On the Limits of Token Reduction for Efficient Unified Vision Language Training

Anonymous CVPR submission

Paper ID *****

Abstract

Unified vision-language models (VLMs) integrate visual understanding and visual generation within a single autoregressive backbone, but their joint training is computationally expensive and largely overlooked from an efficiency perspective. In this work, we study the feasibility and limits of token-reduction-based acceleration for unified VLM training. Through a systematic analysis of layerwise attention allocation, we uncover a fundamental asymmetry: visual understanding exhibits substantial late-layer visual redundancy, whereas visual generation maintains persistent dependence on image tokens across depth. Guided by this observation, we design task-specific accelerators that selectively reduce image-token computation for each objective. While these methods achieve significant efficiency gains in isolated settings, we observe a consistent synergy loss under unified training—task-specific token dropping necessitates divergent parameter pathways and eliminates the mutual performance gains typically observed in joint optimization. Our findings suggest that efficient unified modeling requires preserving shared cross-task structures, highlighting the need for synergy-aware acceleration strategies.

1. Introduction

Unified Vision-Language Models (VLMs) [11, 25, 26, 36, 38, 39] integrate visual generation [6, 7, 29, 33] and understanding [4, 21, 22, 27] within a single model and have demonstrated remarkable scalability and cross-task potential [32, 37, 39]. However, the training of these models is prohibitively expensive; for instance, VILA-U [39] requires approximately 20K A100 GPU hours. While many prior methods propose to reduce inference-time computation in understanding-only VLMs via token pruning or special attention masks [1, 3, 12, 24, 28, 30, 44], these strategies do not directly translate to improve training-time efficiency. Furthermore, existing acceleration techniques for visual understanding do not account for the distinct structural requirements of visual generation, nor do they study the complexities inherent in unifying generative and discriminative objectives within a single VLM.

In this paper, we investigate the feasibility and limits of accelerating the training of unified vision language models. We adopt the pure autoregressive framework as our testbed, as it represents one of the most prevalent architectures for integrating multimodal capabilities [9, 14, 23, 36, 39, 41–43]. Through an analysis of the attention dynamics within this framework (in Figure 2), we reveal a critical asymmetry in task-specific redundancy: while visual understanding tasks exhibit high token redundancy in the deeper layers, visual generation depends heavily on the context of previously generated image tokens within many deep layers. Building on these insights, we develop task-specific strategies to accelerate training by selectively dropping image tokens tailored to the unique requirements of each objective.

Furthermore, we reveal a critical ”synergy loss” phenomenon that occurs when task-specific token reduction methods are applied to the joint training of unified models. We find that task-specific token dropping disrupts the inherent synergy between understanding and generation by: (1) necessitating divergent sets of image-related model parameters, and (2) eliminating the mutual performance gains typically observed when both tasks are trained concurrently. Our diagnostic analysis suggests that aggressive token dropping amplifies task conflicts, offering a cautionary lesson and a new perspective for future research in efficient unified modeling. Our contributions are summarized as follows:

- **Unified Redundancy Analysis**: We characterize task-specific attention patterns in unified VLMs, identifying distinct redundancy zones.
- **Task-Specific Accelerators**: We design and implement training-time acceleration for isolated tasks.
- **Discovery of Synergy Loss**: We discover that task-specific optimization strategies fail in unified settings, revealing that forced token reduction disrupts mutual improvements of discriminative and generative objectives.
- **Lessons for Unified Acceleration**: Our results suggest that effective acceleration methods may benefit from preserving shared cross-task structures and carefully accounting for impact on cross-task learning dynamics.

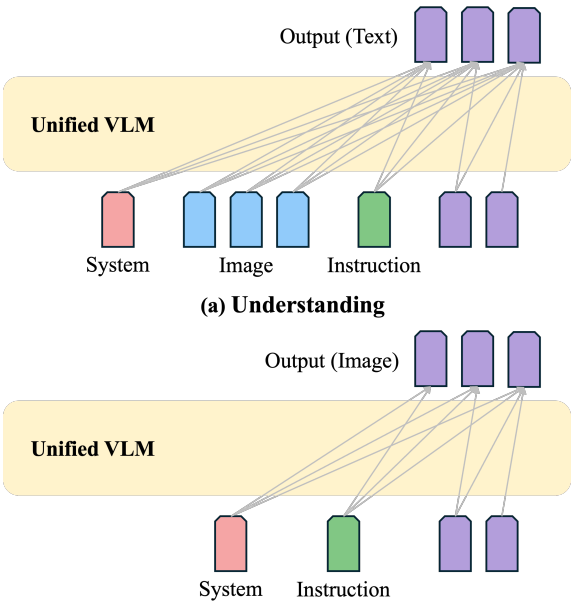

Figure 1. **Unified autoregressive VLM**. A single Transformer backbone processes multimodal sequences under a unified next-token prediction objective. (a) In visual understanding, the model predicts text tokens conditioned on image and textual context. (b) In visual generation, the model autoregressively predicts image tokens conditioned on preceding text and image tokens.

2. Related Works

Unified Vision-Language Models. Recent advancements have shifted toward unifying perception and generation within a single framework. Models like VILA-U [39], Janus [37], and Chameleon [32] utilize discrete visual tokenizers (e.g., VQVAE [35]) to treat images as a "foreign language." While these models simplify the pipeline by using a single next-token prediction objective, their joint training is computationally demanding. Many other hybrid models that append diffusion heads [29] to a transformer also require fine-tuning the entire backbone across multiple modalities, creating the need for efficient training.

Efficiency in Vision-Language Models. Efficiency research in VLMs has primarily focused on visual understanding during inference-time [3, 12, 17, 20, 24, 30, 45]. For instance, LLaVA-PruMerge [30] and LLaMA-VID [19] reduce the number of visual tokens by identifying spatial redundancy and merging tokens. Other works explore efficient attention mechanisms or special masks to skip redundant computations during inference [44, 46]. However, these methods are often designed for "understanding-only" tasks where the model's output is limited to text, and a complete set of image tokens is treated as input, thus having difficulty applying to visual generation, and how to reduce training-time computation remains a challenging problem.

Token Reduction and Attention Redundancy. The concept of "token reduction" or "pruning" originates from the Vision Transformer (ViT) and NLP literature to handle long-sequence data [1, 10, 28, 40]. These methods typically use attention weights or activation statistics as proxies for token importance. In the multimodal domain, recent studies have analyzed attention sinks [40] and sparsity to prune background patches. While effective for single-task models, these importance metrics are not directly transferable to unified models where tokens must serve dual roles in discriminative perception and generative synthesis.

Multi-task Synergy in VLMs. The relationship between understanding and generation has been a subject of ongoing debate. While some studies suggest that generative pre-training provides a stronger world model for perception [36, 41], others have noted the difficulty of balancing these disparate objectives during joint optimization [37]. We build upon this line of inquiry by investigating how structural constraints—specifically, token dropping—affect the stability and synergy of this multi-task learning process.

3. Problem Setup

3.1. Unified Autoregressive Vision-Language Model

We study a unified vision-language model (VLM) that jointly performs visual understanding and visual generation within a single autoregressive Transformer backbone, following the unified next-token prediction paradigm of VILA-U (7B) [39]. Let $x = (x_1, \ldots, x_T)$ denote text tokens and $v = (v_1, \ldots, v_M)$ denote discrete image tokens obtained from a visual tokenizer (e.g., VQ-based). We construct a multimodal sequence

$$z = (z_1, \ldots, z_{|z|}) = (\text{system}, x, v).$$

The model, parameterized by θ, is trained using autoregressive next-token prediction:

$$P_\theta(z) = \prod_{t=1}^{|z|} P_\theta(z_t \mid z_{<t}).$$

Under this formulation, both text and image tokens are treated uniformly as discrete tokens in a single sequence, and a shared next-token objective is applied across modalities. We visualize the generation process in Figure 1.

3.2. Training Objective

Unified training mixes data from visual understanding and visual generation tasks under a single objective. For a multimodal training sample z, the loss is defined as the negative log-likelihood:

$$\mathcal{L}_{\text{unified}} = -\sum_{t=1}^{|z|} \log P_\theta(z_t \mid z_{<t}). \quad (1)$$

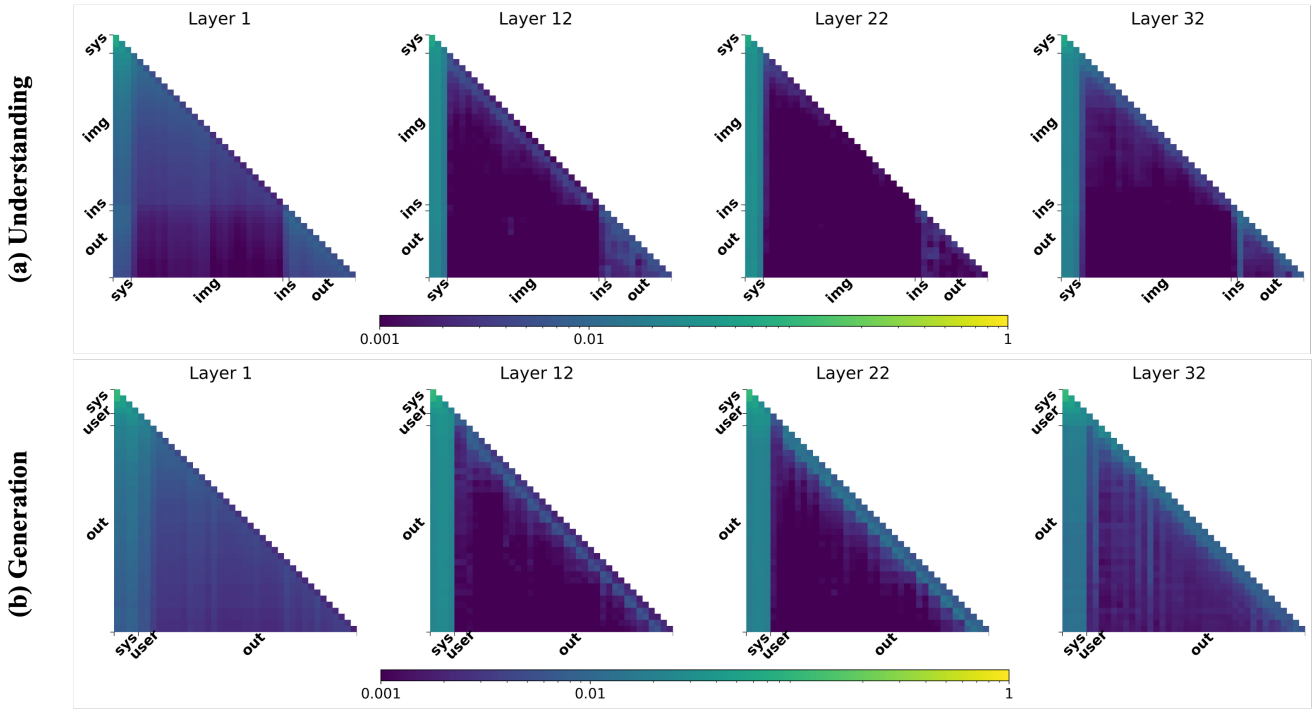

Figure 2. **Asymmetric depth-wise attention patterns in unified VLMs.** Visualization of self-attention heatmaps across layers for understanding (a) and generation (b). Understanding exhibits strong early cross-modal interactions followed by a sharp decay in image-token attention. Generation, however, preserves substantial image-token attention throughout depth, highlighting a fundamental asymmetry in token utilization.

This unified objective simultaneously optimizes: **Visual Understanding:** predicting text tokens conditioned on image and textual context; **Visual Generation:** predicting image tokens conditioned on preceding text and image tokens.

3.3. Computational Bottleneck

Let $N = |z|$ denote the total sequence length. A standard Transformer layer incurs quadratic self-attention cost: FLOPs per layer $\propto N^2$. Since $N = T + M$, where T and M are the numbers of text and image tokens respectively,

$$(T + M)^2 = T^2 + 2TM + M^2.$$

In unified VLM training, image tokens typically dominate the sequence ($M \gg T$), making the M^2 term the primary computational bottleneck.

This naturally motivates token reduction strategies that limit the effective participation of image tokens in attention. In this work, we investigate the effectiveness and limitations of *token-reduction-based training acceleration* for visual understanding, visual generation, and their unification. We begin by analyzing task-specific redundancy patterns through attention statistics.

4. Redundancy Analysis

To guide the design of our acceleration strategies, we analyze the layerwise attention behavior of a pre-trained unified VLM. Our goal is to analyze task-specific redundancy in visual tokens that inspires method design.

4.1. Analysis Setup

Model and Data. We analyze the VILA-U model and collect attention statistics on both visual understanding (with ShareGPT-4v dataset) and visual generation (with JournyDB dataset). For each task, we record attention maps across all transformer layers.

Attention Allocation. Following prior attention decomposition analysis [3], we measure how attention mass is distributed across token segments. Let $A_{i,j}^{(\ell,h)}$ denote the attention weight at layer ℓ and head h from query token i to key token j, with

$$\sum_j A_{i,j}^{(\ell,h)} = 1.$$

Given a partition of tokens into segments (e.g., `system`, `image`, `instruction`, `output`), the *attention allocation* of segment S at layer ℓ is defined as:

$$\alpha_S^{(\ell)} = \frac{1}{H} \sum_{h=1}^{H} \sum_i \sum_{j \in S} A_{i,j}^{(\ell,h)}. \quad (2)$$

This metric captures the fraction of total attention mass directed to each token segment at a given layer. We use $\alpha_S^{(\ell)}$ to quantify redundancy patterns across depth.

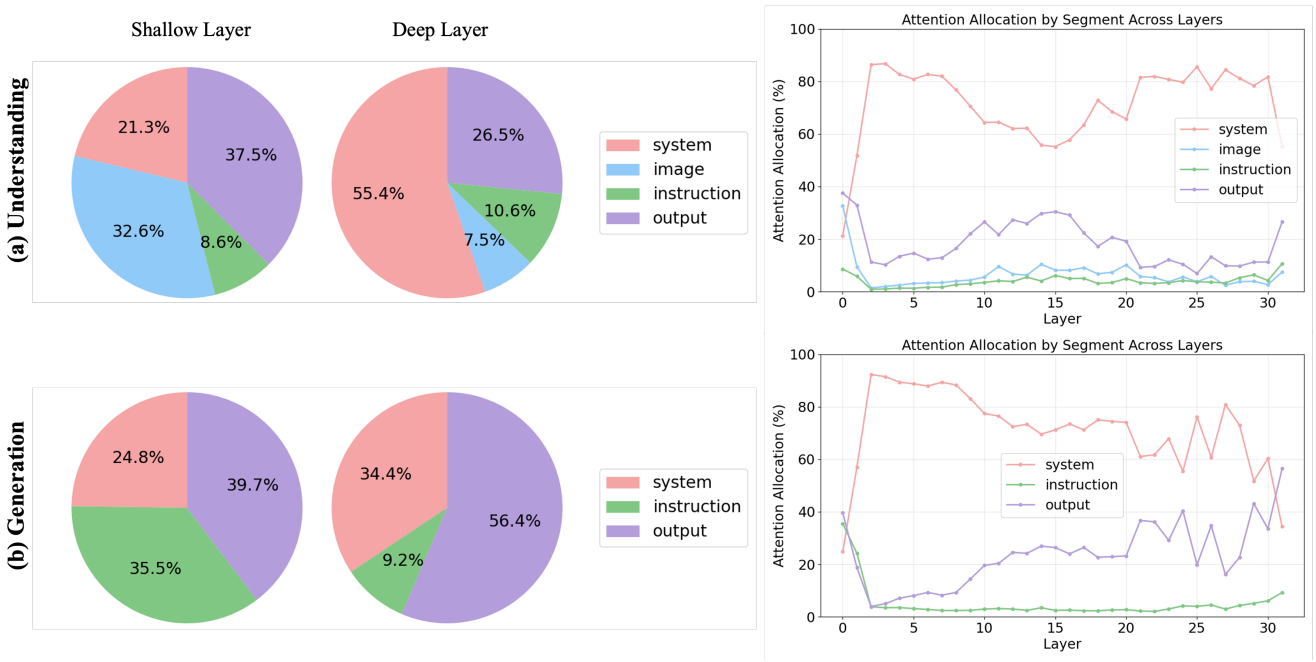

Figure 3. **Quantitative attention allocation reveals depth-dependent visual redundancy asymmetry.** Left: Attention mass distribution over token segments (system, image, instruction, output) at representative shallow and deep layers. Right: Layerwise attention allocation across the full transformer depth. For visual understanding (top), attention to image tokens sharply decreases in deeper layers, while instruction and output tokens dominate, indicating substantial late-layer visual redundancy. In contrast, visual generation (bottom) maintains consistently high attention to image tokens across layers, with increasing allocation to output image tokens in deep layers, reflecting persistent autoregressive dependence on generated image tokens.

4.2. Task-Specific Attention Patterns

We visualize (1) attention allocation (Figure 3) across token segments over layers and (2) attention heatmaps (Figure 2) at representative layers for both visual understanding (U) and visual generation (G). The results reveal a clear asymmetry in visual token redundancy patterns in different tasks.

Visual Understanding (U). For perception tasks (e.g., VQA), visual tokens exhibit clear depth-dependent redundancy. As shown in Figure 2 and Figure 3, attention rapidly shifts away from image tokens as depth increases. Image tokens account for roughly $\sim$30% of attention in the first layer, but this drops below 10% in middle and late layers. Instead, attention becomes dominated by instruction and output tokens, which together exceed 80% of the total attention mass in deeper layers. Across layers, we observe a consistent transition:

- **Early layers:** Strong cross-modal interactions between image and text tokens, indicating visual grounding and alignment.
- **Middle layers:** Attention increasingly concentrates on text tokens, with diminishing image-to-image and image-to-text interactions.
- **Late layers:** Attention is almost entirely confined to textual tokens, suggesting that high-level reasoning becomes predominantly linguistic.

Visual Generation (G). In contrast to understanding, image generation exhibits a persistent and structured dependence on image tokens. Output (image) tokens receive a substantial fraction of attention across early and late layers, typically ranging from 30% to 60%. Unlike the rapid decay observed in understanding tasks, attention to image tokens exhibits a consistent increase of attention allocation in deeper layers. Across depth, the attention pattern follows a hierarchical structure:

- **Early layers:** Broad attention over both textual prompts and previously generated image tokens, establishing global conditioning.
- **Middle layers:** Attention concentrates on recent image tokens and specific prefix positions, reflecting localized autoregressive dependencies.
- **Late layers:** Image-token attention becomes increasingly significant, ensuring consistency in token prediction.

Robustness Across Scales. We repeat the same analysis on a smaller-scale VILA-U model trained by ourselves (LLaMA-3-3B backbone [5]). The qualitative and quantitative patterns remain consistent: late-layer visual redundancy emerges for understanding, while generation preserves significant image-token attention. This suggests the observed asymmetry is not scale-specific.

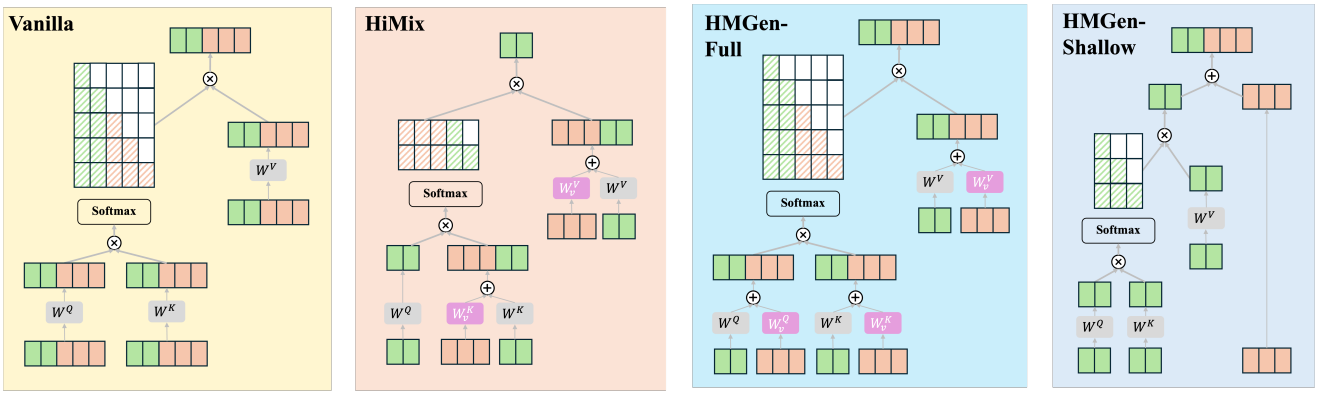

Figure 4. **Task-specific token-reduction-based acceleration mechanisms for unified VLM training.** From left to right: (1) Vanilla Transformer layer, where both text and image tokens participate fully in self-attention and feed-forward computation. (2) HiMix (Understanding) reduce image tokens from the query stream while retaining them in key/value projections, eliminating quadratic image-to-image attention while preserving text-to-image interactions. (3) HMGen–Full layer (Generation) maintains full autoregressive attention but separates image- and text-related projections for stable hierarchical conditioning. (4) HMGen–Shallow layer (Generation) skips image-token attention and feed-forward updates, forwarding their hidden states to reduce computation while preserving autoregressive structure.

4.3. Implications for Acceleration

This implies token reduction must be task-aware:

1. **Understanding:** Visual tokens are redundant after the first few layers. It is possible to reduce image tokens in some way and significantly reduce quadratic attention cost with minimal performance impact.
2. **Generation:** Visual tokens are autoregressively generated during inference, and removing training computation on them must still enable the same next-token prediction for inference. Deep layers in visual generation also have limited bandwidth to reduce image token-related computation.

Therefore, a unified model cannot rely on a single token-dropping rule. The structural roles of visual tokens differ fundamentally between discriminative and generative objectives. We introduce the task-specific training acceleration methods below.

5. Proposed Task-Specific Accelerators

Motivated by the task-specific redundancy revealed in Sec. 4, we investigate whether token-reduction-based acceleration can be done separately for visual understanding and generation. We first evaluate these strategies in isolation before analyzing their behavior under unified training.

5.1. Understanding (U)

Method. We adopt HiMix [46] as the baseline accelerator for visual understanding. Unlike token merging/dropping given complete image token sets based on inference-time analysis, HiMix modifies the attention computation in a manner compatible with both training and inference, making it suitable for unified autoregressive VLMs. The key idea is to *reduce tokens in queries*.

Specifically, as illustrated in Figure 4, image tokens are removed from the *query* projections while retained in the *key* and *value* projections. This eliminates quadratic image-to-image attention while preserving text-to-image interactions. As shown in Sec. 4, visual tokens become increasingly redundant in deeper layers for understanding tasks; thus, removing them from queries reduces computation with minimal impact on prediction. Moreover, noticing that the final output of each layer only includes text tokens as image tokens are removed from queries, this strategy requires the input original image tokens to each layer of the transformer.

Theoretical Efficiency. For a sequence with T text tokens and M image tokens (total length $T + M$) and hidden size d, the per-layer complexity of a vanilla Transformer can be decomposed into: (i) self-attention, dominated by the $QK^\top$ operation, $\mathcal{O}((T + M)^2 d)$; and (ii) the feed-forward network (two linear layers), $\mathcal{O}(8(T + M)d^2)$. Thus,

$$\text{Cost}_{\text{base}} = \mathcal{O}\big((T + M)^2 d + 8(T + M)d^2\big).$$

With HiMix, image tokens are removed from the *query* stream, so attention is computed only for T text queries over $(T + M)$ keys/values, reducing the attention term to $\mathcal{O}(T(T + M)d)$ while keeping the FFN term unchanged:

$$\text{Cost}_{\text{HiMix}} = \mathcal{O}\big(T(T + M)d + 8Td^2\big).$$

When $M \gg T$, the dominant $\mathcal{O}(M^2 d)$ attention term is removed. In practice, this leads to substantial FLOPs reduction while preserving the cross-modal interactions necessary for visual understanding.

Experimental Results. We evaluate HiMix in an understanding-only setting of VILA-U (LLaMA-3-3B backbone [5, 34]), with the ShareGPT-4v dataset [2]. We follow VILA-U to conduct pretraining and finetuning each

Table 1. **HiMix for visual understanding.** Performance and computational cost comparison between the understanding-only VILA-U baseline and HiMix. HiMix reduces training FLOPs to 0.24× (76% reduction) by removing image-token queries, while incurring only moderate performance degradation across GQA, MME, POPE, and SeedBench benchmarks. The relatively small accuracy drop compared to the substantial computational savings confirms significant late-layer visual redundancy in understanding tasks.

Method	GQA	MME-C	MME-P	POPE-A	POPE-P	POPE-R	POPE-F1	SeedBench-Img	FLOPs
VILA-U (U-only)	52.86	258.21	1054.88	81.30	84.67	74.76	79.40	46.05	1×
HiMix (U-only)	49.92	224.64	983.30	78.56	78.03	79.49	78.75	40.88	0.24×

for one epoch, and evaluate on several visual understanding benchmarks [8, 13, 15, 18]. Results are in Table 1.

HiMix reduces FLOPs to $0.24\times$ of the baseline, corresponding to a 76% reduction in computation. Despite this substantial saving, performance degradation remains moderate. For example, GQA accuracy decreases from 52.86 to 49.92, while POPE F1 drops slightly from 79.40 to 78.75. Notably, the performance drop is significantly smaller than the reduction in computational cost, indicating substantial redundancy in late-layer visual processing for understanding tasks. Overall, these results confirm that visual understanding exhibits considerable late-layer image redundancy. Structured removal of image-token queries yields large efficiency gains while largely preserving cross-modal reasoning capability.

5.2. Generation (G)

Design Constraints from Autoregressive Image Generation. Unlike visual understanding, visual generation follows a strict autoregressive process: each predicted image token is appended to the sequence and must serve as a valid query for predicting subsequent tokens. Therefore, image tokens *must remain in the query stream*. Removing them from queries would break the autoregressive dependency chain and make inference inconsistent with training. This constraint fundamentally differentiates generation from understanding and prevents directly applying HiMix-style query removal.

One might instead consider removing image tokens from key/value projections while keeping them in queries. Although this reduces part of the attention computation, two major issues arise. **(1) Limited FLOPs Reduction.** Even if image-to-image attention is partially suppressed, the feed-forward network (FFN) still processes all image tokens. When $M \gg T$, the dominant $\mathcal{O}(8Md^2)$ FFN term remains intact, resulting in minimal overall computational savings. **(2) Severe Performance Degradation.** Image generation exhibits persistent image-token dependence across depth (Sec. 4). Suppressing key/value participation disrupts hierarchical autoregressive conditioning, leading to substantial quality degradation in practice. Empirically, we observe that this naive modification yields both limited efficiency gains and large drops in generative performance. For example, applying this design to one middle layer leads to a significant drop (-3.52) on MJHQ-30K [16].

Figure 5. **Inference-time-only HMGen.** Qualitative comparison between the original model (bottom) and inference-time-only HMGen (top), where image-token computation in shallow layers is skipped without retraining. Visual quality and semantic consistency are largely preserved despite reduced computation.

HMGen: Hierarchical Mixture for Generation. Motivated by the hierarchical attention structure observed in Sec. 4, we instead propose **HMGen**, which is composed of two kinds of layers illustrated in Figure 4. HMGen preserves the autoregressive structure with image in query *from model level* while reducing tokens in specific layers.

We introduce K designated *shallow layers* in the middle portion of the transformer (out of L total layers) while other layers remain as *full layers*. This is because early full layers are required to preserve global conditioning, while late full layers are required to ensure high-fidelity final token prediction. We empirically find that *alternating shallow and full layers in the middle layers* yields the best trade-off between efficiency and generation quality.

In *shallow layers*, image-token attention computation is skipped, and the feed-forward network is applied only to text tokens. The image-token hidden states are directly forwarded from the previous layer to the next without participating in self-attention or FFN updates.

In *full layers*, we further introduce separate projection parameters for image and text tokens. Although the backbone remains unified, decoupling image-related projections stabilizes training and improves generation quality. This separation allows image-token representations to maintain dedicated pathways even when their participation in attention is selectively reduced. Empirically, we observe improved performance compared to fully shared parameterization under the same FLOPs budget.

Theoretical Efficiency. HMGen maintains the autoregressive dependency chain while reducing computation in K designated middle "shallow" layers (out of L total) by skipping image-token attention/MLP computation and forwarding their hidden states. Using the same decomposition

as above, a vanilla layer costs

$$\text{Cost}_{\text{base}} = \mathcal{O}\big((T+M)^2 d + 8(T+M)d^2\big).$$

In a shallow layer, attention is computed only for T text queries, giving $\mathcal{O}(T^2 d)$, and the FFN is applied only to text tokens, giving $\mathcal{O}(8Td^2)$:

$$\text{Cost}_{\text{shallow}} = \mathcal{O}\big(T^2 d + 8Td^2\big).$$

The total complexity across L layers is therefore

$$\text{Cost}_{\text{HMGen}} = (L-K)\,\mathcal{O}\big((T+M)^2 d + 8(T+M)d^2\big) + K\,\mathcal{O}\big(T^2 d + 8Td^2\big).$$

When $M \gg T$, each shallow layer removes the dominant image-related costs in both attention and FFN, i.e., the $\mathcal{O}(M^2 d)$ and $\mathcal{O}(8Md^2)$ terms. Consequently, the overall FLOPs reduction scales with the fraction of layers made shallow (K/L), and is upper-bounded by the compute in the remaining $(L-K)$ full layers. In the idealized regime where shallow layers contribute negligible cost relative to full layers, the relative cost approaches $1 - K/L$, yielding an approximate speedup of $1/(1-K/L)$.

Experimental Results. We first evaluate HMGen in a generation-only setting of VILA-U using the JourneyDB [31] dataset, and evaluate visual generation on MJHQ-30K [16]. Quantitative results are shown in Table 2, and qualitative inference-time only results (without training, just directly skipping image-related computations in middle layers) are visualized in Figure 5.

(1) Inference-Time Applicability. Figure 5 demonstrates that HMGen can be directly applied at inference time without architectural modification. By design, shallow layers preserve the autoregressive query structure, allowing image tokens to be appended and used as subsequent queries during generation. This confirms that HMGen is not merely a training-time approximation but a structurally consistent acceleration mechanism.

(2) Reasonable FLOPs Reduction. As shown in Table 2, introducing K shallow layers yields significant computational savings. With $K = 3$, FLOPs are reduced to $0.85\times$ of the baseline, and with $K = 5$, to $0.75\times$. Since each shallow layer removes both the dominant $\mathcal{O}(M^2 d)$ attention term and the $\mathcal{O}(8Md^2)$ FFN term, the efficiency gain scales approximately with the fraction of shallow layers.

(3) Improved Generation Quality. Notably, HMGen achieves substantially better MJHQ-30K scores compared to the generation-only VILA-U baseline (17.45 $\rightarrow$ 12.16 with $K = 3$). This improvement arises from our separation of image and text projection parameters within the full layers. By decoupling image-specific transformations, the model maintains more stable hierarchical image representations even when computation is selectively reduced.

Table 2. **HMGen for visual generation.** Introducing shallow layers reduces FLOPs (to 0.85× and 0.75×) while improving generative quality compared to the VILA-U baseline, demonstrating efficient and structure-aware acceleration.

Method	#Shallow Layers	MJHQ-30K	FLOPs
VILA-U (G-only)	0	17.45	1×
HMGen	3	**12.16**	0.85x
HMGen	5	**12.55**	0.75x

Overall, HMGen not only reduces computation but also enhances generative quality, demonstrating that hierarchical, structure-aware acceleration is better aligned with the intrinsic dependencies of visual generation.

6. The Limits of Unified Efficiency

While the task-specific token-reduction-based accelerators in Sec. 4 demonstrate substantial efficiency gains when applied to understanding or generation in isolation, our primary objective is to evaluate their behavior under unified training. In unified VLMs, both objectives are optimized jointly under a shared backbone, and improvements in one task often influence the other through shared representations. Efficiency modifications may therefore interact with cross-task learning dynamics in subtle ways. In this section, we examine whether task-specific acceleration strategies remain effective in a unified setting, and identify structural barriers that emerge during joint optimization.

6.1. Synergy Breakage: The Cost of Efficiency

Positive Cross-Task Synergy Baseline We first examine the unified baseline without token reduction. From Table 3, joint training improves both tasks relative to their single-task counterparts. **Understanding improves under unified training:** GQA increases from 52.86 (U-only) to 56 (Unified), POPE F1 from 79.40 to 82.3, and SeedBench from 46.05 to 47.88. **Generation also improves:** MJHQ-30K improves from 17.45 (G-only) to 15.78 (Unified), indicating better generative quality. Formally, let performance on understanding and generation be $\mathcal{U}(\theta)$ and $\mathcal{G}(\theta)$. For the unified baseline,

$$\mathcal{U}(\theta_{\text{unified}}) > \mathcal{U}(\theta_{\text{U-only}}), \quad \mathcal{G}(\theta_{\text{unified}}) > \mathcal{G}(\theta_{\text{G-only}}).$$

This mutual improvement confirms the presence of positive cross-task transfer, which motivates unified modeling.

Severe Collapse with Fully Shared Parameters in HiMix-HMGen. The row *HiMix–HMGen (Share All)* in Table 3 shows substantial degradation than HiMix (U-only) and HMGen (g-only): GQA drops from 56 to 33, POPE F1 from 82.3 to 67.59, SeedBench from 47.88 to 31.38, while MJHQ-30K worsens from 12.16 to 12.53. Although FLOPs are reduced to $0.56\times$, both tasks suffer significant performance collapse. Notably, unified performance becomes worse than the single-task baseline in some metrics,

Table 3. **Unified training performance and efficiency.** The unified baseline improves both understanding (e.g., GQA 0.5600 vs. 0.5286 U-only) and generation (MJHQ 15.78 vs. 17.45 G-only), demonstrating positive cross-task synergy. However, combining HiMix and HMGen under joint training substantially reduces FLOPs (0.55–0.56×) but degrades performance on both objectives (e.g., GQA drops to 0.4705/0.3300 and MJHQ worsens to 14.54/12.53), indicating that task-specific token reduction disrupts mutual gains.

Method	GQA	MME-C	MME-P	POPE-A	POPE-P	POPE-R	POPE-F1	SeedBench-Img	MJHQ	FLOPs
VILA-U (U-only)	52.86	258.21	1054.88	81.30	84.67	74.76	79.40	46.05	–	1×
VILA-U (G-only)	–	–	–	–	–	–	–	–	17.45	1×
VILA-U (Unified)	56.00	250.00	1135.91	83.37	87.95	77.33	82.30	47.88	15.78	1×
HiMix (U-only)	49.92	224.64	983.30	78.56	78.03	79.49	78.75	40.88	–	0.24×
HMGen (G-only)	–	–	–	–	–	–	–	–	12.16	0.85×
HiMix-HMGen (*Share All Params*)	33.00	233.21	662.26	60.84	57.66	81.64	67.59	31.38	12.53	0.56×
HiMix-HMGen (*Share Partial Params*)	47.05	255.00	847.82	76.43	76.10	77.10	76.58	34.50	14.54	0.55×

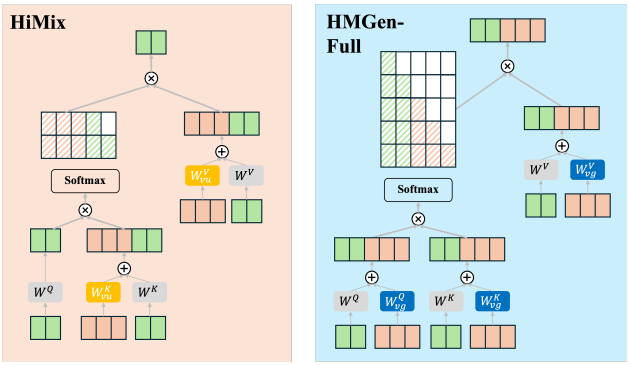

Figure 6. **Separate image projection strategy.** To reduce interference between task-specific routing, we decouple image-related projection parameters (e.g., W_v^Q, W_v^K, W_v^V) for HiMix (highlighted by yellow) and HMGen-Full (highlighted by blue), while keeping the backbone shared. This design aims to stabilize hierarchical image representations when token participation differs across tasks.

indicating negative transfer. Thus, naively combining task-specific accelerators destroys cross-task synergy.

6.2. Separate Image Projection Strategy

To mitigate this issue, we introduce partial decoupling of image-related projections, as illustrated in Figure 6. Instead of fully shared projections, we decompose:

$$W_v^Q = \{W_{vu}^Q, W_{vx}^Q\},$$

and similarly for W_v^K and W_v^V. This creates semi-independent image pathways while preserving the unified backbone. From Table 3, *HiMix–HMGen (Share Partial)* improves over the fully shared variant significantly: GQA increases from 33 to 47.05, POPE F1 from 67.59 to 76.58, and MJHQ-30K from 12.53 to 14.54. However, performance still falls short of the unified baseline per understanding, while better than the unified baseline on generation (56 GQA and 15.78 MJHQ-30K), indicating that parameter separation partially restores synergy.

6.3. Structural Drivers of Synergy Loss

We discuss possible drivers of the observed synergy breakage below to guide future investigation. Unified training implicitly assumes a shared latent space $\phi(z;\theta)$, where discriminative and generative signals co-shape representations. Task-specific token dropping changes which tokens participate in attention and which parameters receive gradients. Consequently, gradients $\nabla_\theta \mathcal{L}_U$ and $\nabla_\theta \mathcal{L}_G$ are computed under incompatible masking operators, leading to potentially fragmented optimization dynamics. This hypothesis is also supported by the separate image projection strategy.

Key Takeaway. Table 3 reveals a consistent pattern: the unified baseline exhibits positive cross-task transfer, whereas task-specific token reduction eliminates or reverses these gains. Efficiency improvements achieved in isolation do not compose under unified optimization. Effective unified acceleration must therefore preserve shared computational pathways that enable cross-task representation alignment, rather than simply aggregating task-optimal pruning strategies.

7. Conclusion

We investigate the feasibility and limits of token-reduction-based acceleration for unified vision-language models and identify a fundamental asymmetry in visual token usage: visual understanding exhibits substantial late-layer redundancy, whereas visual generation maintains persistent image-token dependence across depth. Based on this insight, we design task-specific accelerators that achieve significant efficiency gains in isolated settings; however, when combined under unified training, they induce a consistent *synergy loss*, as task-specific token dropping leads to divergent parameter usage and removes the mutual performance gains typically observed in joint optimization. Our findings suggest that efficient unified modeling requires preserving shared computational pathways that enable cross-task representation alignment, rather than simply aggregating task-specific strategies.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster, 2023. 1, 2

[2] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024. 5

[3] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024. 1, 2, 3

[4] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. 4, 5

[6] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. 1

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 1

[8] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. 6

[9] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer, 2023. 1

[10] Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024. 2

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1

[12] Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models, 2024. 1, 2

[13] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. 6

[14] Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, Di Zhang, Wenwu Ou, Kun Gai, and Yadong Mu. Unified language-vision pretraining in llm with dynamic discrete visual tokenization, 2024. 1

[15] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023. 6

[16] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024. 6, 7

[17] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm, 2024. 2

[18] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. 6

[19] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models, 2023. 2

[20] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference, 2025. 2

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1

[23] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

[24] Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models, 2023. 1, 2

[25] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding, 2025. 1

[26] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation, 2024. 1

[27] 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, 2021. 1

[28] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2

[30] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient

large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 1, 2

[31] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in neural information processing systems*, 36:49659–49678, 2023. 7

[32] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. 1, 2

[33] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. 1

[34] 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, 2023. 5

[35] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 2

[36] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024. 1, 2

[37] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024. 1, 2

[38] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024. 1

[39] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024. 1, 2

[40] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv*, 2023. 2

[41] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 1, 2

[42] 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, 2023.

[43] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, Hang Yan, Jie Fu, Tao Gui, Tianxiang Sun, Yu-Gang Jiang, and Xipeng Qiu. Anygpt: Unified multimodal llm with discrete sequence modeling, 2025. 1

[44] Junyang Zhang, Mu Yuan, Ruiguang Zhong, Puhan Luo, Huiyou Zhan, Ningkang Zhang, Chengchen Hu, and Xiangyang Li. A-vl: Adaptive attention for large vision-language models, 2025. 1, 2

[45] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token, 2025. 2

[46] Xuange Zhang, Dengjie Li, Bo Liu, Zenghao Bao, Yao Zhou, Baisong Yang, Zhongying Liu, Yujie Zhong, Zheng Zhao, and Tongtong Yuan. Himix: Reducing computational complexity in large vision-language models, 2025. 2, 5