MASKERADE: TOKEN-ROUTED MASK EXPERTS FOR DENSE-TO-MOE UPCYCLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparsely activated Mixture-of-Experts (MoE) models increase model capacity without a proportional increase in per-token computation. Dense-to-MoE upcycling reuses pretrained dense models to construct such systems, commonly by copying feed-forward networks (FFNs) into independently trained experts. We introduce **MASKerade**, a dense-to-MoE training method that instead learns experts as sparse subnetworks of a frozen pretrained FFN. Each expert is defined by a learned binary mask, and a token-level router selects which masked FFNs to execute and combine. The router and mask scores are optimized jointly, while the underlying FFN weight values remain unchanged. This formulation supports neuron-structured, semi-structured, and unstructured experts within the same routing architecture. Our main configuration uses four 2:4 experts with top-2 routing, where two half-dense expert passes have the nominal FFN arithmetic of one dense pass, without requiring independent expert weight matrices. On five vision-language benchmarks with Qwen and Gemma backbones, this configuration achieves the highest performance among the compared baselines. Comparisons across mask granularities, routing interventions, and compute-matched controls distinguish the effects of learned connectivity from expert activation count. These results establish mask learning over frozen weights as a practical alternative for constructing token-routed MoE experts.

1 INTRODUCTION

Mixture-of-Experts (MoE) models separate the number of available parameters from per-token computation (Jacobs et al., 1991). A sparsely gated layer selects a small subset of experts and combines their outputs, rather than evaluating every expert for every input (Shazeer et al., 2017; Fedus et al., 2022). Dense-to-MoE upcycling makes this approach accessible from existing pretrained models by converting selected feed-forward networks (FFNs) into routed expert layers, reusing the investment in dense pretraining (Komatsuzaki et al., 2023). This is particularly attractive for multimodal instruction tuning, where a single model must process heterogeneous visual and textual inputs (Lin et al., 2026; Chen et al., 2024a).

An important design decision in upcycling is *what constitutes an expert*. Replication-based approaches initialize multiple FFNs from the same dense FFN and then learn separate weight values. Some of them partially reinitialize these copies to encourage their differentiation (Nakamura et al., 2025). Other methods construct smaller experts by partitioning or pruning pretrained FFNs (Zhang et al., 2022; Zhu et al., 2024; Feng et al., 2025), or introduce expert-specific adapters and low-rank updates (Wu et al., 2024; Chen et al., 2024a).

We investigate a different parameterization: *can experts be learned by selecting connections in a pretrained FFN?* Learning a mask can change a network's function without changing its retained weights, as demonstrated by supermask learning and learnable LLM pruning (Ramanujan et al., 2020; Fang et al., 2024; Zhang et al., 2026; 2025). In an MoE, several such masks can coexist over the same weights. The router then chooses among alternative sparse computations. This replaces expert-specific weight adaptation with expert-specific connectivity learning.

We propose **MASKerade**, which jointly learns a bank of binary expert masks and a token-level router over a frozen FFN. For a given token, the router selects a small subset of masks, executes the corresponding masked FFNs, and combines their outputs using routing weights. Conceptually,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

this resembles a **MASKerade (*masquerade*, with "qu" replaced by "k")** in which tokens enter the same shared FFN but *wear* different masks. MASKerade studies a specific combination that consists of direct masking of the *frozen pretrained FFN weights*, token-level sparse routing, and hard connectivity constraints. We evaluate MASKerade through multimodal instruction tuning of Qwen and Gemma backbones (Yang et al., 2025; Team et al., 2025). MASKerade attains the best performance against the external methods across five benchmarks. Our contributions are:

- We introduce MASKerade, a dense-to-MoE upcycling method that constructs experts as learned binary masks over frozen pretrained FFNs.
- MASKerade jointly optimizes expert masks and token routing under prescribed sparsity constraints, accommodating neuron-structured, semi-structured, and unstructured masks.
- Our experiments and analysis demonstrate that effective dense-to-MoE adaptation can be trained without updating the underlying pretrained FFN weights.

2 Related Work

Sparse MoE and dense-to-MoE conversion. Sparsely gated MoE layers use input-dependent expert selection to increase the available model capacity while restricting active computation (Shazeer et al., 2017; Fedus et al., 2022). At the routing level, MaskMoE uses fixed, token-frequency-dependent masks on routing logits to restrict the experts accessible to each vocabulary token (Su et al., 2024). These masks constrain expert visibility rather than define sparse FFN subnetworks. Sparse Upcycling initializes expert layers from pretrained dense checkpoints (Komatsuzaki et al., 2023), and Drop-Upcycling uses partial reinitialization to reduce the similarity of copied experts (Nakamura et al., 2025). MoEfication interprets FFN neurons as groups of experts and learns to select relevant groups (Zhang et al., 2022). LLaMA-MoE combines expert partitioning with continual pretraining (Zhu et al., 2024), while DIVE uses domain affinity and calibration-dependent pruning to construct diverse experts (Feng et al., 2025). ToMoE learns dynamic structural pruning through a hypernetwork to convert dense models into routed, narrower computations (Gao et al., 2026). MASKerade also starts from a dense model, but learns masks directly on its frozen FFN tensors instead of requiring independently fine-tuned FFN copies.

Learning sparse subnetworks. Supermask methods learn useful subnetworks while keeping their underlying weight values fixed (Ramanujan et al., 2020). For pretrained LLMs, Wanda selects weights using weight magnitudes and activation statistics without gradient-based mask training (Sun et al., 2024). SparseLLM formulates global pruning as coordinated optimization subproblems, using auxiliary variables to make the problem tractable (Bai et al., 2024). MaskLLM learns a distribution over legal semi-structured patterns using Gumbel-Softmax (Fang et al., 2024). Beyond compression, Mask Fine-Tuning (MFT) uses the original fine-tuning objective to learn binary weight masks over a frozen, already fine-tuned LLM, seeking further capability improvements (Zhang et al., 2026). Zhang et al. (2025) extend this approach to pretrained vision-language models, learning hard or soft masks over language-model and projector weights. These methods motivate learning connectivity rather than only pruning by a fixed importance criterion. MASKerade differs from a single learned subnetwork by maintaining multiple masks and selecting them conditionally for each token.

Mask-based and interpolated experts. MoM explicitly studies learnable masks as an alternative to conventional experts (Liu et al., 2025). SMAT learns sparse interpolated experts for few-shot meta-tuning of vision models (Chen et al., 2024b). SIMoE applies sparse expert discovery to LLM instruction tuning, where structured masks act on a shared trainable parameter delta added to the frozen pretrained parameters. Their main routing configuration merges experts at the instance level (Chen et al., 2025). CDSP-MoE learns expert topology within a shared parameter subspace using gradient-conflict signals (Gan & Lei, 2026). Our distinction is to route tokens through hard-masked versions of the pretrained FFN itself, with no trainable shared delta.

3 Method

3.1 Experts as Frozen-Weight Subnetworks

Let θ_0 denote a pretrained dense model and $\mathcal{I}$ the set of FFN layers to convert. In a converted layer ℓ, write its frozen FFN parameters as W_ℓ. Conventional replicated experts learn separate parameter sets $\{W_{\ell,e}\}_{e=1}^{E}$ initialized from W_ℓ. MASKerade instead defines

$$\widetilde{W}_{\ell,e}^{(a)} = W_\ell^{(a)} \odot M_{\ell,e}^{(a)}, \qquad M_{\ell,e}^{(a)} \in \{0,1\}^{\mathrm{shape}(W_\ell^{(a)})}. \tag{1}$$

where a indexes the FFN projections and $\odot$ is elementwise multiplication.

For the gated FFNs used by our backbones, with gate, up, and down projections $W^{(g)}, W^{(u)} \in \mathbb{R}^{d_{\mathrm{ff}} \times d}$ and $W^{(d)} \in \mathbb{R}^{d \times d_{\mathrm{ff}}}$, expert e computes

$$F_{\ell,e}(h) = \widetilde{W}_{\ell,e}^{(d)} \left[\phi_\ell\left(\widetilde{W}_{\ell,e}^{(g)} h\right) \odot \left(\widetilde{W}_{\ell,e}^{(u)} h\right) \right]. \tag{2}$$

In our configuration, the pretrained parameters remain frozen throughout upcycling.

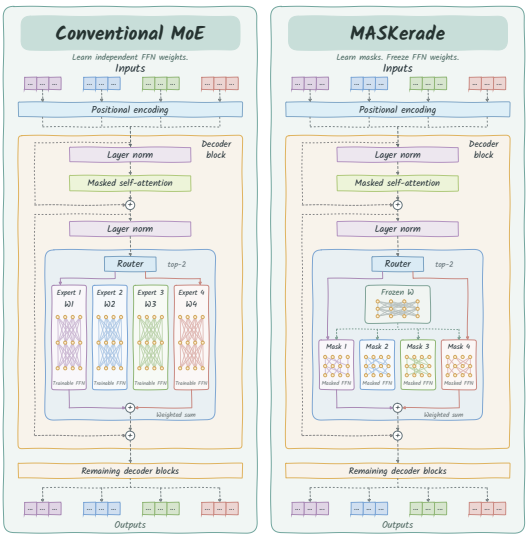


Figure 1: **Conventional MoE and MASKerade.** Tokens select independent FFNs (left) or learned masks over one frozen FFN (right).

3.2 Token-Level Routing

Routing selects from a learned bank of expert masks shared across tokens. A trainable linear router $R_\ell \in \mathbb{R}^{E \times d}$ maps each token's hidden state to probabilities $p_{\ell,e}(h) = \mathrm{softmax}(R_\ell h)_e$. For k experts, deterministic top-k selection gives $\mathcal{S}_\ell(h) = \mathrm{TopK}(p_\ell(h), k)$. Our routing formulation uses deterministic top-k expert selection and dropless dispatch during both training and inference. The selected expert outputs are combined as

$$F_\ell^{\mathrm{MoE}}(h) = \sum_{e \in \mathcal{S}_\ell(h)} g_{\ell,e}(h) F_{\ell,e}(h), \qquad g_{\ell,e}(h) = \begin{cases} p_{\ell,e}(h), & k = 1, \\ \dfrac{p_{\ell,e}(h)}{\sum_{j \in \mathcal{S}_\ell(h)} p_{\ell,j}(h)}, & k \geq 2. \end{cases} \tag{3}$$

For $k \geq 2$, the coefficients sum to one over the selected experts. For top-1 routing with $E > 1$, retaining the selected expert's softmax probability preserves a differentiable path from the task loss to the router. When $E = 1$, routing will be bypassed.

3.3 Mask Parameterization and Granularity

Each mask is obtained from continuous trainable scores S using a hard selection operator $\mathcal{H}$, so $M = \mathcal{H}(S)$ is binary. We denote the retained fraction by ρ and consider three families.

Semi-structured. For an $N{:}M$ pattern, each row of a weight matrix is partitioned into contiguous groups of M entries along its dimension. For each group B we keep the N largest score magnitudes:

$$M_{\ell,e,i}^{(a)} = \mathbf{1}\left[i \in \mathrm{TopN}_{j \in B} \left| S_{\ell,e,j}^{(a)} \right| \right], \qquad \sum_{i \in B} M_{\ell,e,i}^{(a)} = N. \tag{4}$$

Unstructured. For each expert's projection tensor, let P denote its number of entries and $q = \lceil \rho P \rceil$ the retained count. We select the q largest score magnitudes over all flattened positions:

$$M_{\ell,e,i}^{(a)} = \mathbf{1}\left[i \in \mathrm{TopK}_{1 \leq j \leq P} \left(\left| S_{\ell,e,j}^{(a)} \right|, q \right) \right], \qquad \sum_{i=1}^{P} M_{\ell,e,i}^{(a)} = q.$$

Structured. Each expert instead learns a score vector over the d_{ff} intermediate neurons, retaining a fraction ρ. Its binary mask $m_{\ell,e}$ is shared across the corresponding gate/up rows and down columns:

$$M_{\ell,e}^{(g)} = M_{\ell,e}^{(u)} = m_{\ell,e} \mathbf{1}_d^\top, \qquad M_{\ell,e}^{(d)} = \mathbf{1}_d m_{\ell,e}^\top. \tag{5}$$

Table 1: Main comparison with existing methods on small models. All methods start from the same Zero-shot checkpoint. We learn masks as experts over frozen FFN weights. Best and second-best results in each metric are shown in **bold** and underlined, respectively.

	Method	Act. Params.	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B	Zero-shot (Yang et al., 2025)	1.72B	25.97	962.0	83.59	60.93	32.38
	Dense MLP	1.72B	58.41±0.35	1300.4±18.4	85.01±0.24	61.16±0.32	49.19±0.32
	MoEfication (Zhang et al., 2022)	1.46B	53.60±0.20	1280.5±15.8	84.75±0.33	60.11±0.96	49.94±0.55
	LLaVA-MoLE (Chen et al., 2024a)	1.91B	58.05±0.38	1295.0±18.3	84.97±0.36	60.24±0.09	50.25±0.06
	PESC (Wu et al., 2024)	1.73B	56.81±0.36	1309.1±11.3	84.31±0.29	61.60±0.57	49.95±0.25
	LLaMA-MoE (Zhu et al., 2024)	1.46B	57.85±0.16	1233.3±17.5	85.45±0.39	60.56±0.58	50.90±0.58
	Drop-Upcycling (Nakamura et al., 2025)	2.25B	60.77±0.24	1312.9±13.9	85.66±0.19	61.60±0.65	50.82±0.07
	MoM (Liu et al., 2025)	2.25B	53.48±0.13	1283.2±16.0	85.73±0.25	62.37±0.75	50.49±0.27
	DIVE (Feng et al., 2025)	1.46B	57.87±0.16	1300.6±16.7	83.80±0.34	60.84±0.76	50.55±0.62
	MoE-LLaVA (Lin et al., 2026)	2.25B	60.06±0.19	1315.3±11.8	85.19±0.22	62.07±0.76	51.36±0.32
	ToMoE (Gao et al., 2026)	1.46B	58.54±0.22	1302.2±16.2	85.66±0.40	61.61±0.50	50.77±0.22
	[Ours] MASKerade (Semi-structured 2:4)	1.72B	**62.00**±0.28	**1332.4**±11.9	**86.49**±0.16	**63.48**±0.57	**52.50**±0.37
Gemma3-1B	Zero-shot (Team et al., 2025)	1.00B	28.25	823.0	82.13	36.74	16.66
	Dense MLP	1.00B	56.32±0.12	1233.4±13.9	85.71±0.16	52.20±0.69	41.30±0.31
	MoEfication (Zhang et al., 2022)	0.84B	55.36±0.28	1212.9±13.5	80.92±0.35	50.33±0.55	40.68±0.58
	LLaVA-MoLE (Chen et al., 2024a)	1.11B	55.66±0.10	1244.8±16.4	85.77±0.24	51.56±0.44	40.54±0.83
	PESC (Wu et al., 2024)	1.00B	57.54±0.29	1267.1±18.1	85.00±0.36	53.91±0.52	42.49±0.71
	LLaMA-MoE (Zhu et al., 2024)	0.84B	57.67±0.33	1234.7±11.2	85.00±0.42	52.71±0.63	40.96±0.67
	Drop-Upcycling (Nakamura et al., 2025)	1.31B	58.94±0.19	1287.5±14.4	85.59±0.40	54.24±0.40	42.55±0.68
	MoM (Liu et al., 2025)	1.31B	56.79±0.29	1249.1±18.9	82.10±0.40	50.07±0.51	39.45±0.79
	DIVE (Feng et al., 2025)	0.85B	51.66±0.27	1129.3±13.2	82.80±0.28	50.84±0.63	41.79±0.58
	MoE-LLaVA (Lin et al., 2026)	1.31B	59.34±0.45	1288.0±18.4	85.80±0.18	54.83±0.42	41.91±0.50
	ToMoE (Gao et al., 2026)	0.84B	53.96±0.15	1123.0±17.9	81.37±0.43	51.98±0.52	40.63±0.32
	[Ours] MASKerade (Semi-structured 2:4)	1.00B	**60.07**±0.27	**1293.8**±11.9	**86.41**±0.10	**55.30**±0.47	**42.90**±0.47

3.4 Joint Mask and Router Training

Score initialization. We consider two initialization schemes for the mask scores. With *random initialization*, each expert's scores are sampled independently using Kaiming-uniform initialization. With *importance initialization*, weight-level scores instead use the pretrained weight magnitudes with independent expert-specific noise:

$$S^{(a)}_{\ell,e} = |W^{(a)}_{\ell}| + \epsilon^{(a)}_{\ell,e},$$
$$\epsilon^{(a)}_{\ell,e,i} \sim \mathcal{N}\left(0, \left[\alpha_{\text{init}}\ \text{std}(|W^{(a)}_{\ell}|)\right]^2\right). \tag{6}$$

$\alpha_{\text{init}} \geq 0$ controls the relative noise scale. The magnitude term favors high-magnitude connections, while independent perturbations help break symmetry between expert initializations. For neuron-structured masks, the importance of neuron j is $u_{\ell,j} = (\text{mean}\,|W^{(g)}_{\ell,j,:}| + \text{mean}\,|W^{(u)}_{\ell,j,:}| + \text{mean}\,|W^{(d)}_{\ell,:,j}|)/3$. We initialize its score as $S_{\ell,e,j} = u_{\ell,j} + \epsilon_{\ell,e,j}$, with independent Gaussian noise of standard deviation $\alpha_{\text{init}}\,\text{std}(u_\ell)$ across neurons and experts. Thus, structured importance aggregates the three weight slices affected by the same neuron mask.

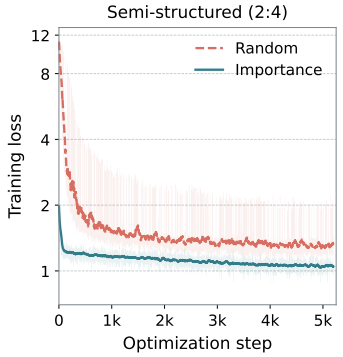


Figure 2: Initialization loss for 2:4 masks on Qwen3-1.7B.

The loss curve comparison in Fig. 2 shows a substantially lower initial loss with importance initialization, as well as a lower training loss near the end of the run. Motivated by this observation, we adopt importance initialization as the default for subsequent MASKerade experiments. Tab. 5 and Fig. 5 show more analysis.

Gradient estimator. Hard selection is not differentiable. Following straight-through and supermask learning (Bengio et al., 2013; Zhang et al., 2026), we use hard masks in the forward pass and a straight-through surrogate to propagate gradients to the continuous mask scores in the backward

Table 2: Main comparison with existing methods on large models. All methods start from the same Zero-shot checkpoint. We learn masks as experts over frozen FFN weights. Best and second-best results in each metric are shown in **bold** and underlined, respectively.

	Method	Act. Params.	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3.6-27B	Zero-shot (Yang et al., 2025)	26.90B	61.39	1692.1	86.23	92.74	71.65
	Dense MLP	26.90B	64.13	1716.0	86.90	93.38	73.41
	MoEfication (Zhang et al., 2022)	22.62B	61.05	1691.2	86.15	91.55	72.00
	LLaVA-MoLE (Chen et al., 2024a)	29.82B	65.71	1702.5	86.47	92.17	74.08
	PESC (Wu et al., 2024)	26.94B	65.61	1717.4	87.08	91.18	77.33
	LLaMA-MoE (Zhu et al., 2024)	22.62B	64.33	1715.5	86.78	93.46	78.63
	Drop-Upcycling (Nakamura et al., 2025)	35.45B	65.75	1715.3	87.26	94.12	79.04
	MoM (Liu et al., 2025)	35.45B	61.27	1689.3	86.29	88.82	79.95
	DIVE (Feng et al., 2025)	22.65B	64.58	1708.0	87.16	90.83	79.68
	MoE-LLaVA (Lin et al., 2026)	35.45B	65.58	1715.4	87.47	94.42	80.27
	ToMoE (Gao et al., 2026)	22.62B	61.87	1681.2	87.58	89.57	71.80
	[Ours] MASKerade (Semi-structured 2:4)	26.90B	**66.37**	**1724.9**	**87.68**	**94.79**	**81.08**
Gemma4-31B	Zero-shot (Team et al., 2026)	30.70B	47.46	1573.8	82.48	83.24	68.28
	Dense MLP	30.70B	60.81	1626.6	86.00	83.59	72.67
	MoEfication (Zhang et al., 2022)	25.50B	58.59	1559.6	79.74	81.17	68.02
	LLaVA-MoLE (Chen et al., 2024a)	34.24B	61.24	1578.3	86.52	74.96	73.26
	PESC (Wu et al., 2024)	30.74B	63.11	1651.3	87.07	86.07	74.96
	LLaMA-MoE (Zhu et al., 2024)	25.50B	63.44	1629.5	86.39	77.24	74.52
	Drop-Upcycling (Nakamura et al., 2025)	41.10B	65.22	1676.2	86.86	81.51	75.11
	MoM (Liu et al., 2025)	41.10B	58.81	1581.4	82.89	84.28	68.87
	DIVE (Feng et al., 2025)	25.53B	61.25	1573.3	83.36	73.82	72.60
	MoE-LLaVA (Lin et al., 2026)	41.10B	64.86	1678.4	86.63	84.50	74.88
	ToMoE (Gao et al., 2026)	25.50B	60.58	1575.6	85.34	83.29	69.37
	[Ours] MASKerade (Semi-structured 2:4)	30.70B	**65.30**	**1694.3**	**87.33**	**87.62**	**75.44**

pass. For each masked projection, the learning signal at the mask is

$$\frac{\partial \mathcal{L}}{\partial M_{\ell,e}^{(a)}} = W_{\ell}^{(a)} \odot \frac{\partial \mathcal{L}}{\partial \widetilde{W}_{\ell,e}^{(a)}}. \tag{7}$$

For neuron-structured masks, gradients from the incident connections are accumulated into the shared neuron scores. All masks are recomputed from the updated scores.

Training objective. Mask scores and routers are learned jointly through supervised autoregressive loss and a load-balancing term:

$$\min_{\{S_\ell, R_\ell\}_{\ell \in \mathcal{I}}} \mathcal{L}_{\mathrm{SFT}}(\theta_0, S, R) + \lambda_{\mathrm{bal}} \sum_{\ell \in \mathcal{I}} \mathcal{L}_{\mathrm{bal}}^{(\ell)}, \qquad \theta_0 \text{ fixed}, \tag{8}$$

where $\mathcal{L}_{\mathrm{SFT}}$ is the mean negative log-likelihood of the supervised response tokens, and λ_{bal} controls the strength of load balancing. Following the probability-load balancing principle of (Fedus et al., 2022), we measure expert usage over all selected assignments and define the penalty over T tokens entering a converted layer as

$$\bar{p}_{\ell,e} = \frac{1}{T} \sum_{t=1}^{T} p_{\ell,e}(h_t), \quad f_{\ell,e} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[e \in \mathcal{S}_\ell(h_t)], \quad \mathcal{L}_{\mathrm{bal}}^{(\ell)} = E \sum_{e=1}^{E} \bar{p}_{\ell,e} f_{\ell,e}. \tag{9}$$

$\bar{p}_{\ell,e}$ measures the mean routing probability, and $f_{\ell,e}$ measures the fraction of tokens assigned to expert e. Each token contributes to all k selected experts, so $\sum_{e=1}^{E} f_{\ell,e} = k$.

4 Experiments

We evaluate MASKerade as a dense-to-MoE adaptation method for multimodal instruction tuning.

Table 3: Mask-based MoE comparison. Best and second-best results in each metric are shown in **bold** and underlined, respectively.

	Method	GQA	MME	POPE	SQA	TextVQA
Qwen3-1.7B	Zero-shot	25.97	962.0	83.59	60.93	32.38
	Dense MLP	58.41±0.35	1300.4±18.4	85.01±0.24	61.16±0.32	49.19±0.32
	Random 2:4	43.36±0.30	871.8±21.3	82.15±0.16	40.75±0.42	28.22±0.48
	Magnitude 2:4	51.54±0.26	1127.6±14.7	85.38±0.53	58.34±0.80	43.15±0.71
	Wanda 2:4 (Sun et al., 2024)	51.97±0.53	1148.2±23.8	83.96±0.44	60.81±0.64	44.01±0.27
	Wanda Unstructured (Sun et al., 2024)	52.24±0.48	1187.4±37.4	86.06±0.27	63.01±0.52	45.21±0.22
	[Ours] Structured	58.81±0.12	1302.0±18.3	85.06±0.48	61.58±0.52	51.05±0.24
	[Ours] Unstructured	**62.06±0.30**	1315.8±11.2	86.27±0.32	**63.71±0.42**	**52.83±0.13**
	[Ours] Semi-tructured 2:4	62.00±0.28	**1332.4±11.9**	**86.49±0.16**	63.48±0.57	52.50±0.37
Gemma3-1B	Zero-shot	28.25	823.0	82.13	36.74	16.66
	Dense MLP	56.32±0.12	1233.4±13.9	85.71±0.16	52.20±0.69	41.30±0.31
	Random 2:4	36.45±0.64	732.8±20.6	64.37±0.69	23.24±0.82	13.50±0.78
	Magnitude 2:4	46.10±0.36	920.5±18.3	83.10±0.56	34.03±0.52	28.87±0.58
	Wanda 2:4 (Sun et al., 2024)	47.45±0.16	904.8±6.4	82.87±0.70	36.23±0.62	30.25±0.75
	Wanda Unstructured (Sun et al., 2024)	47.38±0.17	1001.0±15.7	80.62±0.78	35.94±0.43	32.34±0.97
	[Ours] Structured	56.88±0.47	1253.5±12.9	85.81±0.34	52.33±0.84	41.58±0.37
	[Ours] Unstructured	**60.38±0.11**	1288.9±7.1	85.87±0.58	**56.68±0.74**	42.41±0.49
	[Ours] Semi-tructured 2:4	60.07±0.27	**1293.8±11.9**	**86.41±0.10**	55.30±0.47	**42.90±0.47**

4.1 Experimental Setup

Backbones and training data. We conduct our experiments on both small and large Vision-Language Models (VLMs). For small models, we use LLaVA-style vision-language models (Liu et al., 2023) with Qwen3-1.7B (Yang et al., 2025) and Gemma3-1B (Team et al., 2025) language backbones. Both use a CLIP ViT-L/14 image encoder (Radford et al., 2021). For large models, we use Qwen3.6-27B (Yang et al., 2025) and Gemma4-31B (Team et al., 2026) as the VLM backbones. Adaptation uses the 665K instruction mixture in the MoE-LLaVA pipeline (Lin et al., 2026).

Baselines and benchmarks. We compare with representative strategies. We adapt the baseline expert constructions to a common starting VLM and instruction dataset within each backbone. Expert widths, trainable modules, and learning rates remain method-specific. We evaluate on five benchmarks, including GQA (Hudson & Manning, 2019), MME (Fu et al., 2023), POPE (Li et al., 2023), ScienceQA (Lu et al., 2022), and TextVQA (Singh et al., 2019).

Implementation details. We convert alternating FFN layers, with 14 of 28 for Qwen3-1.7B and 13 of 26 for Gemma3-1B, and 32 of 64 for Qwen3.6-27B and 30 of 60 for Gemma4-31B (the comparison of different layers is in Tab. 18). Layer indices are zero-based and even. Each converted layer has $E = 4$ mask experts with top-2 routing. We use a cosine learning-rate schedule with 3% warmup, zero weight decay, and bfloat16 training. The default initialization noise is $\alpha_{\text{init}} = 0.1$ and the balance coefficient is $\lambda_{\text{bal}} = 0.01$. We run most experiments with three random seeds and report the mean and sample standard deviation. More implementation details are in Tab. 9.

4.2 Main Results

Main Comparison. Tab. 1 and Tab. 2 compare MASKerade with representative MoE adaptation methods from the same *Zero-shot* checkpoint for each backbone. Among the results, MASKerade achieves the highest reported performance on all five benchmarks. *Act. Params.* denotes nominal activated language-model parameter counts, with overlapping masked weights counted separately across selected experts, rather than unique weights, exact FLOPs, or measured latency. Its nominal activated-parameter counts match Dense MLP and are below most baselines. We also provide a low-budget MASKerade compared with the lowest activated-computation methods in Tab. 7. Overall, these results support MASKerade as an effective alternative to learning independent experts.

Table 4: Router learning and inference-time interventions with four 2:4 experts and top-2 routing. Frozen-random routers remain fixed during mask learning.

Setting	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B					
Trained	62.00±0.28	1332.4±11.9	86.49±0.16	63.48±0.57	52.50±0.37
Random (frozen during training)	60.02±0.07	1307.7±21.0	85.96±0.45	61.27±1.05	50.97±0.28
Random (after training)	59.30±0.39	1297.8±12.6	84.60±0.11	60.32±0.66	49.71±0.58
Label-permuted	56.86±0.52	1270.0±12.2	83.82±0.66	60.82±0.84	49.63±0.61
Gemma3-1B					
Trained	60.07±0.27	1293.8±11.9	86.41±0.10	55.30±0.47	42.90±0.47
Random (frozen during training)	58.92±0.14	1267.1±16.1	85.37±0.74	52.67±0.19	41.36±0.43
Random (after training)	57.89±0.18	1272.1±17.7	85.04±0.63	51.26±0.80	39.40±0.55
Label-permuted	56.13±0.28	1226.9±16.1	84.99±0.58	46.75±0.99	40.29±0.65

Table 5: Mask-score initialization on Qwen3-1.7B (four experts, top-2, 50% retention).

Initialization	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Structured 50%					
Random	58.51	1290.0	84.11	60.21	49.82
Importance	58.90	1312.0	84.56	62.08	50.85
Importance-permuted	51.54	1060.0	84.01	57.21	43.76
Unstructured 50%					
Random	60.20	1303.9	85.23	58.95	48.99
Importance	62.06	1322.4	86.27	63.71	52.94
Importance-permuted	60.88	1242.8	84.93	59.44	48.24
Semi-structured 2:4					
Random	54.44	1101.1	84.65	55.43	39.47
Importance	62.08	1335.1	86.63	63.99	52.86
Importance-permuted	60.68	1280.5	85.41	55.23	47.46

Table 6: Per-expert retention across mask granularities on Qwen3-1.7B (four experts, top-2).

Family	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Structured					
25% FFN parameters	57.00	1299.8	83.97	59.19	48.60
50% FFN parameters	58.90	1312.0	84.56	62.08	50.85
75% FFN parameters	60.99	1315.4	85.80	62.57	50.38
Unstructured					
25% FFN parameters	61.70	1322.0	85.86	60.69	50.31
50% FFN parameters	62.06	1322.4	86.27	63.71	52.94
75% FFN parameters	62.13	1375.7	86.42	65.88	54.26
Semi-structured					
1:4 mask	61.20	1312.2	85.46	61.29	51.11
2:4 mask	62.08	1335.1	86.63	63.99	52.86
3:4 mask	62.92	1338.9	87.22	65.94	53.74

Mask construction and granularity. Tab. 3 compares learned mask experts with random, magnitude-based, and Wanda-based (Sun et al., 2024) mask construction. Fixed-mask experts obtain diversity through independent random initialization (Random), expert-specific score perturbations (Magnitude and Wanda Unstructured), or source-specific calibration (Wanda 2:4). Magnitude and Wanda masks improve over random masks, but our learned 2:4 experts achieve higher reported means than all fixed-mask baselines on all five benchmarks. These comparisons support optimizing the retained connections rather than relying solely on predetermined mask construction.

Among the learned variants, both unstructured and semi-structured masks outperform neuron-structured masks on all five benchmarks for both backbones, and neither weight-level variant dominates the other. Because NVIDIA GPUs support 2:4 sparsity for accelerated computation, we use the 2:4 mask as our primary configuration and report it throughout the main experiments.

4.3 Ablations and Analysis

Router optimization. Tab. 4 distinguishes router optimization during training from interventions applied after training. The frozen-random setting keeps randomly initialized router parameters fixed while learning the masks. Jointly learning masks and routers produces higher reported means on all five benchmarks. Replacing the trained router with a randomly initialized one at evaluation, while retaining the learned masks, reduces the model performance. The label-permuted intervention changes which learned mask each expert index selects, which also worsens model performance. This sensitivity to the index-to-mask correspondence supports the importance of the learned binding between the router and masks.

Effect of score initialization. Tab. 5 compares random, importance, and importance-permuted initialization for learned masks across the three granularities. Importance initialization achieves the highest scores on all five benchmarks within each family. Its advantage over random initialization is particularly pronounced for 2:4 masks. The importance-permuted control disrupts the alignment of

Table 7: Matched nominal activation budgets on Qwen3-1.7B. Calculation denotes retained density per expert multiplied by the number of activated experts.

Configuration	Masks	Calculation	Act. params	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
MoEfication (Zhang et al., 2022)	-	-	1.46B	53.60±0.20	1280.5±15.8	84.75±0.33	60.11±0.96	49.94±0.55
LLaMA-MoE (Zhu et al., 2024)	-	-	1.46B	57.85±0.16	1233.3±17.5	85.45±0.39	60.56±0.58	50.90±0.58
DIVE (Feng et al., 2025)	-	-	1.46B	57.87±0.16	1300.6±16.7	83.80±0.34	60.84±0.76	50.55±0.62
ToMoE (Gao et al., 2026)	-	-	1.46B	58.54±0.22	1302.2±16.2	85.66±0.40	61.61±0.50	50.77±0.22
1 mask only	2:4	$50\% \times 1$	1.46B	56.51±0.32	1258.1±15.5	84.50±0.25	59.67±0.56	49.13±0.68
[Ours] 4e/top-2	1:4	$25\% \times 2$	1.46B	61.40±0.32	1322.2±11.8	85.96±0.46	61.79±0.53	50.94±0.34
[Ours] 4e/top-1	2:4	$50\% \times 1$	1.46B	61.85±0.31	1326.6±4.8	86.29±0.07	62.85±0.32	51.76±0.48
[Ours] 4e/top-2	2:4	$50\% \times 2$	1.72B	62.00±0.28	1332.4±11.9	86.49±0.16	63.48±0.57	52.50±0.37

Table 8: Standalone FFN-block latency on an H100 (BF16, CUDA graphs, balanced synthetic routing). Ratios report relative latency.

Model	dense (ms)	Vanilla (ms)	ours (ms)	Vanilla (×dense)	ours (×dense)	ours (×Vanilla)
Qwen3-1.7B	1.13	2.78	2.59	2.46	2.29	0.93
Gemma3-1B	0.78	2.04	2.02	2.62	2.59	0.99
Qwen3.6-27B	7.29	15.14	11.96	2.08	1.64	0.79
Gemma-4-31B	9.62	20.69	15.52	2.15	1.61	0.75

the importance prior with its original parameter locations. This supports retaining location-specific importance information when initializing the scores. Together, these observations support the default importance initialization across mask families.

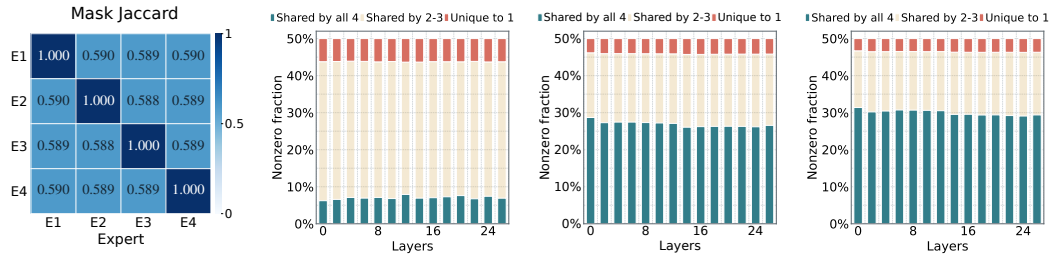
Mask ratio. Tab. 6 varies the per-expert retained fraction from 25% to 75% across the three mask families on Qwen3-1.7B, with four experts and top-2 routing. The relative performance of mask families depends on retention. At 25%, unstructured masks perform best on most benchmarks. Instead, at 50% and 75%, the best results are split between unstructured and semi-structured masks.

Matched activation budgets. Tab. 7 compares low-budget MASKerade variants with a single learned mask and existing MoE methods at the lowest activated parameters on Qwen3-1.7B in the main table. The two MASKerade variants evaluate either one 2:4 expert or two 1:4 experts per token, both using half the nominal arithmetic of one dense pass through each converted FFN. Both variants achieve higher reported means than the four external baselines on all five benchmarks. Relative to a single learned 2:4 mask, the 4e/top-1 configuration improves all five benchmarks at the same nominal budget. At this budget, 2:4/top-1 also outperforms 1:4/top-2 on all five benchmarks. The default 2:4/top-2 configuration improves further but increases nominal activated parameters from 1.46B to 1.72B. These results compare complete routed configurations and do not isolate a benefit from activating more experts alone.

Inference speed. Tab. 8 compares a standalone SwiGLU microbenchmark at the FFN dimensions of the four backbones. The dense reference evaluates one FFN pass, while Vanilla and MASKerade evaluate four-expert top-2 blocks with synthetic balanced expert assignments. Measurements use bfloat16 on an H100 GPU under CUDA-graph replay. The MASKerade's weight compression and algorithm selection occur before timing. The timed block includes router operations, token dispatch, expert computation, and weighted output combination. The normalized columns report latency ratios. Relative to Vanilla, MASKerade takes $0.93\times$ and $0.99\times$ the time at the Qwen3-1.7B and Gemma3-1B shapes, respectively. The reduction is larger at the Qwen3.6-27B and Gemma4-31B shapes, where the ratios are $0.79\times$ and $0.75\times$, corresponding to approximately 21% and 25% lower latency. The benefit of MASKerade is therefore shape-dependent and more at the larger models.

Structure of learned masks. Fig. 3 (a) reports pairwise Jaccard overlap between the learned 2:4 masks, defined as the intersection divided by the union of their retained weight positions and averaged over converted FFN projections and runs. The off-diagonal values range from 0.588 to 0.590, showing that the experts retain overlapping, but nonidentical, connectivity.

Latent shared expert. Fig. 3 (b) to (d) decompose the average retained weight density at each converted layer into connections shared by all four experts, shared by two or three, and unique to one. The fractions are normalized by the number of dense FFN weight positions and averaged over



(a) Trained overlap (b) Structured (c) Semi-structured 2:4 (d) Unstructured

Figure 3: Learned mask structure on Qwen3-1.7B: (a) 2:4 mask Jaccard overlap; (b)-(d) layer-wise sharing of retained connections.

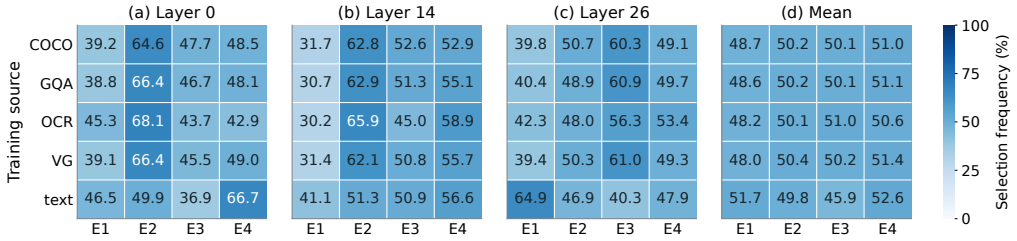


Figure 4: Layer-local expert selection by training source on Qwen3-1.7B. Panels show layers 0, 14, 26, and an equal index-wise mean over all 14 converted layers. Expert IDs are not aligned across layers. 50% indicates balanced top-2 usage.

experts, so each bar sums to 50%. The all-four intersection constitutes the *latent shared expert*: a connectivity core implicitly present in every mask. All three mask families contain both shared and expert-specific connections, but their compositions differ. For semi-structured and unstructured masks, connections shared by all four experts account for more than half of the retained entries. Neuron-structured masks have a much smaller all-four intersection and a larger partially shared component. These profiles remain broadly stable across the converted layers.

Routing across training sources. Fig. 4 reports expert selection frequencies for prompts from five training-source buckets, including COCO (Lin et al., 2015), GQA (Hudson & Manning, 2019), OCR (Mishra et al., 2019), VG (Krishna et al., 2016), and text. Each entry averages binary dispatch indicators over prefill tokens within a prompt, then equally over prompts from the same source. We use 64 prompts per source. The three displayed layers, 0, 14, and 26, are selected by depth. The mean panel equally averages all 14 converted layers, not only these three. Expert IDs are local to each layer and are not aligned before averaging, so this panel summarizes index-wise utilization, not four global experts. With two of four experts selected per token, balanced usage corresponds to a 50% selection frequency per expert. Individual layers exhibit source-related usage differences, while the all-layer mean is closer to balanced usage, obscuring the larger contrasts visible at individual layers. These observations indicate source-related, layer-local routing patterns. More detailed analysis can be found in Tab. 19.

5 Conclusion

We presented MASKerade, a dense-to-MoE upcycling method that learns token-routed mask experts over frozen pretrained FFN weights. Vision-language experiments and controlled ablations support learning expert connectivity as an effective alternative to updating independent expert weights. The learned masks retain a shared connectivity core alongside expert-specific connections, with routing adapting their use to individual tokens. Future work will investigate inference kernels that reuse overlapping computations while preserving expert-specific nonlinear transformations, and assess their end-to-end latency and memory trade-offs.

REFERENCES

Guangji Bai, Yijiang Li, Chen Ling, Kibaek Kim, and Liang Zhao. Sparsellm: Towards global pruning of pre-trained language models. In *Advances in Neural Information Processing Systems*, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms, 2024a.

Shengzhuang Chen, Jihoon Tack, Yunqiao Yang, Yee Whye Teh, Jonathan Richard Schwarz, and Ying Wei. Unleashing the power of meta-tuning for few-shot generalization through sparse interpolated experts, 2024b.

Shengzhuang Chen, Ying Wei, and Jonathan Richard Schwarz. Automatic expert discovery in LLM upcycling via sparse interpolated mixture-of-experts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. Maskllm: Learnable semi-structured sparsity for large language models. In *Advances in Neural Information Processing Systems*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022.

Yuchen Feng, Bowen Shen, Naibin Gu, Jiaxuan Zhao, Peng Fu, Zheng Lin, and Weiping Wang. DIVE into MoE: Diversity-enhanced reconstruction of large language models from dense into mixture-of-experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Yuxing Gan and Ziyu Lei. Mixture-of-experts with gradient conflict-driven subspace topology pruning for emergent modularity, 2026.

Shangqian Gao, Ting Hua, Reza Shirkavand, Chi-Heng Lin, Zheng Tang, Zhengao Li, Longge Yuan, Fangyi Li, Zeyu Zhang, Alireza Ganjdanesh, Qian Lou, Jie Xu, and Yen-Chang Hsu. Tomoe: Converting dense large language models to mixture-of-experts through dynamic structural pruning. *Transactions on Machine Learning Research*, 2026.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*, 2023.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Peiyu Liu, Tianwen Wei, Bo Zhu, Wayne Xin Zhao, and Shuicheng Yan. Masks can be learned as an alternative to experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. Drop-upcycling: Training sparse mixture of experts with partial re-initialization. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Zhenpeng Su, Zijia Lin, Xue Bai, Xing Wu, Yizhe Xiong, Haoran Lian, Guangyuan Ma, Hui Chen, Guiguang Ding, Wei Zhou, and Songlin Hu. Maskmoe: Boosting token-level learning via routing mask in mixture-of-experts, 2024.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi,

Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Gu-

ruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026.

Haoyuan Wu, Haisheng Zheng, Zhuolun He, and Bei Yu. Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

Mingyuan Zhang, Yue Bai, Yifan Wang, Yiyang Huang, and Yun Fu. Rethinking fine-tuning: Unlocking hidden capabilities in vision-language models, 2025.

Mingyuan Zhang, Yue Bai, Huan Wang, Yizhou Wang, Qihua Dong, Yitian Zhang, and Yun Fu. Boosting large language models with mask fine-tuning, 2026.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

APPENDIX

A IMPLEMENTATION DETAILS

Default configuration. Tab. 9 summarizes the MASKerade training setup. Across backbones, alternating FFNs are converted into four 2:4 mask experts with deterministic top-2 routing in both training and inference. Mask scores and routers are optimized while pretrained language-model weights remain frozen. We use importance initialization and one epoch of AdamW training with a peak learning rate of 5×10^{-4}. The small and large model settings differ in global batch size, hardware allocation, and text length limit, as listed in the table.

Table 9: Default MASKerade implementation details.

Setting	Value
Training data	665K multimodal instruction mixture
Converted FFNs	Even-indexed layers: Qwen3-1.7B, 14/28; Gemma3-1B, 13/26; Qwen3.6-27B, 32/64; Gemma4-31B, 30/60
Experts and routing	Four experts with deterministic top-2 in training and inference using dropless dispatch
Default masks	Semi-structured 2:4
Trainable LM parameters	Mask scores and routers
Pretrained LM weights	Frozen
Score initialization	Importance initialization with noise scale $\alpha_{\text{init}} = 0.1$
Gradient estimator	Straight-through estimator for hard mask selection
Optimizer	AdamW
Learning rate	5×10^{-4} for mask scores and routers
Training epochs	1
Learning-rate schedule	Cosine decay with 3% warmup
Weight decay	0
Load-balancing coefficient	$\lambda_{\text{bal}} = 0.01$
Precision	Bfloat16
Global batch size	Small: 128 (8 NVIDIA H100 GPUs); large: 256 (64 NVIDIA B200 GPUs)
Text length limit	Small: 2048; large: 1,536 tokens before visual-token expansion
Evaluation metrics	GQA: accuracy (%) MME: perception score POPE: mean F1 across three splits (%) SQA: image-subset accuracy (%) TextVQA: consensus VQA accuracy (%)

B MORE ANALYSIS

B.1 SCORE INITIALIZATION

Initialization dynamics. Fig. 5 and Fig. 6 compare training-loss trajectories under random and importance initialization on Qwen3-1.7B and Gemma3-1B. Importance initialization lowers the initial loss for unstructured and 2:4 masks on both backbones. The unstructured gap narrows during training, whereas importance-initialized 2:4 masks retain lower loss and exhibit fewer large fluctuations. The neuron-structured curves are closer on Qwen3-1.7B. For Gemma3-1B, random initialization starts with substantially lower loss and remains lower near the end of training. These curves show that the optimization effect of initialization depends on mask granularity and backbone.

Score initialization on Gemma. Tab. 10 compares random, importance, and importance-permuted initialization across the three mask granularities on Gemma3-1B. Importance initialization has the highest reported score on all five benchmarks within each family. Its advantage over random initialization is particularly pronounced for 2:4 masks on GQA and TextVQA, while the GQA difference

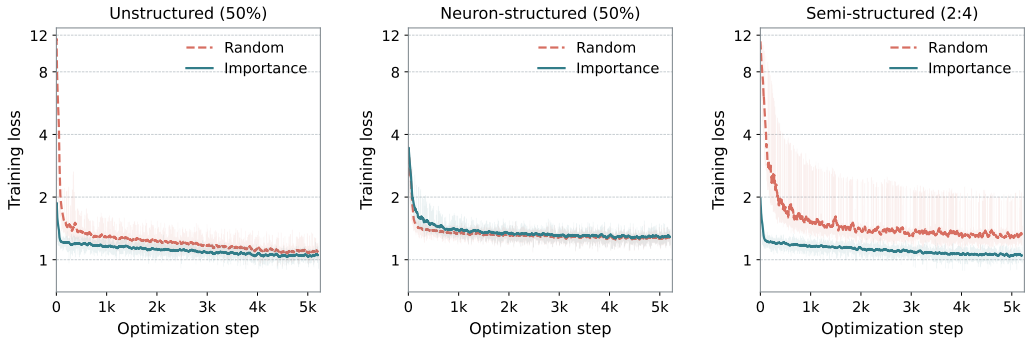


Figure 5: Random versus importance initialization across mask granularities on Qwen3-1.7B.

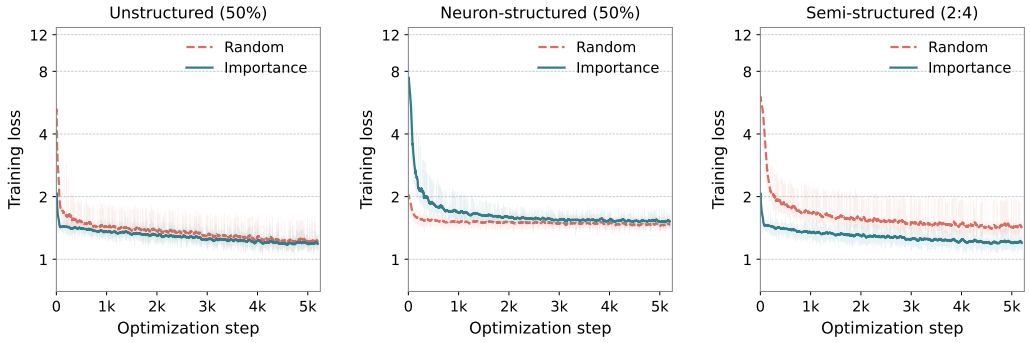


Figure 6: Random versus importance initialization across mask granularities on Gemma3-1B.

is smaller for unstructured masks. Permuting importance scores also reduces performance in every family, supporting the value of aligning the initialization prior with its original parameter locations.

Mask ratio on Gemma. Tab. 11 varies the per-expert retained fraction from 25% to 75% across the three mask families on Gemma3-1B, with four experts and top-2 routing. Unstructured masks lead on four benchmarks at each retention level. Semi-structured masks achieve the highest TextVQA scores at 25% and 50% retention, while neuron-structured masks lead on TextVQA at 75%.

Matched activation budgets on Gemma. Tab. 12 compares low-budget MASKerade variants with a single learned mask and existing MoE methods on Gemma3-1B. At 0.84B nominal activated parameters, 1:4/top-2 achieves higher reported means than the single 2:4 mask and all four external baselines on all five benchmarks. The 2:4/top-1 variant exceeds the single mask on four benchmarks, with a slightly lower GQA mean (58.24 versus 58.30). These comparisons support learning multiple token-selectable masks under a fixed nominal activation budget. Unlike on Qwen3-1.7B, 1:4/top-2 outperforms 2:4/top-1 on four benchmarks, with POPE as the exception, indicating that the preferred allocation between expert density and selection count depends on the backbone.

Structure of learned masks on Gemma. Fig. 7 (a) reports pairwise Jaccard overlap between learned 2:4 masks in the Gemma3-1B runs, averaged over converted FFN projections and runs. The off-diagonal values range from 0.489 to 0.492, indicating overlapping but nonidentical retained connections. The similar overlap across expert pairs suggests that no pair is substantially more alike than the others under this structural measure. This establishes connectivity diversity, while the usefulness of the routed assignments requires separate functional and intervention analyses.

Latent shared expert on Gemma. Fig. 7 (b) to (d) decompose the average retained weight density at each converted layer into connections shared by all four experts, shared by two or three, and unique to one. Each bar sums to 50% of dense weight positions. The all-four intersection forms a *latent shared expert*, with a smaller shared core for neuron-structured masks than for either weight-level family. In these Gemma3-1B runs, the all-four component accounts for less than half of

Table 10: Mask-score initialization on Gemma3-1B (four experts, top-2, 50% retention).

Initialization	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Structured 50%					
Random	53.13	1004.9	81.00	50.10	39.58
Importance	56.65	1251.4	85.68	51.74	41.80
Importance-permuted	53.16	1008.2	82.76	49.85	40.20
Unstructured 50%					
Random	59.92	1210.7	85.72	46.65	38.85
Importance	60.46	1295.3	86.39	56.64	42.73
Importance-permuted	59.34	1221.0	85.41	52.74	39.49
Semi-structured 2:4					
Random	51.54	1068.2	83.37	45.60	26.89
Importance	60.34	1294.4	86.35	55.40	42.81
Importance-permuted	58.89	1203.8	85.27	42.94	37.07

Table 11: Per-expert retention across mask granularities on Gemma3-1B (four experts, top-2).

Family	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Structured					
25% FFN parameters	51.25	1105.5	83.01	48.86	39.03
50% FFN parameters	56.65	1251.4	85.68	51.74	41.80
75% FFN parameters	58.95	1292.8	85.09	51.47	45.21
Unstructured					
25% FFN parameters	59.82	1256.2	85.40	54.22	41.74
50% FFN parameters	60.46	1295.3	86.39	56.64	42.73
75% FFN parameters	60.78	1315.8	86.31	56.48	43.02
Semi-structured					
1:4 mask	59.09	1233.7	85.11	52.99	41.87
2:4 mask	60.34	1294.4	86.35	55.40	42.81
3:4 mask	60.37	1304.4	86.12	55.19	43.21

Table 12: Matched nominal activation budgets on Gemma3-1B. Calculation denotes retained density per expert times the number of activated experts. Only protocol-matched runs are included.

Configuration	Masks	Calculation	Act. params	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
MoEfication (Zhang et al., 2022)	-	-	0.84B	55.36±0.28	1212.9±13.5	80.92±0.35	50.33±0.55	40.68±0.58
LLaMA-MoE (Zhu et al., 2024)	-	-	0.84B	57.67±0.33	1234.7±11.2	85.00±0.42	52.71±0.63	40.96±0.67
DIVE (Feng et al., 2025)	-	-	0.85B	51.66±0.27	1129.3±13.2	82.80±0.28	50.84±0.63	41.79±0.58
ToMoE (Gao et al., 2026)	-	-	0.84B	53.96±0.15	1123.0±17.9	81.37±0.43	51.98±0.52	40.63±0.32
1 mask only	2:4	$50\% \times 1$	0.84B	58.30±0.30	1239.9±13.7	84.87±0.38	52.59±0.69	40.07±0.68
[Ours] 4e/top-2	1:4	$25\% \times 2$	0.84B	59.09±0.33	1248.7±15.5	85.11±0.36	52.99±0.61	42.47±0.64
[Ours] 4e/top-1	2:4	$50\% \times 1$	0.84B	58.24±0.31	1246.8±11.3	85.77±0.35	52.61±0.61	41.46±0.67
[Ours] 4e/top-2	2:4	$50\% \times 2$	1.00B	60.07±0.27	1293.8±11.9	86.41±0.10	55.30±0.47	42.90±0.47

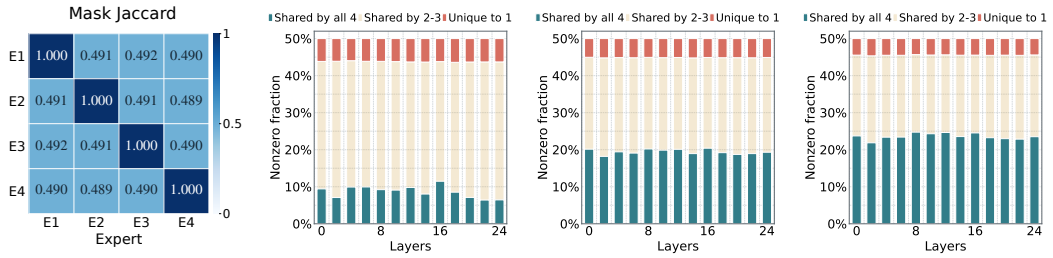
retained entries in every family. All three constructions therefore combine a common connectivity core with partially shared and expert-specific connections.

Routing across training sources. Fig. 8 reports expert selection frequencies for five training-source buckets on Gemma3-1B. Each entry averages token selection frequencies within a prompt and then equally over 64 prompts per source. Layers 0, 12, and 24 show source-related differences, with text prompts exhibiting a distinct selection pattern at layer 24. The mean over all 13 converted layers is closer to the balanced 50% usage level.

Updating shared FFN weights. Tab. 13 examines whether learning the underlying FFN weights improves mask-based adaptation. On Qwen3-1.7B, weight updates improve all five benchmarks for a single mask and four benchmarks for 4e/top-2. For Gemma3-1B, weight updates improve all five benchmarks in both configurations. The frozen-weight multi-expert configuration also exceeds Wanda prune-then-tune on all five benchmarks on Qwen3-1.7B and Gemma3-1B. Weight adaptation can therefore complement mask learning.

Number of activated experts. Tab. 14 and Tab. 16 vary top-k while keeping four 2:4 experts. These are configuration-level comparisons: Eq. 3 retains the selected probability for top-1 but renormalizes the selected probabilities for $k \geq 2$. For Qwen3-1.7B, activating all four experts gives the highest GQA, SQA, and TextVQA scores, whereas top-2 performs best on MME and POPE. For Gemma3-1B, top-2 leads on four benchmarks and top-3 gives a slightly higher MME score. Activating all four experts lowers every Gemma score relative to top-2 despite doubling the nominal selected-expert FFN computation. Since each expert retains half of the FFN weights, increasing k from one to four raises converted-FFN computation from half to twice that of a dense pass without guaranteeing better performance.

Number of available experts. Tab. 15 and Tab. 17 vary the size of the mask bank while selecting one 2:4 expert per token. For Qwen3-1.7B, moving from one to four experts improves all five benchmarks. Eight experts further improve MME and POPE but reduce the other three scores. For Gemma3-1B, two experts give the highest GQA and MME scores, while one expert leads on SQA, four on TextVQA, and eight on POPE. Larger mask banks therefore do not yield uniform gains across backbones or tasks. Selected-expert FFN computation remains fixed in this comparison,



(a) Trained overlap (b) Structured (c) Semi-structured 2:4 (d) Unstructured

Figure 7: Learned mask structure on Gemma3-1B, averaged over three runs per family: (a) 2:4 Jaccard overlap; (b)-(d) layer-wise sharing of retained connections, normalized by dense weight positions.

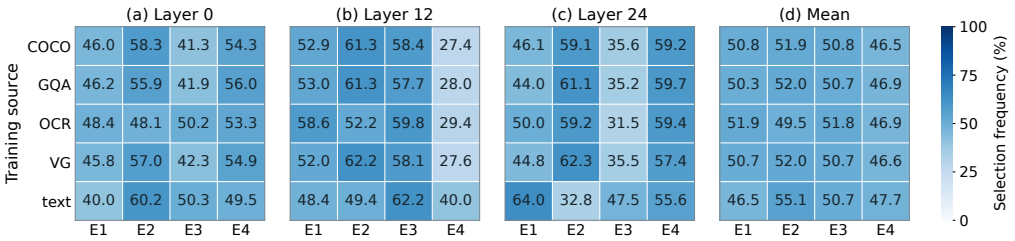


Figure 8: Layer-local expert selection by training source on Gemma3-1B. Panels show layers 0, 12, 24, and the unaligned index-wise mean over all 13 converted layers. 50% denotes balanced top-2 usage.

although the number of mask scores and the router size increase with the expert bank. The single-mask case bypasses routing, whereas multi-expert top-1 uses probability-weighted outputs. The comparison therefore does not isolate expert-bank size from output scaling.

Placement of converted layers. Tab. 18 compares where we introduce mask experts. On Qwen3-1.7B, even-indexed layers give the highest MME and TextVQA scores, while the first and second halves lead on POPE and SQA, respectively. For Gemma3-1B, even-indexed layers lead on MME, POPE, and TextVQA, and odd-indexed layers give the highest SQA score. Converting every FFN gives the highest GQA score on both backbones, but lowers the other four scores relative to even-layer conversion. These results support alternating even layers as a practical default without requiring conversion of the entire FFN stack.

Functional diversity of mask experts. Tab. 19 uses 600 examples per backbone: 300 GQA test-dev and 300 TextVQA validation examples. Identical token hidden states are fed to all four experts. Pairwise statistics average tokens within each prompt, then equally across prompts and the six expert pairs, separately for vision, prompt, and teacher-forced response tokens (including the template separator). Mean pairwise cosine similarities range from 0.65 to 0.90, with positive normalized distances throughout, showing that the masks induce different transformations rather than identical expert outputs. For both backbones, the middle probed layer exhibits lower cosine similarity and greater normalized distance than the early and late layers across all three token types. This complements the connectivity overlap analysis in Fig. 3: shared weight positions can coexist with functionally distinct outputs.

Local expert replacement. Tab. 20 uses the same 600-example subset. At a single layer, one selected expert is replaced at every answer-predictive position, starting with the last prompt token, while retaining its routing weight. NLL uses teacher-forced reference answers, excluding the final labeled token. Diagnostic accuracy uses normalized exact match after greedy decoding (up to eight new tokens). TextVQA accepts any annotated answer rather than using the official consensus score. Every reported intervention increases mean and median NLL and reduces diagnostic accuracy. Loss increases affect 57–78% of evaluated inputs, and correct-to-wrong answer flips consistently exceed

Table 13: Mask learning with frozen or trainable shared FFN weights on Qwen3-1.7B and Gemma3-1B.

	Configuration	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B	learned mask only (1 mask)	56.51±0.32	1258.1±15.5	84.50±0.25	59.67±0.56	49.13±0.68
	learned mask + trained weights (1 mask)	60.69±0.05	1310.8±0.4	85.93±0.39	61.41±0.48	50.49±0.35
	learned mask only (4e/top-2)	62.00±0.28	1332.4±11.9	86.49±0.16	63.48±0.57	52.50±0.37
	learned masks + trained weights (4e/top-2)	62.73±0.33	1353.2±1.2	86.46±0.54	63.59±0.52	52.93±0.24
	prune-then-tune (Wanda, 4e/top-2)	58.27±0.08	1282.7±20.4	85.81±0.40	61.29±0.52	49.88±0.17
Gemma3-1B	learned mask only (1 mask)	58.30±0.30	1239.9±13.7	84.87±0.38	52.59±0.69	40.07±0.68
	learned mask + trained weights (1 mask)	60.06±0.22	1240.9±4.6	85.49±0.17	52.66±0.37	40.70±0.38
	learned mask only (4e/top-2)	60.07±0.27	1293.8±11.9	86.41±0.10	55.30±0.47	42.90±0.47
	learned masks + trained weights (4e/top-2)	60.95±0.33	1296.7±4.1	86.53±0.16	55.90±0.46	44.70±0.43
	prune-then-tune (Wanda, 4e/top-2)	58.25±0.23	1255.0±13.7	84.69±0.47	51.81±0.45	37.01±0.34

Table 14: Varying top-k with four 2:4 experts on Qwen3-1.7B.

Top-k	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
1	62.20	1323.1	86.31	63.10	51.59
2	62.08	**1335.1**	**86.63**	63.99	52.86
3	62.59	1297.8	86.32	64.30	52.52
4	**62.75**	1309.0	86.32	**64.40**	**53.07**

Table 15: Varying the number of 2:4 experts with top-1 routing on Qwen3-1.7B.

N	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
1	56.36	1265.6	84.60	60.07	49.73
2	60.46	1304.4	85.98	61.87	50.99
4	**62.20**	1323.1	86.31	**63.10**	**51.59**
8	61.87	**1330.5**	**86.50**	63.04	51.55

Table 16: Varying top-k with four 2:4 experts on Gemma3-1B.

Top-k	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
1	58.24	1246.8	85.77	52.11	40.86
2	**60.34**	1294.4	**86.35**	**55.40**	**42.81**
3	59.72	**1295.9**	84.93	53.00	39.39
4	60.10	1267.6	85.25	54.49	41.48

Table 17: Varying the number of 2:4 experts with top-1 routing on Gemma3-1B.

N	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
1	58.50	1237.4	84.87	**52.64**	40.57
2	**59.10**	**1301.7**	85.79	50.72	40.27
4	58.24	1246.8	85.77	52.11	**40.86**
8	57.70	1299.4	**85.91**	51.96	36.47

wrong-to-correct flips. The effects are largest at the middle probed layers and when replacing the first-ranked selected expert, although that slot also retains the larger routing weight. Together with Tab. 4, these results support the usefulness of the learned assignments on this subset: selected masks are not freely interchangeable with unselected ones.

Hyperparameter sensitivity. Tab. 21 varies initialization noise and learning rate around the default configuration. On Qwen3-1.7B, removing initialization noise lowers all five scores. Increasing the noise to 0.3 improves GQA, MME, and POPE but lowers SQA and TextVQA. For Gemma3-1B, the default has the highest GQA, MME, and POPE scores, while removing noise slightly improves SQA and using a learning rate of 1×10^{-4} slightly improves TextVQA. The higher learning rate of 1×10^{-3} lowers all five Gemma scores. No tested alternative improves every benchmark over the default, indicating task-dependent trade-offs around the chosen recipe.

Training-time routing policy. Tab. 22 compares deterministic top-2 training with a control that selects the first expert by argmax and samples the second using Gumbel-perturbed logits. Both models use deterministic routing at evaluation, so the comparison concerns training-time expert selection. Deterministic top-2 has higher reported means on all five benchmarks for both backbones. The gains are larger on Gemma3-1B, particularly on MME and SQA, whereas the Qwen3-1.7B differences on GQA and TextVQA are small. This comparison supports the deterministic default, without implying statistically significant improvements on every benchmark or a need for stochastic expert selection during mask learning.

Table 18: Effect of MoE layer placement across model backbones (four 2:4 experts, top-2).

	Placement	MoE layers	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B	Even layers	14	62.08	**1335.1**	86.63	63.99	**52.86**
	Odd layers	14	62.14	1326.6	85.20	61.92	51.69
	First half	14	61.26	1296.0	**86.73**	58.30	50.92
	Second half	14	60.76	1299.2	85.96	**67.23**	50.42
	All layers	28	**62.47**	1298.3	86.58	60.44	50.02
Gemma3-1B	Even layers	13	60.34	**1294.4**	**86.35**	55.40	**42.81**
	Odd layers	13	60.38	1245.5	85.85	**56.12**	40.76
	First half	13	59.33	1236.3	85.05	53.15	34.11
	Second half	13	58.74	1154.9	84.91	50.62	34.29
	All layers	26	**60.57**	1192.4	85.78	46.80	37.90

Table 19: Identical-input expert outputs by token type. Cosine and Distance are pairwise averages. Distance normalizes L_2 differences by summed output norms, and e_i denotes the mean output norm.

Layer	Token type	Cosine	Distance	e_0	e_1	e_2	e_3
Qwen3-1.7B							
0	Vision	0.700	0.390	13.60	13.70	13.50	13.80
0	Prompt	0.780	0.330	9.20	9.30	9.10	9.40
0	Response	0.720	0.370	11.60	11.40	11.80	12.00
14	Vision	0.650	0.420	52.00	53.00	51.00	54.00
14	Prompt	0.730	0.370	44.00	45.00	46.00	44.50
14	Response	0.670	0.410	47.00	49.00	48.00	48.50
26	Vision	0.780	0.330	355.0	360.0	350.0	365.0
26	Prompt	0.800	0.320	510.0	520.0	500.0	515.0
26	Response	0.760	0.350	515.0	525.0	505.0	520.0
Gemma3-1B							
0	Vision	0.880	0.240	24.50	24.30	26.70	27.00
0	Prompt	0.820	0.300	23.70	23.40	23.60	24.20
0	Response	0.760	0.380	19.50	16.80	19.20	26.80
12	Vision	0.680	0.390	2.08	2.00	2.20	1.91
12	Prompt	0.720	0.360	2.10	2.12	2.08	1.92
12	Response	0.660	0.410	2.03	2.08	2.14	1.95
24	Vision	0.900	0.220	2.64	2.67	2.62	2.60
24	Prompt	0.880	0.240	2.58	2.64	2.57	2.54
24	Response	0.890	0.230	2.94	2.91	2.88	2.90

Table 20: Single-layer expert replacement with routing weights fixed. Slot and Alt. give zero-based router ranks among selected and unselected experts. Δ compares with original routing. C/W denote correct/incorrect answers. Accuracy changes and flips are in percentage points and percent, respectively.

Layer	Slot	Alt.	Mean ΔNLL	Median	ΔNLL>0 (%)	ΔAcc	C$\rightarrow$W	W$\rightarrow$C
Qwen3-1.7B								
0	0	0	+0.025	+0.008	62.0	-1.00	1.50	0.50
0	0	1	+0.040	+0.012	67.0	-1.50	2.00	0.50
0	1	0	+0.015	+0.004	60.0	-0.50	1.00	0.50
0	1	1	+0.020	+0.006	63.0	-1.00	1.50	0.50
14	0	0	+0.080	+0.035	73.0	-3.00	4.00	1.00
14	0	1	+0.110	+0.045	78.0	-4.00	5.00	1.00
14	1	0	+0.040	+0.015	66.0	-1.50	2.50	1.00
14	1	1	+0.055	+0.022	70.0	-2.00	3.00	1.00
26	0	0	+0.060	+0.027	72.0	-2.50	3.50	1.00
26	0	1	+0.090	+0.038	76.0	-3.50	4.50	1.00
26	1	0	+0.025	+0.009	64.0	-1.00	2.00	1.00
26	1	1	+0.035	+0.013	67.0	-1.50	2.50	1.00
Gemma3-1B								
0	0	0	+0.018	+0.005	59.0	-0.67	1.17	0.50
0	0	1	+0.030	+0.009	64.0	-1.00	1.50	0.50
0	1	0	+0.012	+0.003	57.0	-0.50	1.00	0.50
0	1	1	+0.017	+0.005	60.0	-0.67	1.17	0.50
12	0	0	+0.060	+0.022	69.0	-2.17	3.00	0.83
12	0	1	+0.075	+0.030	74.0	-2.67	3.50	0.83
12	1	0	+0.032	+0.010	63.0	-1.00	1.67	0.67
12	1	1	+0.045	+0.016	67.0	-1.50	2.17	0.67
24	0	0	+0.045	+0.016	67.0	-1.50	2.50	1.00
24	0	1	+0.060	+0.024	71.0	-2.00	3.00	1.00
24	1	0	+0.022	+0.006	61.0	-0.67	1.50	0.83
24	1	1	+0.030	+0.010	64.0	-1.00	1.83	0.83

Table 21: Sensitivity to initialization noise α_{init} and learning rate across model backbones (four 2:4 experts, top-2).

	α_{init}	Learning rate	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B	0.1	5×10^{-4}	62.08	1335.1	86.63	63.99	52.86
	0	5×10^{-4}	61.96	1324.5	86.34	61.48	52.11
	0.3	5×10^{-4}	62.20	1343.1	86.73	63.81	52.26
	0.1	1×10^{-4}	61.49	1287.2	85.81	65.15	51.92
	0.1	1×10^{-3}	61.89	1321.6	86.78	60.98	52.45
Gemma3-1B	0.1	5×10^{-4}	60.34	1294.4	86.35	55.40	42.81
	0	5×10^{-4}	59.74	1291.5	85.51	55.78	42.12
	0.3	5×10^{-4}	59.63	1288.2	84.87	52.75	41.40
	0.1	1×10^{-4}	59.90	1251.4	85.87	53.30	42.99
	0.1	1×10^{-3}	59.13	1222.1	84.78	52.31	35.38

Table 22: Training-gate comparison across model backbones (four 2:4 experts). Both settings use deterministic evaluation.

Training router	GQA ↑	MME ↑	POPE ↑	SQA ↑	TextVQA ↑
Qwen3-1.7B					
Argmax + Gumbel	61.97±0.27	1309.7±10.6	86.35±0.10	63.20±0.45	52.46±0.23
Deterministic top-2	62.00±0.28	1332.4±11.9	86.49±0.16	63.48±0.57	52.50±0.37
Gemma3-1B					
Argmax + Gumbel	59.20±0.22	1214.3±14.9	85.18±0.16	51.14±0.54	40.66±0.38
Deterministic top-2	**60.07±0.27**	**1293.8±11.9**	**86.41±0.10**	**55.30±0.47**	**42.90±0.47**