Understanding and Harnessing Sparsity for Unified Multimodal Models

Anonymous authors
Paper under double-blind review

Abstract

Large multimodal models have achieved remarkable progress in both understanding and generation. Recent efforts pursue unified multimodal models that integrate heterogeneous components to support both capabilities within a single framework. However, such unification introduces inference inefficiencies, e.g., specific tasks or samples may not require the full knowledge or capacity of the unified model. Yet, a systematic understanding of how these inefficiencies manifest across different components remains limited. In this work, we first conduct a systematic analysis of unified multimodal model components using training-free pruning as a probing methodology, considering both depth pruning and width reduction. Our study reveals that the understanding component exhibits notable compressibility in both understanding and generation tasks, which is more pronounced in the latter. In contrast, the generation components are highly sensitive to compression, with performance deteriorating sharply even under moderate compression ratios. To address this limitation, we propose the Mixture-of-Experts (MoE) Adaptation, inspired by the dynamic activation patterns observed across different samples. This approach partitions the generation module into multiple experts and enables sparse activation to restore generation quality. We validate the effectiveness of sparse activation through expert-frozen tuning and further demonstrate that a fully trainable adaptation delivers additional gains. As a result, the adapted BAGEL model achieves performance comparable to the full model while activating only about half of its neurons. The code will be released upon acceptance.

1 Introduction

Large-scale multimodal models have recently achieved remarkable progress in both multimodal understanding (Liu et al., 2023; Li et al., 2023; Dai et al., 2023; Lu et al., 2024; 2023) and generation (Ramesh et al., 2021; Saharia et al., 2022a; Peebles & Xie, 2023). Traditionally, these two tasks were studied in isolation, leading to distinct research trajectories and model families: *understanding models* for vision–language reasoning with textual outputs, and *generative models* designed for image synthesis. While effective for task-specific purposes, this separation stands in contrast to the broader pursuit of Artificial General Intelligence (Wei et al., 2022; Bubeck et al., 2023), which envisions a unified model capable of both understanding and generating across modalities. Recent research has increasingly explored unified multimodal models (Wu et al., 2024; Chen et al., 2025; Deng et al., 2025; Liang et al., 2025; AI et al., 2025), which integrate heterogeneous components such as vision encoders (Dosovitskiy et al., 2021), language backbones (Dubey et al., 2024; Yang et al., 2024), and image decoders (Peebles & Xie, 2023; AI et al., 2025). These unified architectures can seamlessly support both understanding and generation within a single system, marking a promising step toward general-purpose multimodal intelligence.

However, this unification comes at a substantial ***cost in efficiency***, as many tasks or input samples do not require the full knowledge or capacity of the unified model, but rather rely on a much slimmer or more sparsely activated sub-architecture Frankle & Carbin (2018); Xu et al. (2021). In addition, we identify three complementary observations that shed light on the slimness and sparsity manifested in unified multimodal models: (1) ***Component-wise redundancy***: the understanding and generation components follow distinct computation patterns and serve different functional roles, leading to different levels of redundancy. (2)

Task-specific activation: task-specific inputs tend to activate only a limited subset of parameters during inference, leaving much of the shared capacity underutilized Kudugunta et al. (2021); Sarkar et al. (2023). (3) ***Input variability***: even within the same task, different input queries can activate different portions of the model, further complicating efficient compute allocation Liu et al. (2024); Luo et al. (2025).

Motivated by these factors, We systematically analyze the components of unified multimodal models and examine how they allocate and utilize parameters across tasks and inputs. Specifically, we first adopt ***training-free pruning*** as a probing methodology (Gromov et al., 2025; Men et al., 2024; He et al., 2024; Ma et al., 2023; Xia et al., 2023), as it enables us to infer structural importance without retraining by removing structures and observing the resulting performance change. We begin by analyzing the understanding components, which serve as shared modules that process inputs from multiple modalities and produce language representations or embeddings that guide the generation components. Our results demonstrate that the understanding components are highly compressible in multimodal generation tasks but less so in understanding tasks. Furthermore, we observe clear task-specific activation patterns: understanding and generation tasks predominantly activate different model partitions, underscoring the necessity of dynamic activation for different testing tasks.

In contrast to understanding components, generation components (e.g., image generators) are far less tolerant to static compression, where we find the activation patterns vary significantly across samples and timesteps. Consequently, applying static pruning leads to a drastic drop in the generated image quality due to the inability to accommodate such dynamic activation shifts. To address this issue, we propose a ***Mixture-of-Experts (MoE) Adaptation***, where MLP neurons are partitioned into experts, allowing the model to selectively activate subsets of neurons to ensure efficient dynamic activation. To align the model with sparsely routed inference, we first introduce expert-frozen tuning, which freezes all experts and updates only the remaining parameters. Building on this initialization, we further perform fully end-to-end MoE training, yielding additional improvements, where the adapted BAGEL model (Deng et al., 2025) matches the performance of the full model while activating only about half of neurons. Our contributions are threefold:

- Our study demonstrates that the understanding component in unified multimodal models has high structural redundancy and can be compressed even more aggressively for generation tasks.
- We introduce a training-free neuron partition analysis that uncovers task-specific activation patterns and the corresponding redundancy within the understanding component, offering insights into task-aware compression.
- To cope with the high compression sensitivity of the generation component, we design a tailored MoE Adaptation that preserves generation quality while activating only about half of its parameters.

2 Related Work

Multimodal Models for Understanding and Generation Early multimodal research typically treated understanding and generation as separate tasks, leading to two distinct architectural paradigms (Li et al., 2023; Dai et al., 2023; Zhu et al., 2023). On the one hand, multimodal large language models (MLLMs) extend language models to handle input tokens from multiple modalities, such as LLaVA (Liu et al., 2023), which augments the LLaMA backbone (Touvron et al., 2023a) with visual tokens for vision–language understanding. On the other hand, multimodal generative models employ dedicated generators that can synthesize high-fidelity visual outputs (Ramesh et al., 2022; Saharia et al., 2022b; Chang et al., 2023). For instance, Diffusion Transformers (DiT) (Peebles & Xie, 2023) demonstrate that iterative denoising with transformer-based backbones can effectively generate high-quality images guided by natural language. More recently, several works aim to unify understanding and generation within a single framework (Wu et al., 2024; Chen et al., 2025; Zhang et al., 2025). BAGEL (Deng et al., 2025) employs a mixture-of-transformers design (Liang et al., 2025) to separate understanding and generation modules, while Ming-Omni (AI et al., 2025) uses a mixture-of-experts backbone with multimodal understanding and modality-specific decoders for generation. Although these unified models demonstrate strong versatility, their increased architectural complexity poses new challenges for efficiency, which remains underexplored.

Model Compression toward Parameter Efficiency Despite the remarkable advances of scaling large language models, the continual growth in their size has introduced substantial redundancy and raised critical challenges for scalability. Network pruning (Liu et al., 2019; Cheng et al., 2024) has emerged as an effective technique to identify and remove redundant structures in neural networks, thereby improving parameter efficiency and reducing inference cost. For instance, (Gromov et al., 2025) demonstrated that many deep layers in large language models are relatively unimportant, and that comparable performance can still be maintained after removing these redundant layers. Within single layers, *width compression* provides a complementary approach to reduce parameter count by shrinking the intermediate or hidden dimensions of model components (Ma et al., 2023; Xia et al., 2023). While unimodal compression techniques can be transferred to vision-language models that take multimodal inputs and produce language responses (Sung et al., 2024; Lin et al., 2024; He et al., 2025b), it is unclear whether such methods still work in unified models. Building on prior efforts, we systematically explore and exploit redundancy in multimodal models, where heterogeneous components play distinct roles. This perspective enables us to design compression strategies better aligned with the unified nature of multimodal understanding and generation.

3 Preliminaries: Unified Multimodal Models

Unified multimodal models are model architectures that integrate both multimodal understanding and generation within a single framework.

Understanding Given a unified multimodal model, let $\mathbf{x}$ denote the multimodal input, and $\mathbf{y}$ denote the corresponding output. For understanding tasks, the model predicts textual outputs in an auto-regressive manner:

$$p(\mathbf{y}_{\text{und}} \mid \mathbf{x}; \theta_{\text{und}}) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, \mathbf{x}; \theta_{\text{und}}) \,, \tag{1}$$

where θ_{und} denotes the parameters of the understanding component, and $\mathbf{y}_{\text{und}}$ denotes the sequence of output tokens, i.e., $(y_1, y_2, \ldots, y_T)$.

Generation For generation tasks, the unified model leverages the understanding component to process an instructional input $\mathbf{x}_{\text{inst}}$ (e.g., text prompt and reference images), producing conditional features $f_{\text{und}}(\mathbf{x}_{\text{inst}}; \theta_{\text{und}})$. The generative component, parameterized by θ_{gen}, synthesizes the output $\mathbf{y}_{\text{gen}}$, conditioned on the representation $f_{\text{und}}(\mathbf{x}_{\text{inst}}; \theta_{\text{und}})$ from the understanding component, together with an additional generative input $\mathbf{z}$ (e.g., random noise in diffusion models):

$$\mathbf{y}_{\text{gen}} \sim p(\mathbf{y}_{\text{gen}} \mid f_{\text{und}}(\mathbf{x}_{\text{inst}}; \theta_{\text{und}}), \mathbf{z}; \theta_{\text{gen}}) \,, \tag{2}$$

In unified multimodal models, the understanding component processes inputs from all modalities, while the generation component explicitly operates on non-text outputs such as image or audio synthesis. Due to their heterogeneous objectives and computational pathways, the understanding and generation components could exhibit different characteristics under compression.

4 Methodology

In this section, we first introduce training-free pruning techniques to probe architectural redundancy, and then propose a training-aware MoE adaptation that leverages dynamic sparsity to maintain performance under sparse activation. The proposed methods are illustrated in Figure 1.

4.1 Training-free Compression Strategies

Depth Pruning via Layer Dropping Transformer-based large language models consist of multiple layers, and scaling their depth effectively improves performance. Nevertheless, excessive depth also introduces structural redundancy. Following (Gromov et al., 2025; Men et al., 2024; He et al., 2024), we measure the redundancy of a single layer via:

$$S_l = \text{CosineSim}(\mathbf{x}_l, \mathbf{y}_l), \tag{3}$$

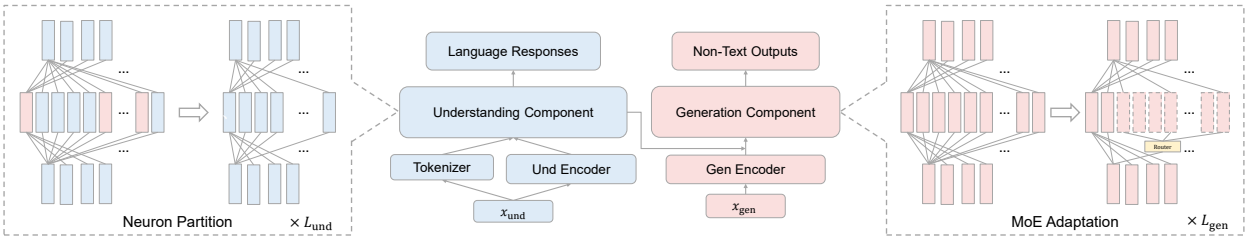


Figure 1: **Overview of the proposed slimness- and sparsity-oriented techniques for efficient unified multimodal models.** We introduce two complementary strategies: **(1) Training-free compression**, exemplified by neuron partitioning, which groups hidden neurons into subsets and prunes those less important for the target task; and **(2) Training-aware MoE Adaptation**, which dynamically activates neurons through a Mixture-of-Experts design, where neurons are organized into shared experts (solid lines) and routed experts (dotted lines) controlled by a router.

where $\mathbf{x}_l$ and $\mathbf{y}_l$ correspond to the input and output of the l-th layer, respectively. The similarity provides a measure of redundancy, with higher values implying that the layer contributes only marginal transformation. The metric has been shown to perform effectively in unimodal LLMs such as Mistral (Jiang et al., 2023) and LLaMA (Touvron et al., 2023b; Dubey et al., 2024), and we next extend this metric to unified models.

Width Reduction via Neuron Partition In addition to depth, scaling the width, particularly within MLP layers, has become a prevalent strategy for enhancing model capability. In Transformer MLP layers, the input is expanded from dimension d to dm, activating dm hidden neurons before being projected back to d. While this expansion increases model capacity, it also introduces redundancy among neurons. To mitigate this redundancy, we propose neuron partitioning that separates the hidden neurons into important and less important subsets, preserving the former while pruning the latter to achieve width reduction without significantly affecting performance.

To measure neuron importance, we take inspiration from Wanda (Sun et al., 2024), which combines weight magnitudes and activation statistics as pruning criteria, and extend this idea from unstructured weight-level pruning to a structured neuron-level metric. Given an input $x \in \mathbb{R}^{s\times d}$, in a Gate-Up-Down MLP, the hidden activations $h \in \mathbb{R}^{s\times dm}$ and output $y \in \mathbb{R}^{s\times d}$ can be written as:

$$h = \left(\mathrm{SiLU}(xW_g^\top)\right) \odot (xW_u^\top), \qquad y = hW_d^\top, \tag{4}$$

where $W_g, W_u \in \mathbb{R}^{dm\times d}$ are the gate-projection matrix and up-projection matrix, $h \in \mathbb{R}^{s\times dm}$ is the gated activation, $W_d \in \mathbb{R}^{d\times dm}$ is the down-projection matrix. The hidden activations consist of dm neurons, and the contribution of the i-th neuron to the final output is:

$$\Delta y_i = h_i W_{d,i}^\top, \tag{5}$$

with $W_{d,i}$ being the i-th column vector of W_d. If the i-th neuron is pruned, the induced output error norm can be approximated by:

$$\|\Delta y\|_2 \approx \|h_i W_{d,i}^\top\|_2. \tag{6}$$

Given all inputs from the calibration dataset $\mathcal{D}$, the expected accumulated error of each neuron is used as its importance metric:

$$s_i = \mathbb{E}_{x\sim\mathcal{D}}\left[|h_i| \cdot \|W_{d,i}^\top\|_2\right], \tag{7}$$

where $|h_i|$ measures the average activation magnitude of the i-th neuron, and $\|W_{d,i}\|_2$ quantifies its amplification effect on the output. Therefore, neurons with larger scores play more critical roles, while those with smaller scores tend to be removed. Unlike unstructured pruning that zeroes individual weights, our approach enforces structured pruning by removing entire neurons. Concretely, this corresponds to removing column i from W_d and row i from both W_u and W_g, thereby ensuring hardware-friendly efficiency.

These models integrate diverse tasks within a single architecture, and different tasks naturally activate different subsets of neurons. Figure 2 illustrates the distinct neuron partitions within the understanding component. We first identify the top 50% of neurons according to their importance scores for understanding and for generation, respectively. Based on these two sets, we then partition neurons into three disjoint groups: those that lie only in the understanding top-50% set, those that lie only in the generation top-50% set, and those that are shared by both. The relatively low overlap ratio suggests that different tasks activate distinct neuron subsets within the understanding component. Therefore, we propose to align the calibration samples with the target tasks before applying the neuron-level importance metric, ensuring a more accurate identification of principal neurons.

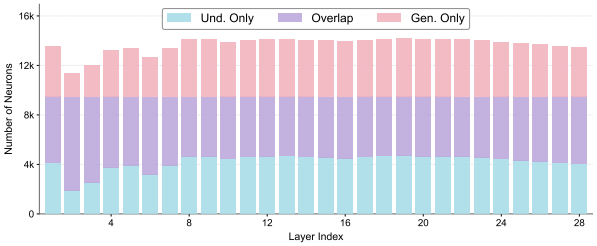


Figure 2: Statistical analysis of high-importance neurons across understanding tasks, generation tasks, and their shared subset.

4.2 Training-Aware MoE Adaptation

Figure 3 visualizes the active neurons (those consistently ranked within the top 50% by activation scores) and inactive neurons (those never entering the top 50%) across layers of the generation component over multiple prompts and time steps. We observe that only a small subset of neurons remain consistently active throughout inference, while most exhibit sample-dependent activation patterns, indicating dynamic specialization across inputs. This observation reveals a dynamic activation phenomenon, where the subset of activated parameters depends on the input, aligning with the intuition behind the Mixture-of-Experts (MoE) architecture. To leverage this property, we incorporate an MoE mechanism into unified models through two key stages: *Expert Partition* and *MoE Adaptation*.

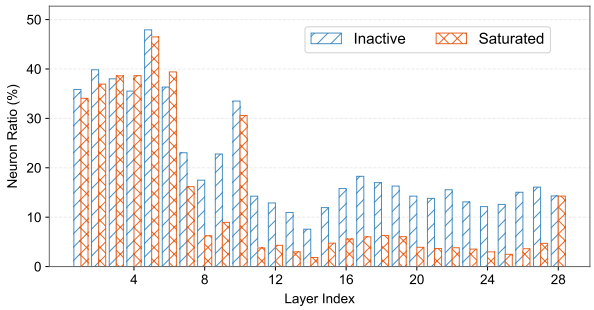


Figure 3: **Visualization of the proportion of inactive and saturated neurons.** Active neurons are defined as those consistently ranked within the top 50% by activation scores, while inactive neurons never enter the top 50%.

Expert Partition To separate universal and sample-specific capacity (He et al., 2023; Dai et al., 2024), we partition MLP neurons into shared and routed experts using cumulative importance across calibration samples. For each neuron, we compute its cumulative importance score using Equation 7. The neurons with scores higher than the threshold are selected as shared experts E_s, preserving features that consistently benefit multiple samples. The remaining neurons, whose relative importance is more sample-dependent, are evenly allocated to routed experts $E_r^{(1)}, \ldots, E_r^{(n)}$ by ranked importance. Specifically, neurons are sorted in descending order of importance and sequentially assigned to experts in alternating forward and reverse order (i.e., $E_r^{(1)}$ to $E_r^{(n)}$, then $E_r^{(n)}$ back to $E_r^{(1)}$), ensuring balanced total importance across experts.

MoE Adaptation After expert partition, we insert a router per layer to dynamically select routed experts for each input. In this case, the output of an MoE layer is formulated as follows:

$$\mathrm{MoE}(x) = f_{\mathcal{S}}(x) + \sum_{j \in \text{Top-}k(\mathcal{G})} \mathcal{G}_j \cdot f_{\mathcal{R}_j}(x), \tag{8}$$

where $\mathcal{G}$ denotes the gating function, and $f_{\mathcal{S}}$ and $f_{\mathcal{R}}$ represent the transformations of the shared and routed experts, respectively. The original MLP layer can be regarded as a special case of Equation 8, where all experts are selected and their gating scores are uniformly set to 1. To enable a smooth transition from dense to sparse activation, we relax the constraint that the gating scores sum to 1 and reparameterize them as $(1 + \mathrm{Router}(x))$, where the router network's gate is initialized to zero. To adapt the model to the converted

sparse activation pattern, we perform Expert-Frozen Tuning as a cold-start phase of MoE Adaptation, where experts remain fixed while the router and other parameters are trainable. This stage allows the model to leverage the pretrained knowledge of existing experts while establishing a preliminary routing policy. Afterward, we release the freezing constraint to fully enable MoE Adaptation.

5 Experiments

5.1 Experimental Setup

Models We focus on several mainstream open-source unified models, including BAGEL (Deng et al., 2025), Ming-Omni (AI et al., 2025), and Qwen-Image (Wu et al., 2025). For the understanding component, BAGEL and Qwen-Image rely on a VLM derived from Qwen-Instruct (Yang et al., 2024), whereas Ming-Omni employs an MoE-based backbone (Team, 2025). The generation components further diverge across models: BAGEL employs a Mixture-of-Transformers (MoT) (Liang et al., 2025) design and reuses the Qwen-Instruct backbone for generation; Qwen-Image incorporates an MMDiT-based generator (Esser et al., 2024) and Ming-Omni adopts a multi-scale DiT block architecture. Table 1 presents a detailed comparison of these models.

Table 1: Summary of evaluated unified models.

Model	Und. Component	Und. Param.	Gen. Component	Gen. Param.
Qwen-Image	VLM	7.62B	MMDiT	20.42B
Ming-Omni	MoE	17.12B	MMDiT	2.51B
BAGEL	VLM	7.62B	LLM	7.62B

We systematically vary the number of experts per MoE layer (16, 32, and 64). Following the design choices in (Dai et al., 2024; DeepSeek-AI et al., 2025), we include shared experts that constitute one-sixteenth of the total number of experts. The overall activation ratio is set to 50% per layer. We exclude the first and last layers from MoE conversion, as they are essential for preserving input encoding and output generation quality (Dai et al., 2024).

Datasets For the calibration datasets used in training-free compression, we use a small number of examples drawn directly from the target task, which is sufficient for training-free compression (Frantar et al., 2022; Lin et al., 2023; Men et al., 2024; He et al., 2024). Note that these samples are used solely to compute activation scores, so no ground-truth annotations are required and no label leakage occurs. For MoE adaptation, we additionally incorporate high-quality image–text pairs, complemented by a small amount of synthetic data generated by existing text-to-image models.

5.2 Redundancy of Understanding Components

Depth Reduction Works in Generation Tasks but Fails in Understanding Since the understanding component is not directly connected to the generation output, we first analyze its potential impacts on generation performance via depth reduction. Specifically, we remove transformer blocks, MLP layers, and attention layers, respectively. As shown in Figure 4, removing 50% of layers in the understanding component proves effective for BAGEL and Qwen-Image, but is less effective for Ming-Omni. We attribute this difference to architectural design: Ming-Omni's generation component is relatively smaller and thus depends more heavily on precise features encoded by the understanding component.

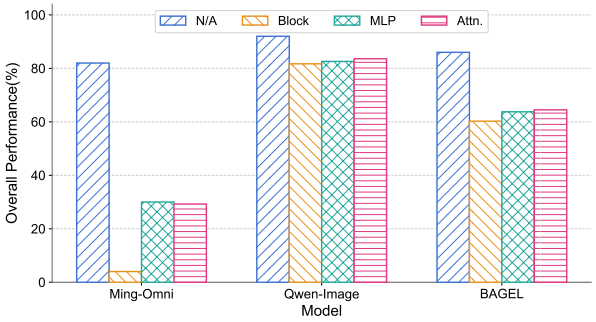


Figure 4: Comparison of the overall performance of depth reduction on GenEval.

However, depth pruning substantially deteriorates the model's understanding capability. For instance, removing half of the MLP layers causes performance on MME (Fu et al., 2023) to drop from 1684.8 to

Table 2: **Performance on GenEval when applying Neuron Partition to the understanding component**. Since only the **understanding component** is compressed, the reported parameter counts correspond to this part rather than the full model size.

Model	Sparsity	Params.	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
BAGEL	0%	7.62B	0.99	0.94	0.81	0.95	0.72	0.77	**0.86**
	50%	4.76B	0.94	0.63	0.62	0.77	0.47	0.34	**0.63**
Qwen-Image	0%	7.62B	0.99	0.98	0.91	0.94	0.80	0.89	**0.92**
	50%	4.76B	0.99	0.94	0.94	0.93	0.76	0.87	**0.90**
	70%	3.62B	0.97	0.88	0.85	0.91	0.60	0.71	**0.82**
Ming-Omni	0%	17.12B	0.97	0.95	0.67	0.92	0.71	0.71	**0.82**
	50%	8.55B	0.97	0.92	0.66	0.89	0.61	0.70	**0.79**
	70%	5.61B	0.96	0.81	0.58	0.86	0.49	0.56	**0.71**

304.5 in perception and from 696.7 to 127.1 in cognition. These results suggest that depth reduction fails to preserve the performance of unified multimodal models on understanding tasks. Understanding tasks rely on autoregressive decoding, which is inherently an error-accumulation process: deviations in early generated tokens propagate through subsequent steps and ultimately cause the output to collapse.

Neuron Partition on Understanding Components: Effective in Both Understanding and Generation We next evaluate the effectiveness of neuron partition on understanding components. Specifically, we compress the MLP layers to the target ratios using a small set of calibration samples. As shown in Table 2, Ming-Omni and Qwen-Image largely maintain their performance even under aggressive compression ratios (i.e., 50% and 70%), whereas BAGEL exhibits a greater loss in capability, likely due to its mixture-of-transformers architecture (Liang et al., 2025), in which components interact more frequently through cross-attention at every layer. Similarly, the neuron partition strategy can be extended to attention heads, and this extension remains effective when compressing the understanding component for generation tasks (see Appendix B).

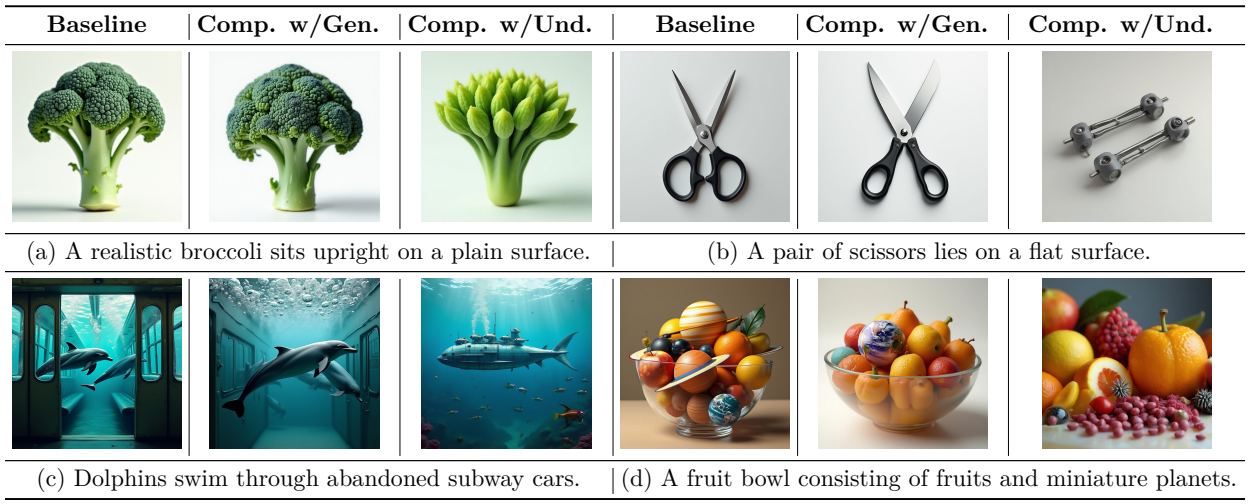
Table 3: Performance of neuron partitioning on understanding tasks at compression ratios of 25% and 50%.

Model	Sparsity	MME-P	MME-C	MMMU	MMBench	MMVP
Ming-Omni	–	1584.3	670.4	66.7	86.7	54.6
	25%	1578.5	560.4	56.7	81.2	51.3
	50%	1269.0	317.9	51.7	81.0	46.0
BAGEL	–	1684.8	696.7	65.0	88.1	69.6
	25%	1558.1	681.7	60.1	85.7	68.7
	50%	1392.6	528.9	56.7	79.2	56.0

Notably, as shown in Table 3, neuron partition consistently outperforms layer dropping across understanding tasks. Although certain layers exhibit substantial redundancy, aggressively removing them would also eliminate the small subset of weights that are critical to task performance (Yu et al., 2025). In contrast, neuron partition selectively preserves important neurons within each layer that are most relevant to the target task, thereby achieving more fine-grained and reliable compression. Nevertheless, since the understanding component directly governs textual outputs, its compression ratio should remain more conservative in understanding tasks than in generation tasks.

Calibration Data Affects the Activated Parameters Neuron partition leverages calibration samples to estimate neuron importance and prunes those that are less critical based on their activation behavior. Figure 2 shows that different neurons are activated for different tasks, suggesting that the choice of calibration data can lead to varying retained weights and thereby affect downstream performance. To assess how different calibration datasets affect parameter retention and downstream performance, we conduct an ablation study using samples from understanding tasks (MME) and generation tasks (GenEval).

We find that aligning calibration data with target tasks consistently improves performance. For instance, using generation samples for calibration degrades MMBench from 79.2 to 74.8. The generation results in Figure 5 further highlight this trend. When calibrated with image generation samples, the outputs remain faithful to the prompts, producing broccoli, scissors, dolphins, and fruit bowls with correct structures. In contrast, calibration with understanding samples introduces distortions and mismatches. This demonstrates



(a) A realistic broccoli sits upright on a plain surface. | (b) A pair of scissors lies on a flat surface.

(c) Dolphins swim through abandoned subway cars. | (d) A fruit bowl consisting of fruits and miniature planets.

Figure 5: **Impact of calibration data selection on neuron partition** within the understanding component for generation tasks. Each triplet presents outputs from the unmodified model (left), the model after neuron partition using image-generation calibration (middle), and using understanding calibration (right).

that task-aligned calibration data yields better performance, while mismatched data degrades generation quality. The effect is particularly critical for unified models, where both input and output types vary in different combinations of modalities.

5.3 Dilemma of Generation Component Compression

We next investigate how compression influences generation quality by applying neuron partition to the generation components. While neuron partition demonstrates strong performance in compressing the understanding components, applying similar compression to the generation components presents a clear dilemma. As illustrated in Figure 6, aggressive compression, such as a 50% width reduction, substantially degrades the fidelity and coherence of generated outputs. Compressed models often produce distorted structures and unrealistic textures, deviating from the intended semantics. These findings are consistent with our observations from depth reduction and attention head reduction experiments presented in Appendix A. This highlights the contrasting compressibility between understanding and generation components: whereas understanding tasks remain robust under compression, generation components are highly sensitive, limiting the extent of feasible compression.

Baseline (uncompressed) | Compressed (50% width reduction)

Figure 6: **Qualitative comparison between the baseline (uncompressed) model and the model with 50% width reduction in the generation component.** The baseline model (left) is evaluated without compression, while the compressed model (right) reduces the generator width by 50%. Results are shown for the prompt "The word START." Compression leads to noticeable degradation in fine details and semantic consistency.

5.4 MoE Adaptation for Gen. Component Sparsity

Static compression inherently conflicts with the dynamic activation patterns required across tasks and samples, leading to notable degradation in generation components. To mitigate this issue, we apply MoE Adaptation, enabling dynamic activation to restore representational capacity and overall performance.

Warmup via Expert-Frozen Tuning MoE Adaptation begins with Expert-Frozen Tuning, which serves as a cold start phase to train the model to effectively leverage the partitioned experts. This strategy mitigates catastrophic forgetting and encourages the model to learn effective expert selection while preserving pretrained knowledge (Houlsby et al., 2019; Qiao & Mahdavi, 2024; He et al., 2025a). Specifically, we examine different numbers of experts by comparing three configurations (16, 32, and 64), as shown in Figure 7 depicting the training loss curves. With a few steps of expert-frozen tuning, we observe a substantial decline in loss values, indicating that the model effectively adapts to and exploits a subset of experts to recover performance. Meanwhile, finer-grained expert partitioning enables more flexible activation combinations, leading to substantially lower training loss.

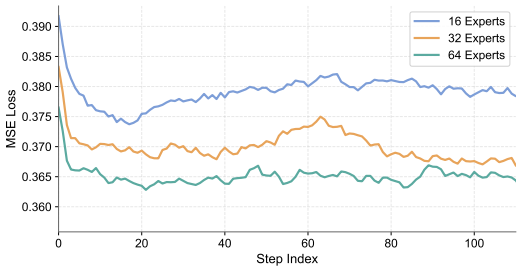


Figure 7: Training curves during the Expert-Frozen Tuning, where MoE layers are configured with different numbers of experts.

Table 4: **Performance comparison across stages of MoE Adaptation**: Expert Partition (no training), Expert-Frozen Tuning, and full MoE Adaptation. We evaluate Expert Partition (without training), Expert-Frozen Tuning, and full MoE Adaptation across two configurations of adapted components (*Gen.* and *Und. & Gen.*). For reference, results from the dense model pruned with neuron partition and subsequent fine-tuning under an equivalent budget of activated parameters are also reported. Activated Params. denote the number of activated parameters in the understanding and generation components, shown in the format "Und. Param. & Gen. Param.".

Method	Adapt. Comp.	Activated Params.	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Baseline	N/A	7.62B + 7.62B	0.99	0.94	0.81	0.95	0.72	0.77	**0.86**
Expert Partition	*Gen.*	7.42B + 4.96B	0.90	0.70	0.49	0.74	0.53	0.34	**0.62**
Dense Finetuning			0.97	0.88	0.75	0.91	0.67	0.71	**0.82**
Expert-Frozen Tuning			0.99	0.94	0.62	0.93	0.69	0.54	**0.78**
MoE Adaptation			**0.99**	**0.95**	**0.85**	**0.95**	**0.75**	**0.79**	**0.88**
Expert Partition	*Und. & Gen.*	4.96B + 4.96B	0.69	0.18	0.23	0.45	0.10	0.05	**0.28**
Dense Finetuning			0.97	0.89	0.76	0.91	0.70	0.64	**0.81**
Expert-Frozen Tuning			0.94	0.63	0.62	0.77	0.47	0.34	**0.63**
MoE Adaptation			**0.99**	**0.96**	**0.78**	**0.95**	**0.70**	**0.72**	**0.85**

For instance, the overall GenEval score improves from 0.62 to 0.78, reflecting more coherent and visually faithful outputs. As illustrated in Figure 8, before tuning, the model generates noisy, low-detail images that fail to capture fine-grained semantics. Expert-Frozen Tuning not only enhances image fidelity but also strengthens the alignment between the generated content and the given instructions. This demonstrates that certain subsets of parameters within the generation components, though difficult to compress, still retain the potential to produce high-quality images.

Effectiveness of MoE Adaptation After a few steps of Expert-Frozen Tuning as a cold start, we lift the restriction on frozen expert parameters to further improve performance. Beyond applying MoE adaptation exclusively to the generation component, we also explore extending it to the understanding component to reduce the number of activated parameters while maintaining task effectiveness. To preserve fidelity in understanding tasks, the experts in the understanding component are kept frozen, remaining fully activated for understanding and only sparsely activated for generation, since generation tasks are more tolerant to sparsity in this component. In summary, we consider two configurations of MoE adaptation: (1) *Gen.*: Expert partitioning and adaptation are applied only to generation experts; and (2) *Und. & Gen.*: Both

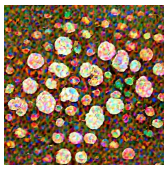
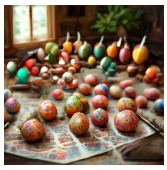
Prompt	Baseline	Zeroshot w/o SE	Zeroshot w/ SE	E.F. Tuning	MoE Adapt.
A famous flower that symbolizes wealth in China.					
Traditional activity during Easter in Western countries.					
Old analog picture of parked car on side street, quiet night.					
A lone astronaut paints swirling galaxies onto a massive canvas in a vast space station.					

Figure 8: **Visual comparison across different stages of MoE Adaptation.** Shown are outputs from the Baseline model (without modification), Zeroshot settings after expert partition with or without shared experts (Zeroshot w/o SE and Zeroshot w/ SE), the model after Expert-Frozen Tuning (E.F. Tuning), and the model further optimized via MoE Adaptation (MoE Adapt.). Test prompts are sampled from WISE (Niu et al., 2025) and the 4o-Image Generator prompt set.

understanding and generation experts are partitioned, but only the generation experts are adapted while the understanding experts remain frozen.

These additional training stages enable the experts to refine their internal representations and develop stronger specialization, improving both structural coherence and semantic consistency in generated outputs. As shown in Table 4 and Figure 8, the model achieves consistently higher generation quality after Expert-Frozen Tuning and further gains after full MoE Adaptation. This demonstrates that allowing experts to adapt beyond static partitioning significantly enhances their representational capacity and task alignment.

6 Conclusion

This work investigates the slimness and sparsity of unified multimodal models, proposing training-free pruning and dynamic activation methods to enhance efficiency. Our investigation reveals that the understanding component exhibits strong compressibility across both understanding and generation tasks. In contrast, the generation component is far more sensitive to compression, where even moderate pruning can lead to catastrophic degradation in generation quality. Inspired by the dynamic activation patterns observed across different samples, we introduce a Mixture-of-Experts (MoE) Adaptation that dynamically activates generation parameters, effectively restoring generation quality under sparse activation. By combining training-free analysis with training-aware adaptation, our study uncovers substantial optimization space for improving parameter efficiency in unified multimodal modeling.

References

Inclusion AI, Biao Gong, Cheng Zou, Chuanyang Zheng, Chunluan Zhou, Canxiang Yan, Chunxiang Jin, Chunjie Shen, Dandan Zheng, Fudong Wang, Furong Xu, GuangMing Yao, Jun Zhou, Jingdong Chen, Jianxin Sun, Jiajia Liu, Jianjiang Zhu, Jun Peng, Kaixiang Ji, Kaiyou Song, Kaimeng Ren, Libin Wang, Lixiang Ru, Lele Xie, Longhua Tan, Lyuxin Xue, Lan Wang, Mochen Bai, Ning Gao, Pei Chen, Qingpei Guo, Qinglong Zhang, Qiang Xu, Rui Liu, Ruijie Xiong, Sirui Gao, Tinghao Liu, Taisong Li, Weilong Chai, Xinyu Xiao, Xiaomei Wang, Xiaoxue Chen, Xiao Lu, Xiaoyu Li, Xingning Dong, Xuzheng Yu, Yi Yuan, Yuting Gao, Yunxiao Sun, Yipeng Chen, Yifei Wu, Yongjie Lyu, Ziping Ma, Zipeng Feng, Zhijiang Fang, Zhihao Qiu, Ziyuan Huang, and Zhengyu He. Ming-omni: A unified multimodal model for perception and generation, 2025. URL https://arxiv.org/abs/2506.09344.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL https://arxiv.org/abs/2303.12712.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, José Lezama, Lu Jiang, Ming Yang, Kevin P. Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. *ArXiv*, abs/2301.00704, 2023. URL https://api.semanticscholar.org/CorpusID:255372955.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

DeepSeek-AI et al. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Abhimanyu Dubey et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:53388625.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323, 2022. URL https://api.semanticscholar.org/CorpusID:253237200.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *ArXiv*, abs/2306.13394, 2023. URL https://api.semanticscholar.org/CorpusID:259243928.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Dan Roberts. The unreasonable ineffectiveness of the deeper layers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ngmEcEer8a.

Shwai He, Liang Ding, Daize Dong, Boan Liu, Fuqiang Yu, and Dacheng Tao. PAD-net: An efficient framework for dynamic networks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14354–14366, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.803. URL https://aclanthology.org/2023.acl-long.803.

Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed, 2024. URL https://arxiv.org/abs/2406.15786.

Shwai He, Tao Ge, Guoheng Sun, Bowei Tian, Xiaoyang Wang, and Dong Yu. Router-tuning: A simple and effective approach for enabling dynamic-depth in transformers, 2025a. URL https://arxiv.org/abs/2410.13184.

Shwai He, Ang Li, and Tianlong Chen. Rethinking pruning for vision-language models: Strategies for effective sparsity. *SIGMETRICS Perform. Eval. Rev.*, 53(2):9–14, August 2025b. ISSN 0163-5999. doi: 10.1145/3764944.3764948. URL https://doi.org/10.1145/3764944.3764948.

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 *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3577–3599, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.304. URL https://aclanthology.org/2021.findings-emnlp.304/.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Weixin Liang, LILI YU, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Nu6N69i8SB.

Haokun Lin, Haoli Bai, Zhili Liu, Lu Hou, Muyi Sun, Linqi Song, Ying Wei, and Zhenan Sun. Mope-clip: Structured pruning for efficient vision-language models with module-wise pruning error metric. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27360–27370, 2024. URL https://api.semanticscholar.org/CorpusID:268363825.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *GetMobile: Mobile Computing and Communications*, 28:12 – 17, 2023. URL https://api.semanticscholar.org/CorpusID:258999941.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=w0H2xGHlkw.

James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. Training-free activation sparsity in large language models, 2024. URL https://arxiv.org/abs/2408.14690.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *ICLR*, 2019.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HtqnVSCj3q.

Yuqi Luo, Chenyang Song, Xu Han, Yingfa Chen, Chaojun Xiao, Xiaojun Meng, Liqun Deng, Jiansheng Wei, Zhiyuan Liu, and Maosong Sun. Sparsing law: Towards large language models with greater activation sparsity. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=SBUc5wirM8.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect, 2024. URL https://arxiv.org/abs/2403.03853.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Chaoran Feng, Bin Zhu, and Li Yuan. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: parameter efficient continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZxtaNh5UYB.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. URL https://api.semanticscholar.org/CorpusID:248097655.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding.

In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022b. URL https://api.semanticscholar.org/CorpusID:248986576.

Rishov Sarkar, Hanxue Liang, Zhiwen Fan, Zhangyang Wang, and Cong Hao. Edge-moe: Memory-efficient multi-task vision transformer architecture with task-level sparsity via mixture-of-experts. *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 01–09, 2023. URL https://api.semanticscholar.org/CorpusID:258967864.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. ECoFLap: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iIT02bAKzv.

Ling Team. Every flop counts: Scaling a 300b mixture-of-experts ling llm without premium gpus. *arXiv preprint arXiv:2503.05139*, 2025.

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, 2023a. URL https://arxiv.org/abs/2302.13971.

Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

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. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

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.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9514–9528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.749. URL https://aclanthology.org/2021.emnlp-main.749/.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Mengxia Yu, De Wang, Colorado Reed, and Alvin Wan. The super weight in large language models, 2025. URL https://openreview.net/forum?id=0Ag8FQ5Rr3.

Xinjie Zhang, Jintao Guo, Shanshan Zhao, Minghao Fu, Lunhao Duan, Jiakui Hu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Additional Results on Generation Component Compression

Figure 9: **Depth reduction on the generation component causes progressive quality collapse.** Removed layers: 14 (50%), 7 (25%), 4 (14%), and 0.

Generation components exhibit substantially higher sensitivity to compression than understanding components. Beyond the neuron partition results, we further investigate depth reduction (Figure 9) and find that removing entire layers also causes catastrophic degradation in generated outputs. These results indicate that, for generation components, static compression along either width or depth alone struggles to preserve the performance of the original model.

On the other hand, compressing the attention layers leads to substantial degradation in both depth and width settings. As shown in Figure 10, applying more than a 10% reduction results in noticeable performance drops.

(a) Depth reduction. (b) Width reduction.

Figure 10: **Attention layers in generation components are highly sensitive to both depth and width reduction.** (a) Depth reduction by removing 7, 4, 2, or 0 layers. (b) Width reduction at ratios of 50%, 25%, 10%, and 0.

B Attention Head Partition

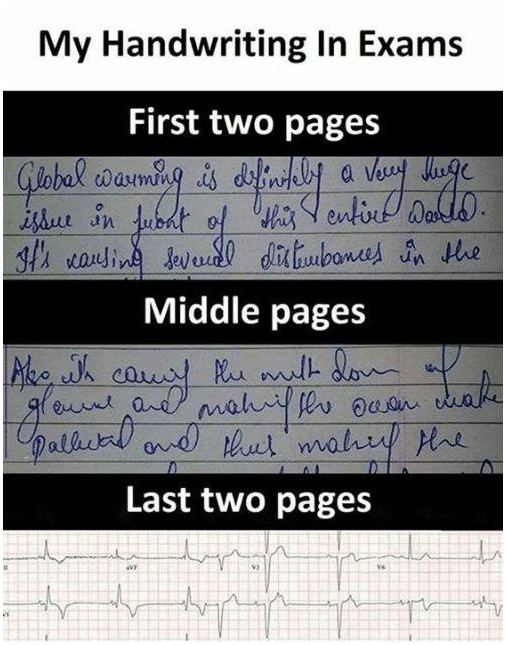
Table 5: Attention head partition on the understanding component remains effective at 50% sparsity.

Model	Compressed Layers	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
BAGEL	N/A	0.99	0.94	0.81	0.95	0.72	0.77	<u>0.86</u>
	3-27	0.97	0.87	0.66	0.88	0.33	0.31	<u>0.67</u>
	4-27	0.98	0.91	0.72	0.89	0.41	0.40	<u>0.72</u>

While our main focus is width reduction in MLP layers, since they account for most of the parameters, our compression metric can be seamlessly extended to attention by computing the importance scores of attention heads. Attention Head Partition still works for understanding components as shown in Table 5.

C Dilemma of Depth Reduction on Understanding Tasks

While reducing depth in the understanding component has only a limited impact on generation tasks, it severely harms multimodal understanding performance, as shown in Table 6. Moreover, Figure 11 illustrates



Meme: "My Handwriting In Exams"

Question: Can someone explain what's funny about this meme?

Neuron Partition:
The humor in this meme comes from the exaggerated change in handwriting style as the exam progresses. At the beginning, the handwriting is clear and legible, indicating a confident and well-prepared student. However, as the exam continues, the handwriting becomes increasingly difficult to read, suggesting that the student is becoming more anxious and less focused.

Depth Reduction:
The comedic resurgence biting meme portrays a comedic portrayal portraying a trope portraying idioms portraying idioms portraying idioms portraying idioms portraying idioms portraying idioms portraying exposes exposing exposing injuring injuring injuring injuring injuring injuring injuring injuring injuring injuring.....

Figure 11: Neuron partition preserves coherent outputs while depth reduction causes degenerate repetition. Both methods use a 50% compression ratio. The depth-reduced model collapses into repeating a single word.

Table 6: Depth reduction on understanding components degrades performance in understanding tasks.

Model	Sparsity	MME-P	MME-C	MMMU	MMBench	MMVP
Ming-Omni	–	1584.3	670.4	66.7	86.7	54.6
	50%	1197.2	308.2	51.7	81.2	46.0
BAGEL	–	1684.8	696.7	65.0	88.1	69.6
	50%	304.5	127.1	16.7	18.6	23.1

that the reduced-depth model fails to produce coherent answers, collapsing after only a few generated tokens. We attribute this to error accumulation in autoregressive decoding: small deviations in early generated tokens compound over time, ultimately causing the entire response to collapse.

D Comparison with Compression Methods

While neuron partition leverages reconstruction errors to estimate the importance of each neuron, another mainstream approach relies on gradient information to approximate the impact of neuron removal. In Table 7, we adopt gradient-based metrics, such as those used in LLM-Pruner (Ma et al., 2023), as references to provide a more comprehensive comparison for our neuron partition method. Neuron Partition not only demonstrates superior performance but also eliminates the dependence on labeled data and explicit gradient computation, thereby enabling more efficient deployment of training-free sparsity methods.

On the other hand, neuron partition also achieves competitive performance compared to quantization. Specifically, in Table 8, we apply AWQ to quantize the understanding component of Qwen-Image, i.e., the Qwen-VL backbone. Notably, neuron partition achieves an overall score of 0.90 at a 50% compression ratio, outperforming the 0.88 obtained by 4-bit quantization. This stands in contrast to traditional LLM

Table 7: Neuron partition matches gradient-based LLM-Pruner without requiring labeled data.

Model	Metrics	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Ming-Omni	LLM-Pruner	0.96	0.82	0.72	0.85	0.47	0.55	0.70
	Neuron Partition	0.96	0.81	0.58	0.86	0.49	0.56	0.71

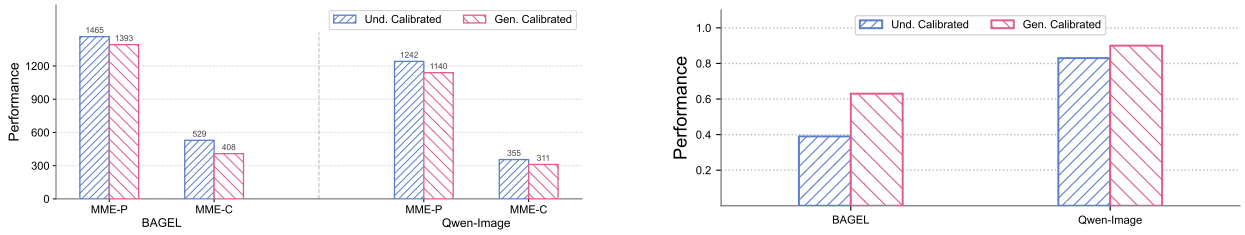
Table 8: Neuron partition at 50% sparsity outperforms 4-bit quantization on the understanding component.

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
Qwen-Image	0.99	0.98	0.91	0.94	0.80	0.89	0.92
w/ 4-bit quantization	0.99	0.97	0.93	0.92	0.79	0.70	0.88
w/ 50% neuron partition	0.99	0.94	0.94	0.93	0.76	0.87	0.90

compression, where pruning at similar ratios typically leads to noticeable performance degradation and remains significantly weaker than 4-bit quantization.

E Calibration Data for Neuron Partitioning

The choice of calibration data influences the measurement of activation scores and thus determines which neurons are retained. To comprehensively assess this effect, we conduct an ablation study on MME for understanding and GenEval for generation. Specifically, we sample a small number of instances from each dataset for calibration, compress the models using these calibration sets, and then evaluate them on MME and GenEval, as shown in Figures 12a and 12b, respectively. Notably, calibration data that align with the target task consistently yield better performance in both understanding and generation. This suggests that unified models rely on distinct neuron partitions for different tasks. Moreover, our test-time few-shot compression, which directly uses a few test samples for calibration, seamlessly adapts to downstream tasks and achieves competitive performance.



(a) Understanding tasks.

(b) Generation tasks.

Figure 12: Task-aligned calibration consistently yields better performance.