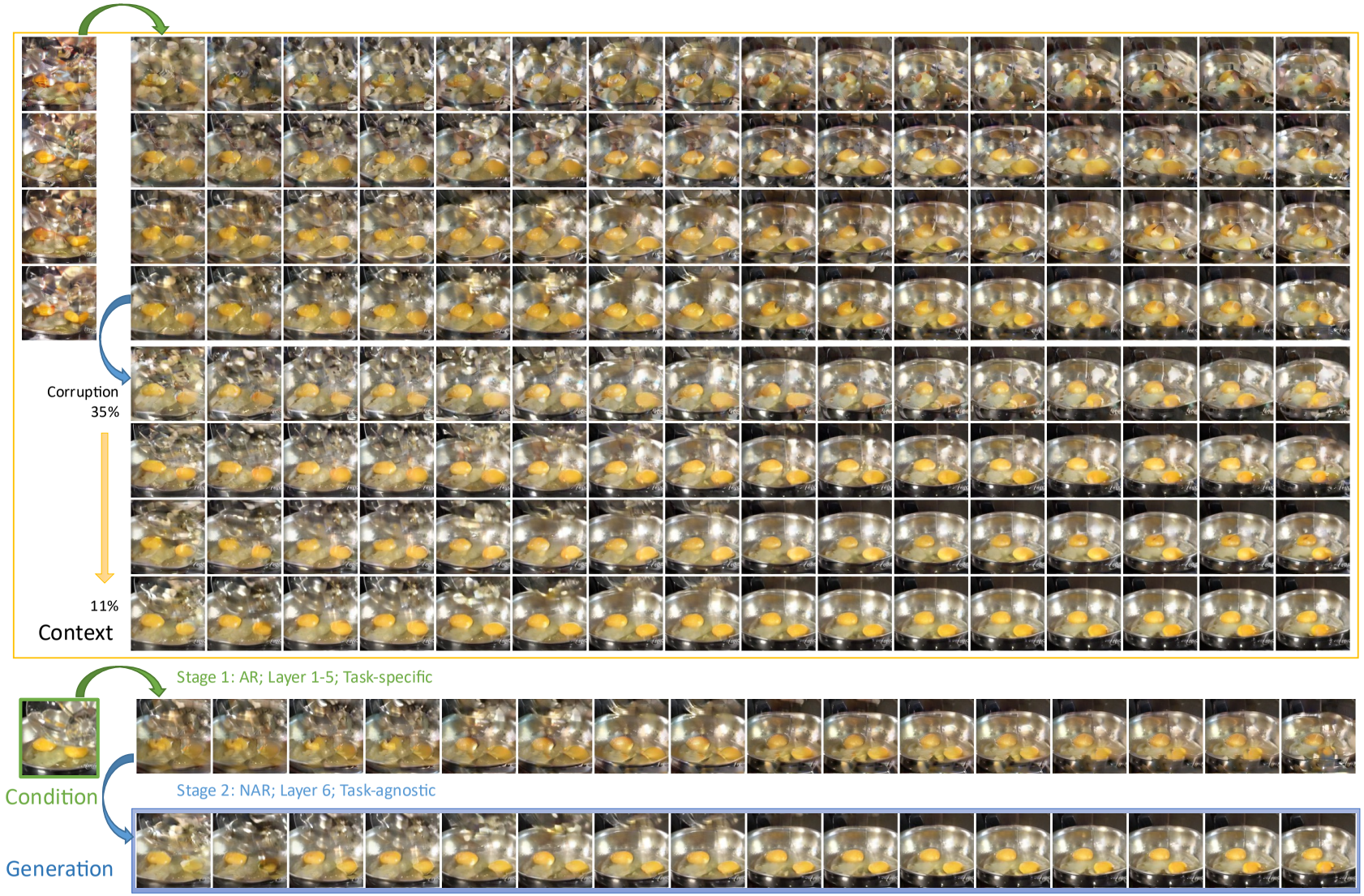


Figure 7.12: **Examples of image-to-video denoising**: frame prediction. We follow the same two-stage generation procedure as in conditional image denoising tasks. Due to the sequence length limit, only four samples can be fit into the context. The generated video clip appear visually different from the context samples, especially around the reflections of the bowl.

out requiring explicit training on these modalities. As an initial research proof-of-concept, we focus on in-context learning, which has limitations in learning context and constrained capabilities. Consequently, there is still a substantial gap to the recent specialized models for text-to-image (*e.g.*, Stable Diffusion) or image-to-text that have been specifically trained using billions of text-image pairs.

The potential impact of our research lies in its influence on future studies, specifically in the area of interacting with pretrained LLMs to enhance their understanding and generation capabilities in the visual modality. For instance, our work can be extended to explore finetuning or adapter tuning of LLMs on large-scale text-image datasets. Future research in these directions may implicate ethical issues around fairness and transparency, which need to be carefully considered beyond the quality measurements employed here. We have found that the generated tokens occasionally include slang terms or words that create inappropriate connotations related to the subject depicted in the image or video. Such concerns must be thoroughly considered and effectively addressed prior to deploying this method in real-world applications.

Chapter 8

Scalable Generative Multi-Modal Transformer

Overview. Leveraging the representation we have built in Chapter 5, we present VideoPoet, a model for synthesizing high-quality videos from a large variety of conditioning signals. VideoPoet employs a decoder-only transformer architecture that enables *multi-task* generation of *multi-modal* outputs – including videos, images, and audio. The training protocol follows that of Large Language Models (LLMs), consisting of two stages: Pretraining and task-specific adaptation. During pretraining, VideoPoet incorporates a mixture of multi-modal generative objectives within an autoregressive Transformer framework. The pretrained LLM serves as a foundation that is adapted to a range of video generation tasks. We present results demonstrating the model's state-of-the-art capabilities in zero-shot video generation, specifically highlighting the generation of high-fidelity motions. Project page: `http://sites.research.google/videopoet/`.

8.1 Motivation

Recently, there has been a surge of generative video models capable of a variety of video creation tasks. These include text-to-video [177, 251], image-to-video [248], video-to-video stylization [32, 38, 205], and video editing [31, 65, 212] among other video applications. Most existing models employ diffusion-based methods for video generation. These video models typically start with a pretrained image model, such as Stable Diffusion [149, 165], that produces high-fidelity images for individual frames, and then fine-tune the model to improve temporal consistency across video frames.

While Large Language Models (LLMs) are commonly used as foundation models across various modalities including language [25], code [128, 145], audio [167], speech [3], and robotics [53, 262], the diffusion model remains the predominant approach for video generation. Although early research has demonstrated the effectiveness of LLMs in text-to-image generation, *e.g.*, DALL-E [161], Parti [235] and CogView [51], and text-to-video, *e.g.*, CogVideo [93], language models have not reached a level of quality on par with video diffusion models in tasks like text-to-video generation as shown in previous studies [143, 203]. In contrast to training exclusively for text-to-video tasks, the generative model of LLMs in the language domain emphasizes a large pretraining stage to learn a foundation [17] by examining pretraining tasks that extend beyond text-to-video generation.

A notable advantage of employing LLMs in video generation lies in the ease of integrating existing LLM frameworks. This integration allows for reusing LLM infrastructure and leverages the optimizations our community has developed over many years for LLMs, including optimizations in learning recipes for model scaling [25, 40], training and inference infrastructure [54], hardware, among other advancements. This couples with their flexibility in encoding many diverse tasks in the same model [157], which stands in contrast to most diffusion models where architectural changes and adapter modules are the dominant approach used to adapt the model to more diverse tasks [252].

In this chapter, we exploit language models for video generation, following the canonical training protocols of LLMs in the language domain. We introduce *VideoPoet*, a language model for video generation. VideoPoet employs a decoder-only LLM architecture [7, 145] that admits image, video, and audio modalities as discrete tokens, each produced by their respective tokenizer.

The training process of VideoPoet consists of two stages: (1) pretraining and (2) task-adaptation. During pretraining, VideoPoet incorporates a mixture of multimodal pretraining objectives within an autoregressive transformer framework. After pretraining, the model functions as a versatile multi-task video generation model such as text-to-video, image-to-video, video editing and video-to-video stylization. These capabilities are inherently integrated into a single LLM, rather than relying on a separate generative model controlled by text prompts [188]. During subsequent task-adaptation, the pretrained model can be further fine-tuned either to enhance its generation quality on the training tasks or to perform new tasks.

Experiments show VideoPoet's state-of-the-art capabilities in generating videos with large and high-fidelity motions. Through the powerful capabilities of the transformer architecture, VideoPoet can be straightforwardly trained on a multi-task, multimodal generative objective, allowing for generating consistent and realistic motion driven by text or other prompts. Fur-

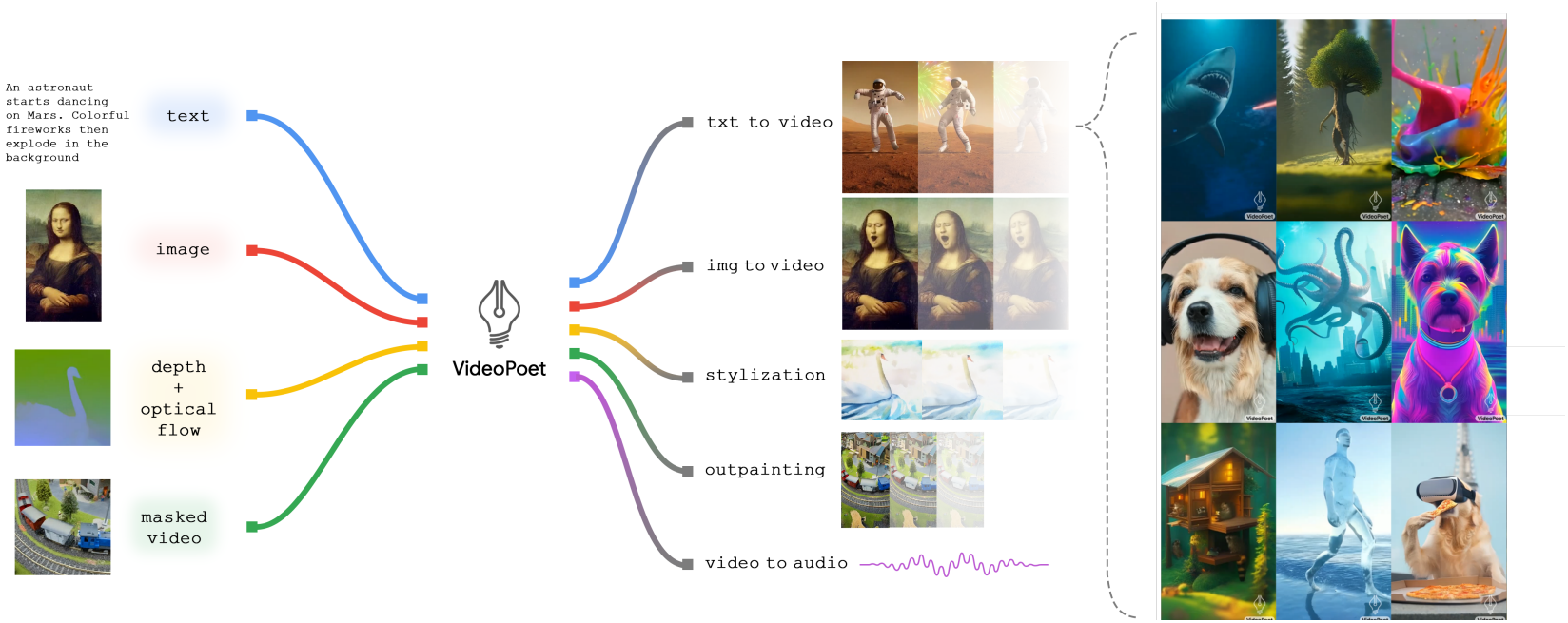


Figure 8.1: **VideoPoet overview**: a versatile video generator that conditions on multiple types of inputs and performs a variety of video generation tasks.

thermore, VideoPoet can synthesize coherent long videos of up to 10 seconds by autoregressively extending the content, conditioned on the last second of the generated video.

We also demonstrate that VideoPoet is capable of zero-shot video generation. We use the term "zero-shot video generation" as VideoPoet processes new text, image, or video inputs that diverge from the training data distribution. Furthermore, VideoPoet handles new tasks not included in its training. For example, VideoPoet is able to perform new editing tasks by sequentially chaining training tasks together. The main contributions of this work are:

- A method for training a Large Language Model (LLM) specifically for video generation tasks, utilizing tokenized video data that incorporates both text-paired and unpaired video data.
- Evaluations and demonstrations to highlight VideoPoet's competitive and state-of-the-art performance, especially in generating realistic and interesting videos with motion.

8.2 Prior Work

Video diffusion models. Recently, numerous video generation methods use diffusion-based methods for text-to-video [15, 16, 64, 81, 88, 177, 209, 214, 215, 250, 251, 251, 260] and video-to-video editing [38, 58, 61, 129]. As video diffusion models are usually derived from text-to-image diffusion models [160, 169], additional tasks and modalities are added via inference tricks [140], architectural changes [58, 129] and adapter layers [73, 252]. Although these models are composable after training, they are not trained end-to-end in a unified framework. Our multitask pretraining strategy in a single model improves performance and provides zero-shot video generation capabilities.

Language models for video and image generation. Video language models are typically derived from the general family of transformer-based language models [157, 201] that easily combine multiple tasks in pretraining and demonstrate powerful zero-shot capabilities. Image generation language models can generate images autoregressively [235] or via masked prediction [33, 34]. Both families have been extended to text-to-video [93, 97, 203, 232] using paired data. While other text-to-video work with transformers only leverages video-text pairs for training, we also leverage unpaired videos (without text) and the same video for different tasks. Since video language models can flexibly incorporate numerous tasks [143, 242], including video-to-video, we extend this family of work to text- and multimodal-conditioned tasks in this work with a synergistic pretraining strategy across various tasks.

Pretraining task design in LLMs. As language models can easily incorporate multiple training tasks, task selection is an important area of research. GPT-3 [25] and PaLM [40] demonstrate that training LLMs on diverse tasks leads to positive scaling effects on zero- and few-shot tasks. Other approaches show that masking approaches are a valuable learning target [91, 242, 245]. As the model size grows, training data must grow as well [91] to maintain similar performance. Our pretraining strategy enables using the same video for multiple training tasks even without paired text. This design facilitates training on a large quantity of video-only examples, thereby decreasing the demand for video-text pairs.

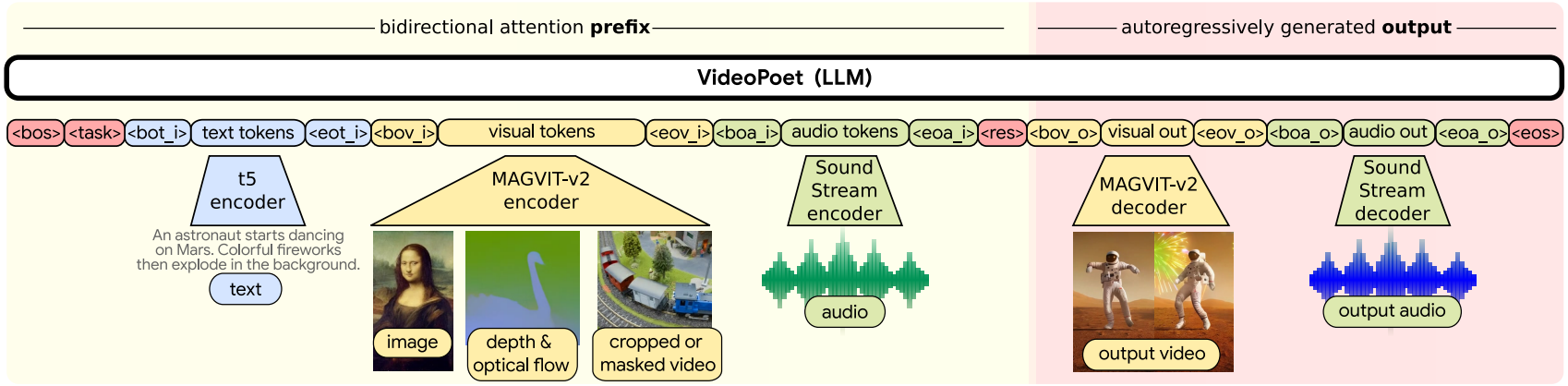


Figure 8.2: **Sequence layout for VideoPoet**. We encode all modalities into the discrete token space, so that we can directly use large language model architectures for video generation. We denote special tokens in <> (see Table 8.3 for definitions). The modality agnostic tokens are in darker red; the text related components are in blue; the vision related components are in yellow; the audio related components are in green. The left portion of the layout on light yellow represents the bidirectional prefix inputs. The right portion on darker red represents the autoregressively generated outputs with causal attention.

8.3 VideoPoet Model Design

We propose an effective method for video generation and related tasks from different input signals by leveraging large language models. Our model consists of modality-specific tokenizers and a language model backbone (Fig. 8.2). The tokenizers map input data – *i.e.* image pixels, video frames, and audio waveforms – into discrete tokens in a *unified* vocabulary. The visual and audio tokens are flattened into a sequence of integers. Next, the LLM accepts these tokens as input along with text embeddings, and is responsible for generative multi-task and multimodal modeling. As illustrated in Fig. 8.2, VideoPoet conditions on text embeddings, visual tokens, and audio tokens, and autoregressively predicts visual and audio tokens.

8.3.1 Tokenization

We employ the MAGVIT-v2 [245] tokenizer introduced in Chapter 5 for joint image and video tokenization, and the SoundStream [249] tokenizer for audio. Visual and audio vocabularies are concatenated into a unified vocabulary. The text modality is represented by embeddings.

Image and video tokenizer. Visual tokenizer is key to generating high-quality video content, often determining the upper limit of achievable video generation quality [245]. After analyzing existing tokenizers [57, 203, 242, 243], we choose the MAGVIT-v2 [245] tokenizer due to its performance in visual quality and high compression capabilities, which effectively reduce the sequence length required by the LLM, thereby facilitating more efficient and effective learning. Specifically, a video clip is encoded and quantized into a sequence of integers, with a decoder mapping them back to the pixel space. MAGVIT-v2 tokenizes 17-frame 2.125-second 128×128 resolution videos sampled at 8 fps to produce a latent shape of $(5, 16, 16)$, which is then flattened into 1280 tokens, with a vocabulary size of 2^{18}. We also tokenize videos into a portrait aspect ratio at 128×224 resolution, producing a latent shape of $(5, 28, 16)$, or 2240 tokens.

We enforce causal temporal dependency, which facilitates the generation of longer videos. To jointly represent images and videos, we encode the initial frame of a video or a static image into tokens with a consistent shape of $(1, 16, 16)$. We use the COMMIT [242] encoding scheme to tokenize the inpainting and outpainting tasks.

Audio tokenizer. We tokenize audio clips with a pretrained SoundStream [249] tokenizer. We embed 2.125 seconds of audio to produce 106 latent frames with a residual vector quantizer (RVQ) of four levels. To improve audio generation performance, we transpose the clip before flattening so that the model predicts the full audio clip at each RVQ granularity level before moving on to the finer grained levels. Finally, each RVQ level has a disjoint vocabulary with each level containing 1,024 codes. This results in a combined audio vocabulary size of 4,096 codes.

Text embedding as input. Pretrained text representations, in general, outperform training our model by learning text tokens from scratch. We use pretrained language embeddings from a frozen T5 XL encoder [157]. For tasks with text guidance, such as text-to-video, T5 XL embeddings are projected into the transformer's embedding space with a linear layer.

8.3.2 Language Model Backbone

After converting the image, video, and audio modalities into discrete tokens within a shared vocabulary, we can directly leverage a language model to generate videos and audios in the token space. We use a prefix language model with a decoder-only architecture as the backbone. By constructing different patterns of input tokens to output tokens during training, we can control the tasks the model is able to perform as explained in Section 8.3.3.

8.3.3 Task Prompt Design

We design a pretraining task mixture, each with a defined prefix input and output. The model conditions on the prefix, applying the loss solely to the output. Fig. 8.2 shows a typical input-output sequence layout. For each task, the input sequence may include three types of values: text embeddings (T5), visual tokens (MAGVIT-v2), and audio tokens (SoundStream). The model outputs two types of tokens: visual and audio tokens. To facilitate training, VideoPoet employs *special tokens*, as listed in Tab. 8.3. In the following, we describe key designs for the task prompts.

Pretraining tasks. We consider the following tasks:

- Unconditioned video generation: Generate video frames without conditioning on an input.
- Text-to-video (T2V): Generate video from a text prompt.
- Video future prediction (FP): Given an input video of variable length, predict future frames.
- Image-to-video (I2V): Given the first frame of a video as an input image, predict the future frames.
- Video inpainting/outpainting (Painting): Given a masked video, predict the video with the masked contents filled in.
- Video stylization: Given text, optical flow, and depth, predict the video frames (Section 8.3.3).
- Audio-to-video: Given an input audio waveform, predict the corresponding video.
- Video-to-audio: Given an input video, predict the corresponding audio waveform.
- Audio-video continuation (AVCont): given an input frame and its audio, predict the rest of the video and audio.

To indicate the type of task, we condition on the `<task>` token, which has a unique value for each unique output. We note that not all input variations need a new `<task>`; the model adapts to different context signals for identical outputs. For instance, text-to-video, image-to-video, and unconditioned video generation share the same `<task>`. If a modality is absent in a task, related input/output tokens and special tokens are excluded, shortening the sequence.

Representing an image as a video. In text-to-image pretraining, we omit the <eos> and <eov_o> tokens from the input sequence, enabling continuous token generation for inference of longer videos. This approach blurs the boundary between video and image generation tasks, enhancing cross-modality information sharing. This design leads to the prediction of higher-quality initial frames and reduces errors and artifacts in subsequent frames.

Video token format. We generate video tokens at two resolutions, 128×128 and 128×224, each available in two lengths: 17 frames and 41 frames, both encoded at 8 frames per second. Special conditioning tokens are used to signal the desired resolutions and durations for video generation. Images are a special case of a 1-frame video, which we tokenize at 128×128 resolution.

8.3.4 Training Strategy

We find that sampling from image and video datasets uniformly across time can lead to sub-optimal results, as training on images can enhance the model's understanding of objects but does not capture any motions that are represented in video data. Thus, we devise a two-stage pretraining strategy, where we augment our sampling weights to sample image data 90% of the time and video data 10% of the time for the first 25% iterations of training. We then switch to training on video 90% and image 10% for the remaining iterations.

We fine-tune our pretrained model for enhanced performance on specific tasks or for new task adaptation, such as text-to-video and image-to-video tasks, using a high-quality data subset. This results in improved generation quality, consistent with [259], and addresses decoding collapse issues, characterized by repetitive token predictions. Such fine-tuning not only diversifies outputs but also allows for a higher classifier-free guidance scale [86], boosting overall quality.

8.4 Experimental Results

8.4.1 Experimental Setup

Training tasks. We train the model on a mixture of pretraining tasks as detailed in Section 8.3.3. We finetune a model on a high-quality training subset for text-to-video evaluations, as discussed in Section 8.3.4. Unless explicitly stated, we do not finetune on specific tasks for evaluations.

Datasets. We train on a total of 1B image-text pairs and ~270M videos (~100M with paired text, of which ~50M are used for high-quality finetuning, and ~170M with paired audio) from the public internet and other sources, *i.e.* around 2 trillion tokens across all modalities. The data has been filtered to remove egregious content and sampled to improve contextual and demographic diversity.

Evaluation protocol. We employ a zero-shot generation evaluation protocol, as the model has not been trained on the training data of target benchmarks. Specifically, the evaluation benchmark includes two text-to-video generation datasets, MSR-VTT [229] and UCF-101 [184], as well as the frame prediction task on Kinetics 600 (K600) [30], in which the first 5 frames are provided as the condition to predict the next 11 frames. We also include inpainting and outpainting tasks [242] on Something-Something V2 (SSv2) [68].

We adopt widely used metrics such as Fréchet Video Distance (FVD) [198], CLIP similarity score [224], and Inception Score (IS) [171] for evaluation. Note that the specific metrics and evaluation methods vary across different datasets. Detailed information on these variations can be found in Section 8.4.6.

8.4.2 Pretraining Task Analysis

We investigate the learning capabilities of different combinations of pretraining tasks using a model with 300 million parameters. All task combinations are trained using a learning rate of 10^{-3} for the same number of steps (300k) with a batch size of 1024.

For the analysis of pretraining tasks, we consider text-to-video (T2V), text-to-image (T2I), and four self-supervised learning (SSL) tasks: frame prediction (FP), central inpainting and central outpainting (Painting) [242] and audio-video continuation (AVCont) where the model is provided with the first frame and its corresponding audio to predict the subsequent 16 frames and matching audio. For each video task, we uniformly select 20% of training samples from a random subset of 50 million videos. For the text-to-image task, we randomly sample 50 million text-image pairs from our training dataset. For tasks involving audio, our sampling is exclusive to videos that contain an audio track.

The evaluation results are presented in Tab. 8.1. We assess a model across the four tasks within the zero-shot evaluation benchmark: the T2V task on MSR-VTT [229] and UCF 101 [184], FP on K600 [30], and the Painting tasks on SSv2 [68]. In these experiments, we employ a single model to perform all the tasks. The model is not trained on the training data of these evaluation datasets, and thus it is a zero-shot evaluation.

The top rows of Tab. 8.1 depict each pretraining task configuration of the 300 million parameter model, which are comparable in their setups. Our evaluation benchmarks span diverse visual domains, posing a challenge to achieving consistent improvement across all of them. Nevertheless, incorporating all pretraining tasks results in the best overall performance, on average, across all evaluated tasks. Additionally, the significant disparity observed in the "SSL" row suggests the limitations of self-supervised training and underscores the necessity for text-paired data during training. The last row, "ALL (8B)", is the model with 8 billion parameters, trained on the pretraining tasks as discussed in Section 8.3.4 and utilized significantly more compute.

8.4.3 Model Scaling and Performance

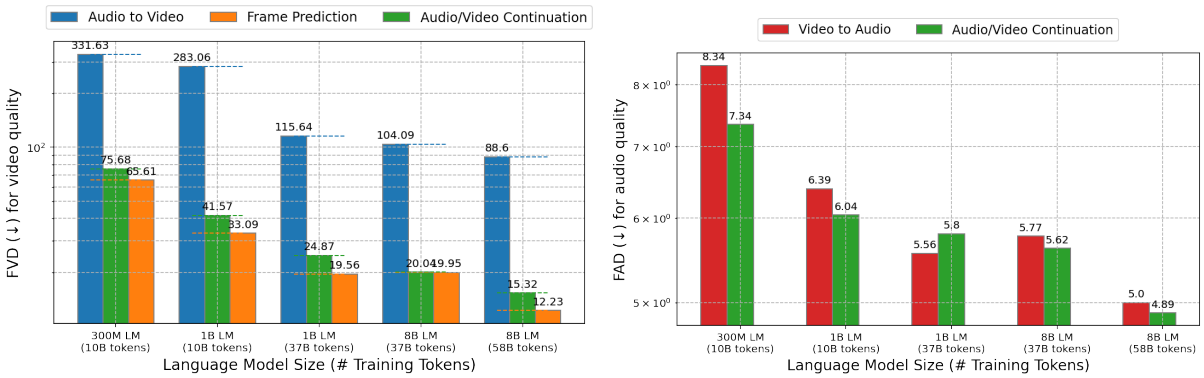
To analyze model performance versus model scale, we use a subset of the training set without text-paired data and a slightly different task prompt design. We evaluate the video generation quality using FVD [198] and audio generation quality using the Fréchet Audio Distance (FAD),

Method	Pretraining Tasks						Zero-shot Evaluation Benchmark				
							T2V		FP	Inpainting	Outpainting
	T2I	T2V	Uncond	FP	Painting	AVCont	MSR-VTT CLIPSIM ↑	UCF101 FVD ↓	K600 FVD ↓	SSv2 FVD ↓	SSv2 FVD ↓
T2V		✓					0.244	822	759	2,333	2,310
T2V+I	✓	✓					0.247	1,025	794	2,118	1,916
SSL			✓	✓	✓	✓	0.226	1,742	700	1,093	1,500
NO T2I		✓	✓	✓	✓	✓	0.235	1,008	755	95	389
ALL	✓	✓	✓	✓	✓	✓	0.240	1,085	729	127	636
ALL (8B)	✓	✓	✓	✓	✓	✓	0.305	355	687	4.7	13.76

Table 8.1: **Pretraining task analysis on 300M models.** The top rows list models with 300M parameters, trained on a subset of the data, and are comparable to each other. The last row shows an 8B model trained on the entire dataset. **T2I** (text-to-image), **T2V** (text-to-video), **FP** (frame prediction), **Painting** (inpainting/outpainting), **Uncond** (unconditional generation), **AVCont** (audio-video continuation), and **SSL** (self-supervised learning).

which uses the VGGish model as the embedding function [84]. Both FVD and FAD metrics are calculated using a held-out subset of 25 thousand videos.

Fig. 8.3 shows that as the model size grows and the amount of training data increases, performance improves across visual and audiovisual tasks. After obtaining the above results, we retrain our 1B and 8B models using the task design and text-paired training data discussed in Section 8.3.4.



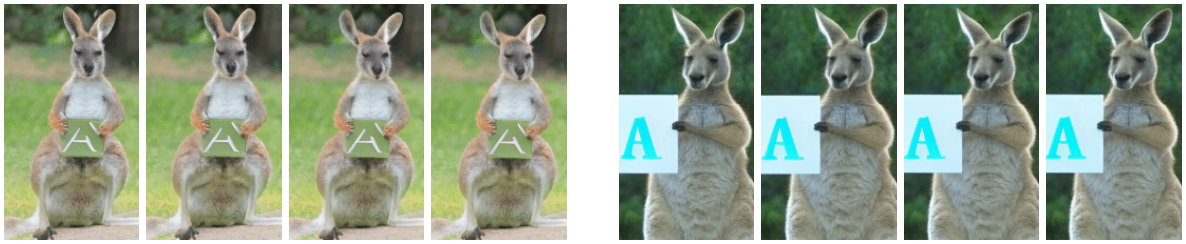
(a) Video generation quality in FVD (↓).

(b) Audio generation quality in FAD (↓).

Figure 8.3: **Effects of model and data scale on video and audio generation quality**. The performance, depicted on a log-log scale, improves significantly when we scale up the model and training data. Language models with 300 million, 1 billion, and 8 billion parameters are trained on datasets comprising 10, 37, and 58 billion visual and audio tokens, respectively.

Qualitative comparison of 1B and 8B models. We show a qualitative comparison of the 1B and 8B pretrained models. In Figure 8.4, we show outputs of 1B and 8B parameter models on the same prompts. Four frames from the best video output of each model in a batch of four text-to-video samples were selected to represent the model. In the first row, the 1B model is unstable with large changes to the subject over time and misses elements from the complex prompt. This prompt was originally used for scaling comparisons in [235], and compared to a dedicated image-only model, our model does not preserve text as well given the training data used. In the second row, we use a simpler text task and show that the 8B model can represent a single letter clearly, but the 1B model still produces artifacts. In the third row, we show that the 8B model learns spatial positioning such as the river being in front of the astronaut and horse. In the fourth row, we show that the 8B parameter model learned a stop motion style to have items disappear “one by one" and can follow a complicated layout from a long prompt. In contrast, the 1B model includes all of the nouns, but is unstable over time and does not follow the layout indicated in the prompt. In the bottom row, we show that the 8B model understands counts of objects in that it displays a full bouquet (though 12 roses are not explicitly in frame) and smooth consistent motion as opposed to the 5 roses and distorting objects produced by the 1B model. Overall, scaling the model improves temporal consistency, prompt fidelity, and motion dynamics while adding capabilities for limited text rendering, spatial understanding, and counting.

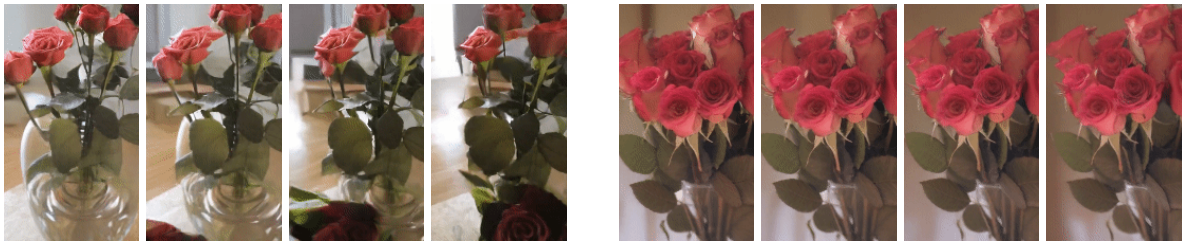
prompt: A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



prompt: A kangaroo holding a sign with the letter A on it

prompt: A photo of an astronaut riding a horse in the forest. There is a river in front of them with water lilies

prompt: A zoomed out map of the United States made out of sushi. It is on a table next to a glass of red wine. Pieces of sushi disappear one by one



prompt: Rotating around a vase holding a dozen roses

Figure 8.4: **A comparison between 1B (left) and 8B (right) parameter models** on the same prompt and settings.

Model	MSR-VTT		UCF-101	
	CLIPSIM	FVD	FVD	IS
CogVideo (EN) [93]	0.2631	1294	702	25.27
MagicVideo [260]	-	998	655	-
Video LDM [16]	0.2929	-	551	33.45
ModelScopeT2V [209]	0.2930	550	-	-
InternVid [215]	0.2951	-	617	21.04
VideoFactory [214]	0.3005	-	410	-
Make-A-Video [177]	0.3049	-	367	33.00
Show-1 [251]	0.3072	538	394	35.42
VideoPoet (Pretrain)	0.3049	**213**	**355**	**38.44**
VideoPoet (Task adapt)	**0.3123**	-	-	-

Table 8.2: **Comparison on zero-shot text-to-video benchmarks.**

8.4.4 Comparison with the State-of-the-Art

Text-to-Video (T2V). Tab. 8.2 shows zero-shot text-to-video evaluation results on the common MSR-VTT [229] and UCF-101 [184] datasets. Our model performs favorably in terms of CLIP similarity and FVD scores on MSR-VTT and UCF-101. The pretrained foundation model already achieves competitive performance on all metrics. After finetuned on high-quality subset of text-video pairs, VideoPoet achieves even better CLIPSIM on MSR-VTT. More details on the evaluation settings can be found in Section 8.4.6.

Human Evaluations with Text-to-Video (T2V). We analyze VideoPoet using human raters and compare with other recent models: Show-1 [251], VideoCrafter [36], Phenaki [203], Pika [148], and Gen2 [168]. Show-1, VideoCrafter, Pika, and Runway are video diffusion models while Phenaki is a token-based model using masked token modeling [33]. We ran the most up-to-date model versions as of January 2024.

We first develop a unified evaluation prompt bank consisting of prompts from a variety of categories and styles. Our prompts are sourced from published prompt sets (*e.g.*, Show-1, Video LDM [16]). We select the prompts prior to generating videos and fix these choices after initial selection. We also select preferentially for prompts that contain an explicit mention of motion so that the evaluation would not be biased for models that generate high quality videos that are almost still (e.g., "person jumping off of a chair" over "person standing on a chair").

For this user study we use the fine-tuned version of VideoPoet as discussed in Section 8.3.4 and compare against alternative models in a side-by-side fashion for each prompt. Raters are shown videos generated by two models at a time (in randomized order so as to not bias raters). Not all methods generate videos at the same size or aspect ratio, and we resize each video to a fixed area while maintaining its original aspect ratio. Raters are then asked to compare the videos along 5 dimensions and for each dimension to report which video is better. The 5 dimensions are: (1) text fidelity (which video follows the text prompt most faithfully), (2) video quality, (3) motion "interestingness", (4) motion realism, and (5) temporal consistency. Raters

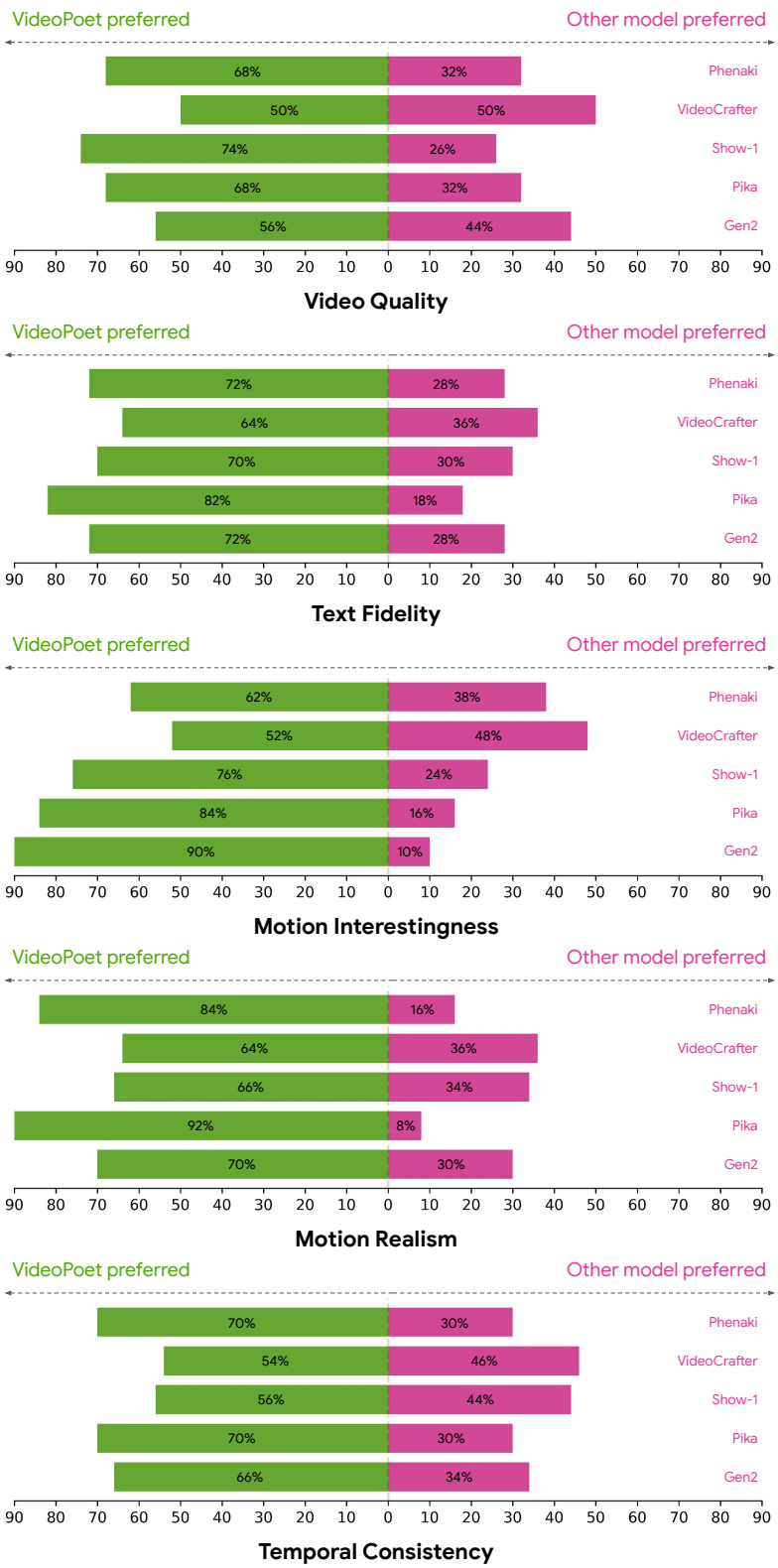


Figure 8.5: **Human evaluation results on text-to-video (T2V) generation.** Green and pink bars represent the proportion of trials where VideoPoet was *preferred over* or *less preferred* to an alternative, respectively.

Figure 8.5: **10-Second long video generation example.** By predicting 1-second video segments from an initial 1-second clip, VideoPoet can iteratively generate videos of extended lengths.

are required to go through a collection of training examples for each of these 5 dimensions before they begin.

Our findings are summarized in Fig. 8.5, where green and pink bars represent the proportion of trials where VideoPoet was *preferred* or *less preferred* over an alternative, respectively. In summary, VideoPoet outperforms all baseline models along almost all of the dimensions and achieves its most significant wins along the motion categories.

8.4.5 LLM's Diverse Capabilities in Video Generation

This subsection presents several capabilities we discover from the pretrained VideoPoet, shedding light on the LLM's promising potential in video generation. By combining the flexibility of our autoregressive language model to perform diverse tasks such as extending video in time, inpainting, outpainting, and stylization, VideoPoet accomplishes multiple tasks in a unified model.

Coherent long video generation and image-to-video. A benefit of an decoder-based language model is that it pairs well with autoregressively extending generation in time. We present two different variants: Generating longer videos (Fig. 8.5) and converting images to videos. Encoding the first frame independently allows us to convert any image into the initial frame of a video without padding. Subsequent frames are generated by predicting remaining tokens, transforming the image into a video as shown in Fig. 8.6[1].

This results in the capability to generate videos longer than 10 seconds or to allow users to iteratively extend video clips based on previously generated video, and produces temporally

[1]For image-to-video examples we source images from Wikimedia Commons: https://commons.wikimedia.org/wiki/Main_Page

Animated from historical photo

Animated from painting

Figure 8.6: **Examples of videos animated from still images** plus text prompts tailored to each initial image.

consistent videos without significant distortion. Such capabilities are rarely observed in contemporary diffusion models.

Zero-shot video editing and task chaining. With the multi-task pretraining, VideoPoet exhibits task generalization that can be chained together to perform novel tasks. We show the model can apply image-to-video animation followed by video-to-video stylization in Fig. 8.7. Fig. 8.8 shows another example applying video-to-video outpainting, followed by editing them with additional video-to-video effects. At each stage, the quality of the output is sufficient to remain in-distribution (*i.e.* teacher forcing) for the next stage without noticeable artifacts. These capabilities can be attributed to our multimodal task design within an LLM transformer framework that allows for modeling multimodal content using a single transformer architecture over a unified vocabulary.

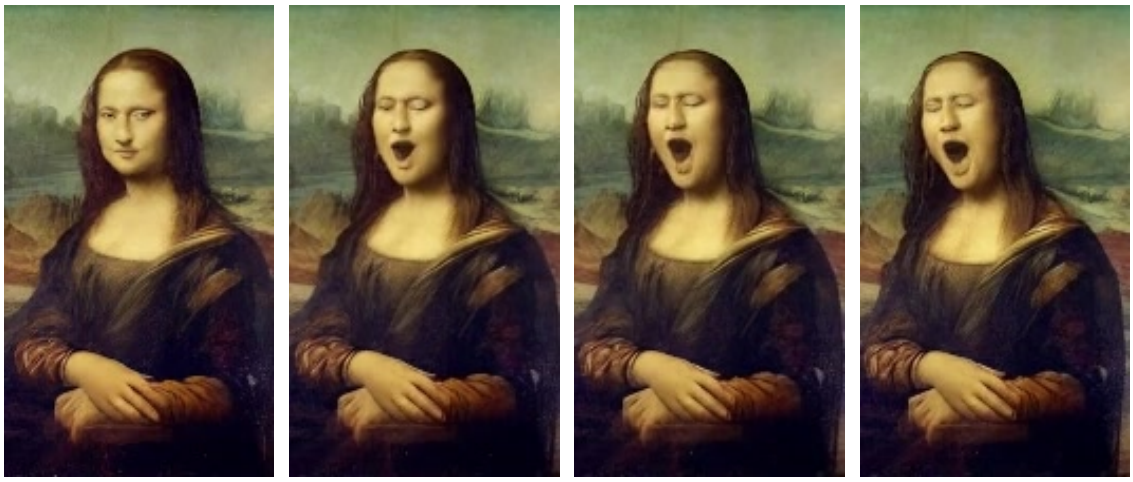
3D structure, camera motion, visual styles. Even though we do not add specific training data or losses to encourage 3D consistency, our model can rotate around objects and predict reasonable visualizations of the backside of objects. Additionally, with only a small proportion of input videos and texts describing camera motion, our model can use short text prompts to apply a range of camera motions to image-to-video and text-to-video generations (see Fig. 8.9).

8.4.6 Additional Implementation Details

Training settings. The unified vocabulary is constructed as follows: the initial 256 codes are reserved for special tokens and task prompts. Table 8.3 lists some examples of special tokens. Subsequently, the next 262,144 codes are allocated for image and video tokenization. This is followed by 4,096 audio codes. We also include a small text vocabulary of English words. Overall, this produces a total vocabulary size of approximately 300,000.

Since the first frame is tokenized separately, MAGVIT-v2 allows images to be represented in the same vocabulary as video. In addition to being more compact, images provide many learnable characteristics that are not typically represented in videos, such as strong visual styles (*e.g.*, art paintings), objects which are infrequently seen in video, rich captions, and significantly more text-image paired training data. When training on images, we resize the images to 128×128 which are then tokenized to a latent shape of $(1, 16, 16)$, or 256 tokens. We scale the MAGVIT-v2 model's size and train it on the datasets discussed in Section 8.4.1. The training follows two steps: image training, inflation [245] and video training. Due to images requiring fewer tokens, we can include roughly 5× more images per batch than videos, *i.e.* 256 image tokens vs. 1280 video tokens. We use up to a maximum of 64 text tokens for all of our experiments. For the `<res>` token, the resolution is only specified for 128×224 output, 128×128 resolution is assumed otherwise.

The video-to-video tasks use the COMMIT encoding [242] to obtain the tokens for the tasks such as inpainting and outpainting. Text is encoded as T5 XL embeddings [157] and are inserted into reserved sequence positions right after the `<bot_i>` token as shown in Figure 8.2.



Animated from still image

Stylized video

Prompt: An oil painting of a snowman with a red hat opening their mouth to yawn

Figure 8.7: **Example of zero-shot video editing via task chaining** (text conditioned image-to-video and stylization)

Original Video

Outpainted Video

Stylized Video
Prompt: A gingerbread and candy train on a track

Figure 8.8: **Example of zero-shot video editing via task chaining** (outpainting and stylization) – the original video is first outpainted and then stylized via a text prompt.

Camera Motion: Arc shot

Camera Motion: FPV drone shot

Figure 8.9: **Examples of directed camera movement** from the same initial frame.

Special Token	Usage
<bos>	Beginning of sequence
<task>	Task to perform for this sequence
<bot_i>	Beginning of the text input.
<eot_i>	End of the text input.
<bov_i>	Beginning of the visual input.
<eov_i>	End of the video input.
<boa_i>	Beginning of the audio input.
<eoa_i>	End of the audio input.
<source>	The source of the video to generate.
<res>	Output resolution for the video.
<bov_o>	Beginning of the video output.
<eov_o>	End of the video output.
<boa_o>	Beginning of the audio output.
<eoa_o>	End of the audio output.
<eos>	End of the entire sequence.

Table 8.3: **List of representative special tokens** used in training and inference.

Zero-shot text-to-video evaluation settings. We report the details of our zero-shot text-to-video settings here. We note that some details are missing in previous papers and different papers use different settings. Hence, we provide all the details and hope this evaluation setting can serve as a standard text-to-video generation benchmark. Our results are reported on the 8B model and we adopt classifier-free guidance [86]. All metrics are evaluated on generated videos containing 16 frames with a resolution of 256×256. We first generate videos of 128×128 resolution and then resize to 256×256 via bicubic upsampling.

Zero-shot MSR-VTT. For CLIP score, we used all 59,794 captions from the MSR-VTT test set. We use CLIP ViT-B/16 model following Phenaki [203]. We note that some papers use other CLIP models, *e.g.*, VideoLDM [16] uses ViT-B/32. Our CLIP score evaluated on the ViT-B/32 backbone for MSR-VTT is 30.01. For the FVD metric, to evaluate on a wide range of captions as well as to be comparable with previous papers that evaluate on 2,048 videos, we evaluate on the first 40,960 captions in the MSR-VTT test set. More specifically, we report the FVD metrics on 2048 videos with 20 repeats. The FVD real features are extracted from 2,048 videos sampled from the MSR-VTT test set. We sample the central 16 frames of each real video, without any temporal downsampling, *i.e.*, we use the original fps in the MSR-VTT dataset (30 fps as reported in [229]). The FVD is evaluated with an I3D model trained on Kinetics-400.

Zero-shot UCF-101. Following VDM [90], we sample 10,000 videos from the UCF-101 test set and use their categories as the text prompts to generate 10,000 videos. We use the class text prompts provided in PYoCo [64] to represent the 101 categories. To compute the FVD real features, we sample 10K videos from the training set, following TGAN2 [171]. We sample the central 16 frames for each real video , without any temporal downsampling, *i.e.*, we use the original fps in the UCF-101 dataset (25 fps as reported in [184]). The FVD metric is evaluated with an I3D model trained on Kinetics-400 and the IS metric is evaluated with a C3D model

trained on UCF-101.

Self-supervised tasks evaluation settings. Self-supervised learning tasks include frame prediction on K600 with 5 frames as condition, as well as inpainting and outpainting on SSv2. FVD [198] is used as the primary metric, calculated with 16 frames at 128×128 resolution. We follow MAGVIT [242] in evaluating these tasks against the respective real distribution, using 50000×4 samples for K600 and 50000 samples for SSv2.

8.5 Summary

VideoPoet demonstrates the potential of a large language model that is trained on discrete visual, text and audio tokens, in generating videos of compelling state-of-the-art quality. A particular strength of our model lies in its ability to generate high-fidelity, large, and complex motions. Our large language model formulation benefits from training across a variety of *multi-modal* tasks with a unified architecture and vocabulary. Consequently, the pretrained model is adept at *multi-task* video creation, and serves as a foundation for a diverse variety of video generation related capabilities, including multiple forms of editing.

Limitations. Despite VideoPoet demonstrating highly competitive performance of LLMs relative to state-of-the-art models, certain limitations are still observed. For example, the RGB frame reconstruction from compressed and quantized tokens place an upper bound on the generative model's visual fidelity. Second, the per-frame aesthetic biases in static scenes does not match the best baseline. This difference is largely due to a choice of training data, where we focus our training on more natural aesthetics and excluded some sources containing copyrighted images, such as LAION [175], which is commonly used in other work. Finally, small objects and fine-grained details, especially when coupled with significant motions, remains difficult within the token-based modeling.

Conclusion

Conclusion

Summary

This thesis has explored the development of multi-task models for generating videos and other modalities under diverse conditions, as well as for understanding and compression applications. Throughout this research, we have demonstrated the effectiveness of integrating multiple tasks, crafting high-fidelity latent representations, and generating multiple modalities.

Initial work on separate multi-task and multi-modal setups with pixel-space models has revealed potential, but also demonstrated the constraints imposed by fixed label spaces and specialized architectures. This has underscored the importance of pursuing more flexible and adaptable designs.

Seeking compact and informative representations of high-dimensional visual data, we have developed spatial-temporal video tokenizers that have maintained remarkable fidelity. We have introduced a novel method for bidirectional mapping between visual observations and interpretable linguistic concepts. Additionally, we have created a scalable visual token representation that has proven beneficial across generation, compression, and understanding tasks. These advancements have marked early instances of language models outperforming diffusion models in visual synthesis and a video tokenizer exceeding the performance of industry-standard codecs.

Building upon these multi-modal latent spaces, we have investigated the design of multi-task generative models. Our masked multi-task transformer has demonstrated favorable performance in video generation across quality, efficiency, and flexibility. We have successfully enabled a language model, trained exclusively on text, to generate images and videos. Finally, we have constructed a scalable multi-modal generative transformer trained from scratch, capable of generating videos and audio under various conditions. This model has notably generated high-fidelity videos with corresponding audio in a zero-shot setting.

Contributions

We briefly summarize our contributions organized by these included publications:

- Chapter 1: [241] **Lijun Yu**, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann.
  Argus++: Robust Real-Time Activity Detection for Unconstrained Video Streams with Overlapping Cube Proposals.
  In WACV 2022 HADCV Workshop.

- We propose Argus++, a real-time activity detection system for unconstrained video streams, which is robust across different scenarios.
- We introduce overlapping spatial-temporal cubes as the core concept of activity proposals to ensure coverage and completeness of activity detection through oversampling.
- The proposed system has achieved outstanding performance in a large series of activity detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

- Chapter 2: [244] **Lijun Yu**, Jin Miao, Xiaoyu Sun, Jiayi Chen, Alexander G. Hauptmann, Hanjun Dai, and Wei Wei.
  DocumentNet: Bridging the Data Gap in Document Pre-Training.
  In EMNLP 2023 Industry Track.
  - We propose a pretraining-finetuning paradigm with lightweight UniFormer models and three objectives for unified token representation.
  - Extensive experiments on VDER tasks demonstrate the favorable performance of UniFormer pretrained on DocumentNet compared to the commonly used IIT-CDIP.
- Chapters 3 and 6: [242] **Lijun Yu**, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa and Lu Jiang.
  MAGVIT: Masked Generative Video Transformer.
  In CVPR 2023 as a Highlight paper.
  - To the best of our knowledge, we present the first masked multi-task transformer for efficient video generation and manipulation. We show that a trained model can perform ten different tasks at inference time.
  - We propose an effective embedding method with diverse masks for numerous video generation tasks.
  - We show that MAGVIT achieves the best-published fidelity on three widely-used benchmarks, including UCF-101, BAIR Robot Pushing, and Kinetics-600 datasets.
- Chapters 4 and 7: [243] **Lijun Yu**, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David A. Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, Kevin Murphy, Alexander G. Hauptmann, Lu Jiang.
  SPAE: Semantic Pyramid AutoEncoder for Multimodal Generation with Frozen LLMs.
  In NeurIPS 2023 as a Spotlight paper.
  - This is the first successful method, to the best of our knowledge, that uses a frozen language model, trained solely on language tokens, to directly generate image content through in-context learning.
  - We propose a new progressive prompting method that facilitates in-context generation of long cross-modal sequences.
  - We evaluate our method on visual understanding and generation tasks, and notably, our approach outperforms the best-published few-shot image classification

accuracy [132] by an absolute 25% under the same in-context setting.

- Chapter 5: [245] **Lijun Yu**, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang.
  Language Model Beats Diffusion – Tokenizer is Key to Visual Generation.
  To appear in ICLR 2024.
  - A new video tokenizer that outperforms the previously best-performing video tokenizer in three areas: visual generation, video compression, and action recognition.
  - A novel lookup-free quantization approach that enables improving the visual generation quality of language models by learning a large vocabulary.
  - To the best of our knowledge, the first evidence suggesting that a language model can outperform diffusion models on ImageNet when provided with the same training data, an equivalent model size, and a similar training budget.
  - A video compressor with better quality than HEVC and VVC, at similar bit rates, according to user studies. To our knowledge, this is the first successful attempt of a visual tokenizer designed for video generation to achieve comparable results to standard codecs.
- Chapter 8: [113] Dan Kondratyuk*, **Lijun Yu***, Xiuye Gu*, José Lezama*, Jonathan Huang*, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold*, and Lu Jiang*. * indicates equal contribution.
  VideoPoet: A Large Language Model for Zero-Shot Video Generation.
  To appear in ICML 2024.
  In this large collaborative effort, my contribution primarily includes
  - A method for training a Large Language Model (LLM) specifically for video generation tasks utilizing tokenized video frames and corresponding audio.
  - A preliminary study on the scaling behavior of VideoPoet on audio-visual generation tasks.

In addition, we have the following publications referenced:

- [236] **Lijun Yu**, Dawei Zhang, Xiangqun Chen, and Alexander G Hauptmann.
  Traffic Danger Recognition With Surveillance Cameras Without Training Data.
  In AVSS 2018.
- [237] **Lijun Yu**, Peng Chen, Wenhe Liu, Guoliang Kang, and Alexander G Hauptmann.
  Training-Free Monocular 3D Event Detection System for Traffic Surveillance.
  In Big Data 2019.
- [239] **Lijun Yu**, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann.
  CMU Informedia at TRECVID 2020: Activity Detection with Dense Spatio-Temporal Proposals.

In TRECVID 2020.

- [152] Yijun Qian*, **Lijun Yu***, Wenhe Liu, and Alexander G Hauptmann. * indicates equal contribution.
ELECTRICITY: An Efficient Multi-Camera Vehicle Tracking System for Intelligent City.
In CVPRW 2020.
- [238] **Lijun Yu**, Qianyu Feng, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann.
Zero-VIRUS: Zero-shot Vehicle Route Understanding System for Intelligent Transportation.
In CVPRW 2020.
- [240] **Lijun Yu**, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann.
CMU Informedia at TRECVID 2021: Activity Detection with Argus++.
In TRECVID 2021.
- [154] Yijun Qian, **Lijun Yu**, Wenhe Liu, and Alexander G Hauptmann.
Rethinking Zero-Shot Action Recognition: Learning from Latent Atomic Actions.
In ECCV 2022.
- [187] Haoran Sun, **Lijun Yu**, Bo Dai, Dale Schuurmans, and Hanjun Dai.
Score-based Continuous-time Discrete Diffusion Models.
In ICLR 2022.
- [141] Jackson Michaels, Juncheng Li, Laura Yao, **Lijun Yu**, Zach Wood-Doughty, and Florian Metze.
Audio-Journey: Open Domain Latent Diffusion Based Text-to-Audio Generation.
In ICASSP 2024.
- [75] Agrim Gupta, **Lijun Yu**, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama.
Photorealistic Video Generation with Diffusion Models.
In submission to ECCV 2024.

During the course, we have given these invited talks:

- Traffic Danger Recognition With Surveillance Cameras Without Training Data.
  - ICCV Demo, Oct 2019.
  - TRECVID, Nov 2019.
- Argus++: Real-Time Activity Detection in Unknown Facilities with Dense Spatio-Temporal Proposals.
  - WACV HADCV, Jan 2021.
  - CVPR ActivityNet, Jun 2021.
- ArgusRoad: Road Activity Detection with Connectionist Spatio-Temporal Proposals.
  - ICCV ROAD, Oct 2021.
- Language Model Beats Diffusion: Tokenizer is Key to Visual Generation.
  - Hong Kong University of Science and Technology, Jan 2024.
  - Institute of Computing Technology, Chinese Academy of Sciences, Jan 2024.

- Baidu, Jan 2024.
- New York University, Feb 2024.
- ByteDance, Feb 2024.
- Peking University Alumni Association of Nothern California, Mar 2024.
- Kunlun Tech, Mar 2024.
- California Institute of Technology, Mar 2024.
- Adobe, Apr 2024.
- Hong Kong Shanghai AI Forum, Apr 2024.

We have won the first place at the following international challenges:

- TRECVID ActEV, 2019 & 2020.
- CVPR AI City: City-Scale Multi-Camera Vehicle Tracking, 2020.
- MediaEval Sports Video: Stroke Classification, 2021.
- WACV ActEV SDL Unknown Facility, 2021.
- CVPR ActivityNet: Kinetics-700, 2021.
- ICCV ROAD: Action Detection, 2021.
- CVPR ActivityNet: ActEV, 2021 & 2022.

Applications

The technologies developed within this thesis hold the potential of revolutionizing multiple sectors, offering applications that stretch from creative content creation to enhancing communication and predictive simulations.

One of the most immediate applications lies in the realm of digital content creation. For movie makers and vloggers, the ability to generate high-fidelity videos with corresponding audio on demand could streamline the production process, reducing the need for extensive filming schedules and potentially lowering production costs. Filmmakers could leverage these models to create realistic scenes or backgrounds without the logistical challenges of on-location shoots, while vloggers might use them to enhance their content with high-quality visuals created from textual descriptions, making their production process more efficient and creative.

In the sphere of social media and digital platforms, these technologies could pioneer a new era of content discovery and engagement. By replacing traditional recommendation systems with automated content generation based on user preferences and interaction history, platforms akin to Instagram could offer an endless stream of personalized, generated content. This could not only enhance user engagement through highly customized experiences but also address some of the challenges associated with content moderation by generating safer, tailored content that adheres to platform guidelines.

Another impactful application is in communication, specifically in translating sign language into text or speech and vice versa in real time. This could dramatically improve the accessibil-

ity of digital content for the deaf and hard-of-hearing community, making information more inclusive and fostering a more connected world. The ability of these models to understand and generate video content opens the door to real-time sign language interpretation services, which could be integrated into video conferencing platforms, educational content, and public broadcasting.

Furthermore, the implications for scientific research and predictive modeling are profound. In weather forecasting, these technologies could enhance the accuracy of predictions by generating and analyzing complex weather patterns from vast datasets, potentially offering more reliable forecasts that could aid in disaster preparedness and climate research. In the realm of physics, the ability to simulate intricate systems through generated visual content could advance our understanding of phenomena that are challenging to observe directly, such as subatomic particle interactions or astronomical events.

These applications merely scratch the surface of what is possible with the advancements presented in this thesis. As these technologies continue to evolve, they will likely uncover new opportunities and challenges, pushing the boundaries of creativity, communication, and scientific exploration.

Limitations and Future Work

Limitations. While the advancements discussed in this thesis represent significant progress in the field of artificial intelligence, particularly in video generation, they are not without limitations.

- *Imperfect realism*: Models may struggle to perfectly reproduce subtle details and emotional nuances, hindering use in applications demanding absolute realism (e.g., film, VR).
- *Controllability*: Generated output may not always perfectly match complex or specific conditions in the input text. This limits use in fields requiring precise visual representation.
- *Resolution and length*: Current technology could struggle with high-resolution or long-duration videos, impeding use in industries reliant on detailed, lengthy visualizations.
- *Efficiency and cost*: The amount of compute required by current method remains infeasible to generate large-scale media content, *e.g.*, the daily uploaded amount to YouTube.
- *Data bias*: Biases in training data can perpetuate harmful stereotypes or lack of diversity within the generated content.

Disccusion. Beyond the approaches presented in this thesis, the relevant literature involves latent diffusion transformer [147], which has been promoted by the very recent release of Sora [23] for minute-long full-HD high-quality video generation. Concurrently, we have also worked on diffusion transformers in W.A.L.T [75] through a collaborative effort, utilizing the MAGVIT-v2 tokenizer. It remains challenging to compare our models with the proprietary Sora side-by-side especially due to its undisclosed training compute and dataset, which are estimated to be orders of magnitude larger.

Through the MAGVIT series, VideoPoet, and W.A.L.T, we have studied video generation with masked, autoregressive, and diffusion transformers. While it is beyond the scope of this thesis, a conceptual comparison between them appears valuable for further study.

- *Latent model*: All of these transformer models operate in a compressed latent space, where the encoder compresses and the decoder decompresses. This is partly due to the high-dimensional nature of videos and the compute cost of transformers. As we have demonstrated in this thesis, crafting high-quality latent representation is crucial for video generation. Our MAGVIT-v2 tokenizer is widely applicable for these generative transformers.
- *Architecture*: All of these transformers share a similar architecture for sequence modeling, with different input and output layers to adapt for discrete or continuous latents, according to the chosen objective. The overall training is agnostic to architecture details, where efficiency optimizations such as windowed attention or even state-space models could be adopted.
- *Continuous* vs. *discrete space*: Autoregressive transformers and masked transformers predict discrete variables, whereas diffusion models predict continuous variables. Equivalent information can be stored in either discrete or continuous representations, where preliminary MAGVIT-v2 experiments show 4 LFQ bits are roughly equal to one VAE channel. Discrete tokenizers are usually more challenging to train than continuous versions due to the non-differentiable quantization layer and the strict bandwidth. Comparing to the regression of a continuous variable with Gaussian prior, or even simpler regression of the added noise, categorical prediction of a uniform discrete variable is more challenging for the model, but may also lead to better scalability.
- *Decoding*: To produce an output of $\mathbf{x} \in \mathbb{R}^n$, diffusion decoding is factorized along an extra axis with t levels of noise, where conditional independence between the elements of $\mathbf{x}$ can be assumed given a previous prediction. On the other hand, autoregressive decoding is factorized along the axis of n, where the joint distribution is enforced by sampling one position at a time. Masked decoding can be viewed as a discrete diffusion process with absorbing noise [10], where independent samples are drawn for the positions decoded at the same step. An n-step decoding is the price that autoregressive models pay to sample from the exact joint distribution. Diffusion models approximate the joint distribution through the denoising of independent Gaussian noises, where the approximation gets more precise as $t \to n$. One nice property of diffusion decoding is that it can correct or refine the previous decoded sample, while autoregressive cannot.
- *Inference efficiency*: Despite taking n steps, the theoretical compute cost of autoregressive decoding is equivalent to just one decoding step of diffusion or masked models. However, token-by-token autoregressive decoding is usually memory-bound by the KV-cache on modern hardware, decaying the actual performance. On the other hand, diffusion decoding is compute-bound since it runs multiple full forward passes without caching.
- *Training efficiency*: Autoregressive transformer has the best training efficiency among these models, as one forward pass can cover the entire decoding trajectory thanks to the causal attention mask. Diffusion and masked transformers usually sample a random noise ratio for each forward pass, which corresponds to one decoding step.

- *Causality*: Diffusion and masked transformers adopt bidirectional attention for full dependency, whereas autoregressive models only allow the prediction to depend on the past. While enforcing a raster-scan order for spatial autoregressive modeling may not be the optimal solution, temporal causality remains critical for long generation and streamed generation in interactive setups.
- *Unification*: Autoregressive video generation models such as VideoPoet can be easily unified with existing LLM approaches and benefit from the relevant optimizations, which is one of the top advantages of this approach. It remains challenging to unify diffusion and LLM into one model.

Suggestions. As technologies evolve, future approaches are likely to appear differently from what we have presented in this thesis. However, a few suggestions from our experiences might remain helpful for a longer period of time than the methodologies themselves.

- *Transformers* are never designed to produce a large amount of data. Even with an optimistic estimation of 10M sequence length with 2^{20} vocabulary size, the model can only produce 25MB data. This is still far from an hour-long full-HD video which can be several gigabytes. Therefore, either a new architecture or a highly-compressed latent is required.
- *Hardware-software co-design* could play an important role for new techniques to be deployed at scale. For example, hardware accelerators for video codecs could be a potential solution to improve generation efficiency by 100×. In addition, the coarse-to-fine setup in discrete cosine transformation could be a better order than the raster scan.
- *Modality unification* is worth exploring where one model should be able to jointly learn the generation of multiple modalities. There is a huge space to search for the optimal learning recipe which should take care of the inter-modal interaction and also enable post-training addition of new modalities at minimal costs.
- *Realtime interactive* video generation, if made possible, could be applied to a wide array of scenarios, including gaming, story telling, software UI, world simulator, *etc.* This could become the next major milestone in the history of computer software.
- *Embodied intelligence* could benefit from neural simulators and end-to-end controller with visual world knowledge, both enabled via generative video learning. As LLMs become widely deployed, robotics might be a great direction for next-generation PhD students.
- *Foundation models* should learn from raw signals like video and audio rather than the distilled human knowledge from text. The endless sources for these raw signals could enable better scalability than exhausted text data. Models can be pretrained without text and just aligned with text for human interaction. This sets up a better stage to enable the discovery of new rules of the universe without restrictions from human prior.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8M: A large-scale video classification benchmark. *arXiv:1609.08675*, 2016. 46

[2] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced wasserstein GANs. *arXiv:1810.02419*, 2018. 88, 97

[3] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating music from text. *arXiv:2301.11325*, 2023. 64, 132

[4] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *ICCV*, 2019. 69, 79

[5] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. 86, 95, 103, 106

[6] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 113

[7] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-

Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. PaLM 2 technical report. *arXiv:2305.10403*, 2023. 3, 26, 64, 112, 113, 117, 132

[8] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. DocFormer: End-to-end transformer for document understanding. In *ICCV*, 2021. 26, 27, 31

[9] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *ICCV*, 2021. 79, 80

[10] Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021. 88, 161

[11] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. *TRECVID*, 2020. 19, 21

[12] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, Gareth J. F. Jones, and Georges Quénot. Evaluating multiple video understanding and retrieval tasks at TRECVID 2021. In *TRECVID*, 2021. 21

[13] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in pixel-level video prediction. *arXiv:2106.13195*, 2021. 88, 99

[14] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2021. 65, 79

[15] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127*, 2023. 134

[16] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent

diffusion models. In *CVPR*, 2023. 67, 134, 143, 150

[17] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021. 3, 132

[18] Łukasz Borchmann, Michał Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. DUE: End-to-end document understanding benchmark. In *NeurIPS*, 2021. 27

[19] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023. 50

[20] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`. 95

[21] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2018. 76, 88

[22] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A. Efros, and Tero Karras. Generating long videos of dynamic scenes. In *NeurIPS*, 2022. 88

[23] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. URL `https://openai.com/`

research/video-generation-models-as-world-simulators. 160

[24] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 65, 77, 79

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 3, 64, 88, 112, 113, 132, 134

[26] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 27

[27] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 86, 95, 103, 106

[28] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial intelligence*, 134(1-2):57–83, 2002. 3

[29] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *CVPR*, 2017. 41, 44, 95

[30] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 55, 72, 86, 95, 139

[31] Duygu Ceylan, Chun-Hao P. Huang, and Niloy J. Mitra. Pix2Video: Video editing using image diffusion. In *CVPR*, 2023. 132

[32] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. StableVideo: Text-driven consistency-aware diffusion video editing. In *CVPR*, 2023. 132

[33] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *CVPR*, 2022. 40, 44, 45, 50, 55, 56, 64, 66, 69, 71, 75, 76, 86, 88, 89, 91, 92, 93, 104, 107, 114, 134, 143

[34] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *ICML*, 2023. 55, 56, 66, 67, 134

[35] Xiaojun Chang, Wenhe Liu, Po-Yao Huang, Changlin Li, Fengda Zhu, Mingfei Han, Mingjie Li, Mengyuan Ma, Siyi Hu, Guoliang Kang, Junwei Liang, Liangke Gui, Lijun Yu, Yijun Qian, Jing Wen, and Alexander G. Hauptmann. MMVG-INF-Etrol@ TRECVID 2019: Activities in extended video. In *TRECVID*, 2019. 13

[36] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu

Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv:2310.19512*, 2023. 143

[37] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 40, 65, 88

[38] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-A-Video: Controllable text-to-video generation with diffusion models. *arXiv:2305.13840*, 2023. 132, 134

[39] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2022. 71

[40] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *JMLR*, 24(240):1–113, 2023. 64, 112, 113, 132, 134

[41] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, 2016. 88, 96

[42] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv:1907.06571*, 2019. 86, 88, 97, 99, 107

[43] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. MEVA: A large-scale multiview, multimodal video dataset for activity detection. In *WACV*, 2021. 12, 19

[44] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016. 13

[45] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022. 64

[46] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *TMLR*, 2023. 50

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 27, 44, 55, 72

[48] Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 114, 117, 119

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 3, 26, 31, 32, 40, 50, 64, 66, 74, 86, 89, 95

[50] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 66, 75, 76

[51] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 40, 86, 88, 132

[52] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3, 31, 55, 95

[53] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *ICML*, 2023. 132

[54] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaMs: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022. 64, 132

[55] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, 2017. 86, 95

[56] Arpad E. Elo and Sam Sloan. *The rating of chessplayers : past and present.* Ishi Press International, 2008. 77

[57] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 40, 41, 50, 54, 64, 66, 68, 69, 76, 86, 88, 89, 136

[58] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *CVPR*, 2023. 134

[59] Gustav Theodor Fechner. *Elemente der psychophysik*, volume 2. Breitkopf u. Härtel, 1860.

77

[60] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 12, 13, 19

[61] Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. CCEdit: Creative and controllable video editing via diffusion models. *arXiv:2309.16496*, 2023. 134

[62] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, 2023. 64, 75, 76

[63] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic VQGAN and time-sensitive transformer. In *ECCV*, 2022. 40, 41, 42, 44, 45, 50, 54, 67, 74, 86, 87, 88, 96, 97, 98, 107

[64] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*, 2023. 67, 134, 150

[65] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024. 132

[66] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 12, 19

[67] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *EMNLP-IJCNLP*, 2019. 89

[68] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 72, 86, 95, 105, 139

[69] Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *ACL-IJCNLP Findings*, 2021. 89

[70] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. UniDoc: Unified pretraining framework for document understanding. In *NeurIPS*, 2021. 26, 27, 31

[71] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022. 86, 89

[72] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *ICML*, 2020. 29

[73] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 134

[74] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked visual pre-training for video prediction. In *ICLR*, 2022. 40, 67, 86, 88, 96, 99, 104, 107

[75] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv:2312.06662*, 2023. 158, 160

[76] Sonam Gupta, Arti Keshari, and Sukhendu Das. RV-GAN: Recurrent GAN for unconditional video generation. In *CVPRW*, 2022. 88

[77] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas, and Sergey Tulyakov. Show me what and tell me how: Video synthesis via multimodal conditioning. In *CVPR*, 2022. 86, 88

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 19, 40

[79] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 19

[80] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 32

[81] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv:2211.13221*, 2023. 134

[82] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL `http://github.com/google/flax`. 95

[83] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv:1606.08415*, 2016. 41

[84] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017. 141

[85] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 55, 75

[86] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 75, 76, 138, 150

[87] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 66, 88

[88] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen Video: High definition video generation with diffusion models. *arXiv:2210.02303*, 2022. 66, 67, 88, 134

[89] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 23 (1):2249–2281, 2022. 76

[90] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *ICLR Workshops*, 2022. 66, 74, 86, 87, 88, 96, 97, 99, 103, 150

[91] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv:2203.15556*, 2022. 134

[92] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*, 2022. 27

[93] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. In *ICLR*, 2023. 88, 96, 97, 99, 132, 134, 143

[94] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 66, 75, 76

[95] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *TMLR*, 2022. 86, 87, 97, 99, 100, 101

[96] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICLR*, 2019. 113

[97] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv:2309.17080*, 2023. 134

[98] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2021. 113

[99] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for document AI with unified text and image masking. In *ACM MM*, 2022. 26, 28, 31

[100] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. ICDAR2019 competition on scanned receipt OCR and information extraction. In *ICDAR*, 2019. 27

[101] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 41

[102] Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *ICML*, 2023. 66, 74, 75, 76

[103] Aren Jansen, Daniel P. W. Ellis, Shawn Hershey, R. Channing Moore, Manoj Plakal, Ashok C. Popat, and Rif A. Saurous. Coincidence, categorization, and consolidation: Learning to recognize sounds with minimal supervision. In *ICASSP*, 2020. 69

[104] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. FUNSD: A dataset for form understanding in noisy scanned documents. In *ICDAR Workshops*, volume 2, 2019. 26, 27

[105] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 54

[106] Norman P. Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B. Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, Thomas Norrie, Nishant Patil, Sushma Prasad, Cliff Young, Zongwei Zhou, and David Patterson. Ten lessons from three generations shaped google's TPUv4i. In *ISCA*, 2021. 86, 96

[107] Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural Networks*, 132:506–520, 2020. 88, 97

[108] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 41, 70

[109] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 12, 19, 72

[110] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 34, 55, 95

[111] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. *arXiv:2303.00848*, 2023. 75, 76

[112] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *ICML*, 2023. 112, 113

[113] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. VideoPoet: A large language model for zero-shot video generation. *arXiv:2312.14125*, 2023. 157

[114] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. BLT: Bidirectional layout transformer for controllable layout generation. In *ECCV*, 2022. 89

[115] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, 2018. 116

[116] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. CCVS: Context-aware controllable video synthesis. In *NeurIPS*, 2021. 40, 87, 96, 97, 98, 99

[117] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 3

[118] Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. FormNet: Structural encoding beyond sequential modeling in form document information extraction. In *ACL*, 2022. 26

[119] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 52, 53, 58, 60

[120] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Draft-and-revise: Effective image generation with contextual RQ-transformer. In *NeurIPS*, 2022. 64, 76

[121] Yooyoung Lee, Jonathan Fiscus, Andrew Delgado, Lukas Diduch, Eliot Godard, Baptiste Chocot, Jesse Zhang, Jim Golden, Afzal Godil, and Diane Ridgeway. Actev 2021 sequestered data leaderboard (SDL) evaluation plan. *NIST*, 2021. 21

[122] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 64, 113

[123] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *ACM SIGIR*, 2006. 26, 27

[124] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *NeurIPS*, 2022. 113

[125] José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with Token-Critic. In *ECCV*, 2022. 76, 88, 89

[126] José Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *ICLR*, 2023. 55, 56, 74, 75, 76

[127] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I. Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. SelfDoc: Self-supervised document representation learning. In *CVPR*, 2021. 28, 31

[128] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Ku-

nakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. StarCoder: may the source be with you! *TMLR*, 2023. 64, 132

[129] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. MagicEdit: High-fidelity and temporally coherent video editing. *arXiv:2308.14749*, 2023. 134

[130] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 13

[131] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 19, 119

[132] Hao Liu, Wilson Yan, and Pieter Abbeel. Language quantized autoencoders: Towards unsupervised text-image alignment. In *NeurIPS*, 2023. 50, 55, 112, 114, 117, 118, 119, 157

[133] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2024. 3

[134] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, and Alexander G. Hauptmann. Argus: Efficient activity detection system for extended video analysis. In *WACVW*, 2020. 12, 13

[135] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 3

[136] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv:2003.04035*, 2020. 74, 86, 99

[137] Chengzhi Mao, Lu Jiang, Mostafa Dehghani, Carl Vondrick, Rahul Sukthankar, and Irfan Essa. Discrete representations strengthen vision transformer robustness. In *ICLR*, 2021. 65

[138] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for VQA on document images. In *WACV*, 2021. 27

[139] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955. *AI magazine*, 27(4):12–12, 2006. 3

[140] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 134

[141] Jackson Sam Michaels, Juncheng B Li, Laura Yao, Lijun Yu, Zach Wood-Doughty, and Florian Metze. Audio-Journey: Open domain latent diffusion based text-to-audio generation. In *ICASSP*, 2024. 158

[142] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 27

[143] Charlie Nash, João Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *TMLR*, 2023. 86, 88, 99, 132, 134

[144] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 12, 19

[145] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny,

Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 technical report. *arXiv:2303.08774*, 2023. 3, 26, 64, 112, 132

[146] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. CORD: a consolidated receipt dataset for post-OCR parsing. In *NeurIPS Workshops*, 2019. 26, 27

[147] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 66, 75, 76, 160

[148] Pika. Pika 1.0, 2023. URL `https://pika.art`. 143

[149] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 132

[150] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *ICDAR*, 2021. 27

[151] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *EACL*, 2017. 71

[152] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. ELECTRICITY: An efficient multi-camera vehicle tracking system for intelligent city. In *CVPRW*, 2020. 19, 158

[153] Yijun Qian, Lijun Yu, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Adaptive feature aggregation for video object detection. In *WACVW*, 2020. 13

[154] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Rethinking zero-shot action recognition: Learning from latent atomic actions. In *ECCV*, 2022. 13, 158

[155] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. TRM: Temporal relocation module for video recognition. In *WACVW*, 2022. 19

[156] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 29, 53

[157] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020. 132, 134, 136, 147

[158] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *VISIGRAPP (5: VISAPP)*, 2021. 86, 88, 99

[159] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. In *ICLR Workshops*, 2018. 41

[160] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 40, 65, 86, 88, 89, 113, 134

[161] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 2022. 67, 132

[162] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *NeurIPS*, 2015. 119

[163] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 13, 16

[164] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan R Dave, Yogesh S Rawat, and Mubarak Shah. Gabriella: An online system for real-time activity detection in untrimmed security videos. In *ICPR*, 2021. 12, 13

[165] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 50, 55, 56, 64, 66, 76, 86, 113, 132

[166] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 66

[167] Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirović, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Frank. AudioPaLM: A large language model that can speak and listen. *arXiv:2306.12925*, 2023. 64, 132

[168] Runway. Gen-2, 2023. URL `https://research.runwayml.com/gen2`. 143

[169] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffu-

sion models with deep language understanding. In *NeurIPS*, 2022. 66, 134

[170] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 86, 88, 97

[171] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *IJCV*, 128(10):2586–2606, 2020. 87, 95, 97, 139, 150

[172] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016. 55, 75

[173] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959. 3

[174] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling stylegan to large diverse datasets. In *SIGGRAPH*, 2022. 75, 76

[175] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35, 2022. 151

[176] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv:1911.02150*, 2019. 64

[177] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-Video: Text-to-video generation without text-video data. *arXiv:2209.14792*, 2022. 66, 67, 88, 96, 97, 132, 134, 143

[178] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Salman Khan, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, Jordan Omokeowa, Stanislao Grazioso, Andrew Bradley, Giuseppe Di Gironimo, and Fabio Cuzzolin. ROAD: The road event awareness dataset for autonomous driving. *IEEE TPAMI*, 45(1):1036–1054, 2022. 12, 22

[179] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Towards expert-level medical question answering with large language models. *arXiv:2305.09617*, 2023. 64

[180] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In *CVPR*, 2022. 88, 97

[181] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 66, 88

[182] Kihyuk Sohn, Yuan Hao, José Lezama, Luisa Polania, Huiwen Chang, Han Zhang, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. In *CVPR*, 2023. 88

[183] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 66, 88

[184] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 41, 72, 86, 95, 96, 139, 143, 150

[185] Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: key information extraction datasets involving long documents with complex layouts. In *ICDAR*, 2021. 27

[186] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 65, 77, 79

[187] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *ICLR*, 2022. 88, 158

[188] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. In *NeurIPS*, 2023. 132

[189] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 88, 97

[190] Aleksei Timofeev, Andrew Tomkins, Chun-Ta Lu, Da-Cheng Juan, Futang Peng, Krishnamurthy Viswanathan, Lucy Gao, Sujith Ravi, Tom Duerig, Yi ting Chen, and Zhen Li. Graph-RISE: Graph-regularized image semantic embedding. In *ACM WSDM*, 2020. 29

[191] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 3, 64

[192] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 95

[193] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 12, 13, 19

[194] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *CVPR*, 2021. 41, 54, 69

[195] Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, 2021. 118, 119

[196] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing

motion and content for video generation. In *CVPR*, 2018. 86, 88, 97

[197] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv:1905.09883*, 2019. 88

[198] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv:1812.01717*, 2018. 55, 74, 86, 87, 95, 96, 139, 151

[199] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 88

[200] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 40, 50, 51, 65, 66, 89

[201] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 40, 112, 134

[202] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv:2211.04325*, 2022. 3

[203] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. In *ICLR*, 2023. 50, 66, 67, 68, 69, 74, 88, 99, 132, 134, 136, 143, 150

[204] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016. 115, 117

[205] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022. 86, 99, 132

[206] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 86, 88, 97

[207] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*, 2019. 113

[208] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *ICIP*, 2016. 72, 77

[209] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. ModelScope text-to-video technical report. *arXiv:2308.06571*, 2023. 134, 143

[210] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 12, 17, 19

[211] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-

Gang Jiang, Luowei Zhou, and Lu Yuan. BEVT: BERT pretraining of video transformers. In *CVPR*, 2022. 65, 79, 80

[212] Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv:2303.17599*, 2023. 132

[213] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEiT pretraining for all vision and vision-language tasks. In *CVPR*, 2023. 86

[214] Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. VideoFactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv:2305.10874*, 2023. 134, 143

[215] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. InternVid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024. 134, 143

[216] Ze Wang, Jiang Wang, Zicheng Liu, and Qiang Qiu. Binary latent diffusion. In *CVPR*, 2023. 66, 76

[217] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 13, 19

[218] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 95

[219] Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. VRDU: A benchmark for visually-rich document understanding. In *ACM SIGKDD*, 2023. 27

[220] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022. 79

[221] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *TMLR*, 2022. 112

[222] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2019. 86, 88, 99

[223] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 13

[224] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. GODIVA: Generating open-domain videos from natural descriptions. *arXiv:2104.14806*, 2021. 139

[225] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan.

NÜWA: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 67, 87, 88, 99

[226] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual ChatGPT: Talking, drawing and editing with visual foundation models. *arXiv:2303.04671*, 2023. 112

[227] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 41

[228] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 19

[229] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 139, 143, 150

[230] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multimodal pre-training for visually-rich document understanding. In *ACL-IJCNLP*, 2021. 27, 31

[231] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of text and layout for document image understanding. In *ACM SIGKDD*, 2020. 27, 31

[232] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv:2104.10157*, 2021. 40, 67, 88, 97, 99, 134

[233] Pengcheng Yin, Wen-Ding Li, Kefan Xiao, Abhishek Rao, Yeming Wen, Kensen Shi, Joshua Howland, Paige Bailey, Michele Catasta, Henryk Michalewski, Alex Polozov, and Charles Sutton. Natural language to code generation in interactive data science notebooks. In *ACL*, 2023. 113

[234] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022. 67, 68, 89

[235] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022. 40, 65, 86, 88, 132, 134, 141

[236] Lijun Yu, Dawei Zhang, Xiangqun Chen, and Alexander G. Hauptmann. Traffic danger recognition with surveillance cameras without training data. In *AVSS*, 2018. 13, 157

[237] Lijun Yu, Peng Chen, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Training-free monocular 3D event detection system for traffic surveillance. In *Big Data*, 2019. 13, 157

[238] Lijun Yu, Qianyu Feng, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Zero-VIRUS: Zero-shot vehicle route understanding system for intelligent transportation. In *CVPRW*, 2020. 19, 158

[239] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. CMU Informedia at

TRECVID 2020: Activity detection with dense spatio-temporal proposals. In *TRECVID*, 2020. 12, 13, 21, 157

[240] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. CMU Informedia at TRECVID 2021: Activity Detection with Argus++. In *TRECVID*, 2021. 21, 158

[241] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Argus++: Robust real-time activity detection for unconstrained video streams with overlapping cube proposals. In *WACVW*, 2022. 155

[242] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. MAGVIT: Masked generative video transformer. In *CVPR*, 2023. 40, 46, 50, 54, 55, 57, 64, 65, 66, 67, 69, 72, 74, 75, 79, 134, 136, 139, 147, 151, 156

[243] Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David A. Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, Kevin Murphy, Alexander G. Hauptmann, and Lu Jiang. SPAE: Semantic pyramid autoencoder for multimodal generation with frozen llms. In *NeurIPS*, 2023. 136, 156

[244] Lijun Yu, Jin Miao, Xiaoyu Sun, Jiayi Chen, Alexander G. Hauptmann, Hanjun Dai, and Wei Wei. DocumentNet: Bridging the data gap in document pre-training. In *EMNLP-Industry*, 2023. 156

[245] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation. *arXiv:2310.05737*, 2024. 50, 134, 136, 147, 157

[246] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, Candace Ross, Adam Polyak, Russell Howes, Vasu Sharma, Puxin Xu, Hovhannes Tamoyan, Oron Ashual, Uriel Singer, Shang-Wen Li, Susan Zhang, Richard James, Gargi Ghosh, Yaniv Taigman, Maryam Fazel-Zarandi, Asli Celikyilmaz, Luke Zettlemoyer, and Armen Aghajanyan. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv:2309.02591*, 2023. 67

[247] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 88, 97

[248] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *CVPR*, 2023. 132

[249] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 30:495–507, 2021. 52, 53, 136

[250] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. *arXiv:2311.10982*, 2023. 134

[251] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv:2309.15818*, 2023. 132, 134, 143

[252] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023. 132, 134

[253] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 70

[254] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 55, 95

[255] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv:2205.01068*, 2022. 113

[256] Zhu Zhang, Jianxin Ma, Chang Zhou, Rui Men, Zhikang Li, Ming Ding, Jie Tang, Jingren Zhou, and Hongxia Yang. M6-UFC: Unifying multi-modal controls for conditional image synthesis. *arXiv:2105.14211*, 2021. 88, 89

[257] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *TMLR*, 2024. 76

[258] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: largest dataset ever for document layout analysis. In *ICDAR*, 2019. 27

[259] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less is more for alignment. In *NeurIPS*, 2023. 138

[260] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. MagicVideo: Efficient video generation with latent diffusion models. *arXiv:2211.11018*, 2022. 134, 143

[261] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 13

[262] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023. 64, 132