Soft Modality-Guided Expert Specialization in MoE-VLMs

Supplementary Material

1. Detailed Experimental Setup

1.1. Vision Encoder and Projector

All VLM configurations use the same vision encoder and projector so that backbone and routing changes are the only variables:

Vision Encoder: CLIP ViT-L/14 [44] with 336×336 input resolution, producing 576 visual tokens for every image. **Projector:** A two-layer MLP with GELU activation that maps the 1024-dimensional CLIP features to the hidden size of each MoE backbone (e.g., 2048 for DeepSeekMoE).

1.2. MoE Backbone Architectures

We provide detailed specifications for the four MoE-based VLM backbones used in our experiments. They cover different router designs (softmax vs. sigmoid), expert settings (with or without shared experts), attention variants (MHA, MLA, GQA), and model scales from 7B to 30B parameters. This variety allows us to validate SMoES under heterogeneous architectures. Tab. S2 summarizes the main differences.

DeepSeekMoE [11] employs two shared experts together with 64 routed experts under top-6 gating, providing both stable training and sufficient specialization capacity.

OLMoE [41] is an open-source lightweight (only 7B parameters in total) MoE model with 64 experts and top-8 routing. Unlike DeepSeekMoE, it does not use shared experts, relying entirely on the routing mechanism for expert selection. It is not optimized for multilingual tasks, which can be seen in the MMB-CN benchmark.

Moonlight-MoE [36] is a recent architecture that is similar to DeepSeekMoE in scale. However, it differs significantly in its router design: unlike other models that use softmax-based routing, Moonlight-MoE employs sigmoid-based routing with router score bias, which provides a different gating mechanism for expert selection. The introduction of router score bias also induces certain fluctuations to the router, for example, hard routing does not perfectly separate visual and text tokens (see MSI in Tab. S4).

Qwen3-MoE [62] is a 30B-parameter production model with 128 experts and top-8 routing. It uses GQA attention and head-wise QK norm, and serves as the strongest baseline among the four backbones.

1.3. Training Configuration

We follow the standard LLaVA [35] two-stage training protocol:

Stage 1: Feature Alignment. We train only the projector while keeping the vision encoder and language model frozen. This stage uses 558K image-caption pairs from the LLaVA-Pretrain dataset.

Stage 2: Instruction Tuning. We fine-tune the entire model (except the vision encoder) on the LLaVA-Instruct-665K dataset with diverse vision-language instructions.

Table S1. Training details.

	Stage 1	Stage 2
Optimizer	AdamW	AdamW
$\text{LR}_{\text{vision-encoder}}$	0	0
$\text{LR}_{\text{projector}}$	1e-3	2e-5
$\text{LR}_{\text{language-model}}$	0	2e-5
LR Scheduler	cosine	cosine
Warmup Ratio	0.03	0.03
Num Samples	558K	665K
Num Epochs	1	1
Batch Size	256	128
Weight Decay	0.0	0.0
Gradient Clipping	1.0	1.0

We show training details in Tab. S1. All experiments use AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 0.0. For our method-specific hyperparameters, we set $N_{\text{bins}} = 8$, $\alpha_{\text{bal}} = 0.001$, $\alpha_{\text{MI}} = 0.0001$, EMA decay $\beta = 0.99$, and Gaussian temperature $\tau = 0.5 \cdot D$ as default values across all backbones.

All experiments are conducted on 8 NVIDIA A800-80GB GPUs using mixed-precision training (BF16) and gradient checkpointing to reduce memory usage. In cases where memory constraints persist, we use gradient accumulation to achieve the effective batch sizes. During benchmark evaluation, we use greedy search without sampling to produce consistent results. We set the maximum number of new tokens in generation to 256 for the GSM8k benchmark and 20 for the others.

We do not use dynamic expert migration during training in our experiments. However, a reasonable migration frequency according to the expert specialization is encouraged when scaling. Thanks to EMA, expert groupings remain stable over a certain period, making the migration overhead reasonable.

1.4. Efficient Implementation of SMoES

Although the attention-accumulated score formulation (Sec. 3.2.1) in SMoES uses the attention matrix $\text{Attn}^{(l)}$ for clarity, the aggregation can be efficiently implemented without materializing the full attention matrix, making it compatible with memory-efficient attention kernels such

Table S2. Detailed specifications of MoE backbone architectures.

Architecture	DeepSeekMoE	OLMoE	Moonlight-MoE	Qwen3-MoE
Huggingface Checkpoint	deepseek-moe-16b-base	OLMoE-1B-7B-0125	Moonlight-16B-A3B	Qwen3-30B-A3B
Total Parameters	16B	7B	16B	30B
Activated Params	3B	1B	3B	3B
Number of Routed Experts	64	64	64	128
Top-k Routing	6	8	6	8
Number of Shared Experts	2	0	2	0
Router Type	softmax	softmax	sigmoid	softmax
Router Score Bias	no	no	yes	no
Hidden Dimension	2048	2048	2048	2048
Number of Layers	28	16	27	48
Dense Replace for First k Layers	1	0	1	0
QK Norm	no	token-wise	no	head-wise
Attention Type	MHA	MHA	MLA	GQA

as Flash Attention [12]. Specifically, we concatenate the modality scores $M_{ij,m}^{\text{attn},(l)}$ to the value vectors before applying attention, allowing the kernel to perform the weighted aggregation implicitly. After the attention operation, we extract the aggregated modality scores from the corresponding dimensions of the attention output. This implementation strategy preserves the aggregation semantics while leveraging efficient attention implementations without additional memory or computational overhead.

2. Supplementary Results

2.1. Main results on Moonlight-MoE and Qwen3-MoE

In addition to the primary evaluation on DeepSeekMoE and OLMoE reported in Tab. 1, we provide the detailed performance on Moonlight-MoE and Qwen3-MoE in Tab. S4. The results demonstrate that SMoES consistently outperforms various routing baselines across these backbones, exhibiting a performance gain trend similar to that observed on DeepSeekMoE and OLMoE. For Moonlight-MoE, we notice that the improvement in the Modality Specialization Index (MSI) is relatively moderate, increasing to 0.442 from 0.380 in the vanilla soft routing baseline. This phenomenon likely stems from the architectural characteristics and pretraining strategies of Moonlight-MoE, which appear to inherently foster a certain level of modality differentiation even without explicit routing constraints. Despite this existing bias, SMoES still effectively refines the routing mechanism to achieve a more structured expert specialization.

Beyond the multimodal performance, we observe a critical limitation in existing modality-specialized methods: they often suffer from substantial performance degradation on language-only tasks. Such a decline suggests that traditional rigid routing or aggressive specialization constraints can inadvertently compromise the model's fundamental linguistic knowledge. While the instruction-tuning data incorporates pure text samples, these baseline methods

Table S3. Ablation on the modality score type in SMoES.

Method	MSI	Multi-Modal	Language	Overall
VLM based on DeepSeekMoE (A3B/16B, top-6/64 experts)				
No Specialization	.177	100%	100%	100%
SMAR (best)	.543	+0.6%	-11.3%	-3.9%
SMoES				
hard-score	.904	-0.8%	+0.5%	-0.3%
attention-soft	.487	**+1.8%**	**+6.2%**	**+3.5%**
gaussian-soft	.440	+1.3%	+4.2%	+2.4%
VLM based on OLMoE (A1B/7B, top-8/64 experts)				
No Specialization	.205	100%	100%	100%
SMAR (best)	.485	-0.4%	-0.1%	-0.3%
SMoES				
hard-score	.756	-0.0%	+1.9%	+0.7%
attention-soft	.620	+0.5%	**+6.7%**	**+2.9%**
gaussian-soft	.754	**+0.6%**	+4.3%	+2.0%
VLM based on Moonlight-MoE (A3B/16B, top-6/64 experts)				
No Specialization	.380	100%	100%	100%
SMAR (best)	.673	-2.8%	-30.3%	-13.1%
SMoES				
hard-score	.451	+0.1%	+3.1%	+1.2%
attention-soft	.442	+0.2%	+4.5%	+1.8%
gaussian-soft	.449	**+0.7%**	**+8.0%**	**+3.5%**
VLM based on Qwen3-MoE (A3B/30B, top-8/128 experts)				
No Specialization	.284	100%	100%	100%
SMAR (best)	.553	+0.7%	-5.4%	-1.6%
SMoES				
hard-score	.890	-1.8%	-1.5%	-1.7%
attention-soft	.726	**+1.3%**	**+0.1%**	**+0.9%**
gaussian-soft	.566	+0.8%	-0.2%	+0.4%

Table S4. Multimodal and language-only results on VLMs based on Moonlight-MoE and Qwen3-MoE. **Bold** and underline indicate the first and second best performance. MSI denotes Modality Specialization Index. †t/v/s denote the number of text/vision/shared experts. *[a, b] denotes KL divergence threshold range.

Method	MSI	Multimodal Tasks (10)											Language-Only Tasks (6)							Overall
		MMMU^{val}	*MMMU^{test}*	*GQA*	*POPE*	*SQA-IMG*	*TextVQA*	*MME*	*MMB*	*MMB-CN*	*VQAv2*	**Avg**	*MMLU*	*HellaSwag*	*ARC-C*	*ARC-E*	*GSM8k*	*TruthfulQA*	**Avg**	**Avg**
VLM based on Moonlight-MoE (A3B/16B, top-6/64 experts)																				
No Specialization	.380	34.1	31.0	58.8	85.7	67.7	58.9	**1715**	67.0	63.7	77.0	100%	51.0	54.1	60.1	78.3	25.3	**51.4**	100%	100%
Hard Routing [39]																				
t32-v32†	.991	27.1	27.6	57.9	85.0	59.9	54.4	1525	59.9	59.8	76.1	-8.2%	33.1	39.5	40.1	53.7	3.4	43.1	-38.3%	-19.5%
t48-v16†	.952	31.2	29.6	58.5	85.4	64.1	58.1	1597	64.8	60.0	76.6	-3.7%	41.8	47.3	49.5	67.9	14.1	44.4	-19.9%	-9.8%
MoIIE [53]																				
t16-v16-s32†	.686	32.2	29.9	58.4	85.7	65.5	58.2	1636	63.8	61.3	76.7	-2.8%	44.1	45.6	53.2	72.9	13.1	46.3	-17.7%	-8.3%
t24-v24-s16†	.836	29.8	28.4	58.4	85.7	63.4	56.8	1615	61.7	59.8	76.5	-5.2%	35.4	39.6	39.8	53.8	4.9	42.1	-36.9%	-17.1%
t32-v16-s16†	.833	32.4	29.8	58.6	**85.7**	65.1	57.4	1583	63.5	60.6	76.6	-3.4%	43.7	47.5	52.8	70.3	12.4	40.3	-20.3%	-9.7%
SMAR [61]																				
d_{KL}*-[0.5, 1.0]**	.599	32.3	30.3	58.3	84.1	66.9	57.5	1590	65.7	61.3	76.7	-2.7%	38.9	38.1	47.0	64.3	11.7	32.6	-30.6%	-13.2%
d_{KL}*-[1.5, 2.0]**	.646	31.6	29.3	58.2	84.6	66.1	55.8	1626	64.7	60.8	76.2	-3.7%	36.8	39.1	39.9	55.4	6.9	26.6	-39.9%	-17.3%
d_{KL}*-[2.5, 3.0]**	.673	31.9	29.6	58.1	84.9	68.1	57.7	1692	63.7	59.7	76.4	-2.8%	40.0	37.2	44.6	62.8	11.1	37.6	-30.3%	-13.1%
SMoES (ours)																				
attention-soft	.442	33.7	**32.2**	58.8	85.1	67.4	58.5	1668	**68.2**	**64.7**	77.0	+0.2%	**52.9**	55.6	67.7	84.1	26.4	49.6	+4.5%	+1.8%
gaussian-soft	.449	**35.7**	32.0	**59.0**	85.2	**68.6**	**58.9**	1671	67.0	64.1	**77.0**	**+0.7%**	52.9	**56.3**	**68.9**	**85.0**	**30.8**	49.2	**+8.0%**	**+3.5%**
VLM based on Qwen3-MoE (A3B/30B, top-8/128 experts)																				
No Specialization	.284	39.8	37.4	57.3	85.7	76.5	58.6	**1830**	67.9	**71.7**	75.6	**100%**	**67.9**	82.2	87.8	94.5	66.1	60.7	100%	100%
Hard Routing [39]																				
t64-v64†	1.	37.0	31.9	56.1	85.4	66.2	55.4	1630	66.4	60.0	74.5	-7.4%	51.9	68.5	75.2	87.2	34.2	54.8	-20.1%	-12.1%
t96-v32†	1.	39.6	34.6	58.0	86.6	71.7	59.1	1741	68.7	66.5	75.1	-2.3%	60.5	77.9	81.3	91.1	53.3	58.8	-8.3%	-4.5%
MoIIE [53]																				
t32-v32-s64†	.509	39.9	34.6	57.0	85.9	70.8	57.7	1766	69.2	66.7	75.2	-2.5%	60.9	79.3	78.7	87.3	52.6	58.2	-9.4%	-5.1%
t48-v48-s32†	.754	37.8	33.5	56.9	85.6	70.7	56.8	1649	68.7	62.0	74.9	-5.0%	55.9	73.0	71.1	84.6	46.0	53.0	-16.9%	-9.5%
t64-v32-s32†	.800	38.3	34.5	57.4	85.6	72.1	58.6	1693	67.0	65.3	74.7	-3.6%	60.9	78.1	83.3	91.9	54.1	59.7	-7.2%	-5.0%
SMAR [61]																				
d_{KL}*-[0.5, 1.0]**	.747	**44.8**	38.6	56.8	**86.8**	72.9	57.9	1711	60.6	63.7	75.2	-1.9%	56.5	69.9	75.9	86.0	56.4	**63.4**	-10.8%	-5.2%
d_{KL}*-[1.5, 2.0]**	.553	43.5	**40.0**	56.3	85.5	**78.2**	58.8	1664	69.9	69.4	75.3	+0.7%	60.6	74.3	82.7	92.2	61.1	63.0	-5.4%	-1.6%
d_{KL}*-[2.5, 3.0]**	.647	41.9	38.7	56.2	86.1	77.6	59.8	1718	69.7	69.3	75.3	+0.4%	60.8	74.5	81.1	91.1	60.3	59.7	-6.9%	-2.4%
SMoES (ours)																				
attention-soft	.726	40.5	38.9	58.2	85.5	75.5	**60.8**	1775	**72.7**	71.0	75.9	**+1.3%**	67.7	**83.0**	88.0	96.1	**66.2**	59.5	**+0.1%**	**+0.9%**
gaussian-soft	.566	41.7	37.9	**58.9**	85.8	76.0	59.2	1760	71.5	69.3	**76.1**	+0.8%	67.5	82.7	**88.4**	**96.7**	65.2	59.0	-0.2%	+0.4%

tend to erode the model's ability to handle complex linguistic tasks, particularly those requiring multi-step reasoning like GSM8k. In the broader context of VLM development, language capabilities are frequently undervalued; however, they remain indispensable for ensuring a high-quality user experience and enabling sophisticated long-context reasoning. Our results underscore that SMoES successfully maintains robust language performance while enhancing multimodal synergy, which is vital for building truly unified and versatile multimodal models.

2.2. Gaussian Estimator

In our primary methodology, we employ a unimodal Gaussian Estimator with a diagonal covariance matrix, primarily to ensure computational efficiency and numerical stability in high-dimensional feature spaces. However, such a simplified distribution may be overly restrictive, potentially failing to capture the intricate, non-linear dependencies and multi-modal structures inherent in fused modality representations. To investigate the impact of estimator expressiveness, we conduct preliminary experiments exploring more sophisticated density modeling techniques. While adopting a full covariance matrix could theoretically cap-

Benchmark	Baseline	k=1	k=2	k=4
multimodal	100%	+0.6%	**+1.2%**	+0.8%
language	100%	+4.3%	**+5.6%**	+4.3%
Overall	100%	+2.0%	**+2.8%**	+2.1%

Table S5. Gaussian Mixture Model (GMM) estimator with different n-components on OLMoE.

Table S6. Ablation on inter-bin specialization objectives. KL: Kullback-Leibler divergence. MI: Mutual Information.

Method	MSI	Multi-Modal	Language	Overall
VLM based on DeepSeekMoE (A3B/16B, top-6/64 experts)				
No Specialization	.177	100%	100%	100%
w/ binning	.415	+0.9%	+3.0%	+1.7%
w/ Inter-bin				
KL	.724	-1.5%	-8.5%	-4.1%
MI-attention	.487	**+1.8%**	**+6.2%**	**+3.5%**
MI-gaussian	.440	+1.3%	+4.2%	+2.4%
VLM based on OLMoE (A1B/7B, top-8/64 experts)				
No Specialization	.205	100%	100%	100%
w/ binning	.324	-12.2%	-9.6%	-11.2%
w/ Inter-bin				
KL	.545	+0.4%	-0.9%	-0.1%
MI-attention	.620	+0.5%	**+6.7%**	**+2.9%**
MI-gaussian	.754	**+0.6%**	+4.3%	+2.0%
VLM based on Moonlight-MoE (A3B/16B, top-6/64 experts)				
No Specialization	.380	100%	100%	100%
w/ binning	.441	-0.0%	-1.5%	-0.6%
w/ Inter-bin				
KL	.748	-5.9%	-8.1%	-6.7%
MI-attention	.442	+0.2%	+4.5%	+1.8%
MI-gaussian	.449	**+0.7%**	**+8.0%**	**+3.5%**
VLM based on Qwen3-MoE (A3B/30B, top-8/128 experts)				
No Specialization	.284	100%	100%	100%
w/ binning	.468	+0.6%	-0.1%	+0.3%
w/ Inter-bin				
KL	.540	-2.0%	**+1.0%**	-1.0%
MI-attention	.726	**+1.3%**	+0.1%	+0.9%
MI-gaussian	.566	+0.8%	-0.2%	+0.4%

Table S7. Ablation on expert binning strategy.

Method	MSI	Multi-Modal	Language	Overall
VLM based on DeepSeekMoE (A3B/16B, top-6/64 experts)				
No Specialization	.177	100%	100%	100%
w/ binning				
fixed	.357	+0.9%	+2.9%	+1.6%
adaptive	.415	+0.9%	+3.0%	+1.7%
attention-soft				
fixed	.450	+2.0%	+0.2%	+1.3%
adaptive	.487	+1.8%	+6.2%	+3.5%
gaussian-soft				
fixed	.398	+1.9%	-1.0%	+0.8%
adaptive	.440	+1.3%	+4.2%	+2.4%
VLM based on OLMoE (A1B/7B, top-8/64 experts)				
No Specialization	.205	100%	100%	100%
w/ binning				
fixed	.324	-18.6%	-10.5%	-15.6%
adaptive	.324	-12.2%	-9.6%	-11.2%
attention-soft				
fixed	.355	+0.1%	+3.3%	+1.4%
adaptive	.620	+0.5%	+6.7%	+2.9%
gaussian-soft				
fixed	.433	-0.6%	+1.8%	+0.5%
adaptive	.754	+0.6%	+4.3%	+2.0%
VLM based on Moonlight-MoE (A3B/16B, top-6/64 experts)				
No Specialization	.380	100%	100%	100%
w/ binning				
fixed	.445	-9.8%	-37.3%	-20.2%
adaptive	.441	-0.0%	-1.5%	-0.6%
attention-soft				
fixed	.377	+0.1%	-0.6%	-0.2%
adaptive	.442	+0.2%	+4.5%	+1.8%
gaussian-soft				
fixed	.388	+0.3%	+5.0%	+2.1%
adaptive	.449	+0.7%	+8.0%	+3.5%
VLM based on Qwen3-MoE (A3B/30B, top-8/128 experts)				
No Specialization	.284	100%	100%	100%
w/ binning				
fixed	.308	-0.1%	+0.6%	+0.1%
adaptive	.468	+0.6%	-0.1%	+0.3%
attention-soft				
fixed	.325	+1.3%	+1.8%	+1.5%
adaptive	.726	+1.3%	+0.1%	+0.9%
gaussian-soft				
fixed	.332	+0.6%	+0.0%	+0.4%
adaptive	.566	+0.8%	-0.2%	+0.4%

ture cross-dimension correlations, it introduces $O(D^2)$ parameters and incurs significant overhead for online updates and likelihood computations. As a more practical alternative, we evaluate the Gaussian Mixture Model (GMM) with $k \in \{2, 4\}$ components, as detailed in Tab. S5. Our results indicate that $k = 2$ yields measurable performance gains, which aligns with our qualitative observations in Figs. S10 to S12 where certain layers exhibit distinct bi-modal or multi-peak token distributions. These findings suggest that the occasional performance gap between the Gaussian and attention-based estimators likely arises from the limited capacity of a simple Gaussian model to characterize highly complex or heterogeneous token samples.

2.3. Ablation study

Ablation on the modality score type on all four MoE backbones is shown in Tab. S3. Our attention-soft and gaussian-soft scores outperform the hard routing score (0/1 hard-coded) on all four MoE backbones. This demonstrates the importance of soft modality scores in modality differentiation. Furthermore, even the hard modality score can outperform SMAR, indicating that MI-based modality differentiation is more effective than KL-based approaches.

Ablation on inter-bin specialization objective is shown in Tab. S6. Expert binning can increase MSI since it provides the feasibility for modality differentiation. Inter-bin MI-based specialization further enhances modality differen-

Table S8. Ablation on Gaussian temperature. D: number of the feature dimension.

Method	MSI	Multi-Modal	Language	Overall
VLM based on DeepSeekMoE (A3B/16B, top-6/64 experts)				
No Specialization	.177	100%	100%	100%
gaussian-τ				
$0.05D$	.516	-0.9%	**+4.8%**	+1.2%
$0.1D$	.574	+0.6%	+4.2%	+1.9%
$0.3D$	.564	**+1.6%**	+1.4%	+1.5%
$0.5D$ *(default)*	.440	+1.3%	+4.2%	**+2.4%**
$0.7D$	.444	+1.4%	+3.1%	+2.1%
$1.0D$	.475	+0.2%	+2.4%	+1.1%
VLM based on OLMoE (A1B/7B, top-8/64 experts)				
No Specialization	.205	100%	100%	100%
gaussian-τ				
$0.05D$	.744	-0.4%	+0.6%	-0.0%
$0.1D$	.744	-0.1%	**+4.3%**	+1.7%
$0.3D$	.759	-0.4%	+4.3%	+1.4%
$0.5D$ *(default)*	.754	**+0.6%**	+4.3%	**+2.0%**
$0.7D$	.745	+0.2%	+3.9%	+1.6%
$1.0D$	.740	+0.5%	-2.0%	-0.5%

tiation and improves model performance. In contrast, KL-based inter-bin modality differentiation, despite achieving higher modality differentiation degrees on DeepSeekMoE and Moonlight-MoE, degrades model performance, which is unacceptable.

Ablation on expert binning strategy is shown in Tab. S7. We compare our momentum-adaptive binning strategy (adaptive) to the static, predefined binning (fixed) strategy. In all cases, adaptive binning is superior to fixed binning in terms of model performance. Furthermore, in most cases, adaptive binning yields a higher degree of modality specialization. This is because the pretrained MoE backbone already exhibits certain expert capability tendencies, and forcing predefined modality bins for experts would interfere with their capability expression. Furthermore, during training, experts may gradually differentiate, and fixed binning may constrain the experts' modality differentiation capacity.

Ablation on Gaussian temperature in gaussian-soft SMoES is shown in Tab. S8. We evaluate $\tau \in \{0.05D, 0.1D, 0.3D, 0.5D, 1.0D\}$, where D is the feature dimension. As shown in the table, $0.5D$ provides the best balance between confident and smooth modality scores. The table shows that both excessively high and low temperatures lead to performance degradation. Moderate temperature levels yield similar and stable performance improvements. Therefore, we adopt a moderate value of $0.5D$ as the default setting.

Ablation on modality specialization loss weight (α_{MI}) and load balance loss weight (α_{bal}) in attention-soft SMoES is shown in Tab. S9. We evaluate α_{MI} and α_{bal} in $\{1e-1, 1e-2, 1e-3, 1e-4, 1e-5\}$. The table shows that when α_{MI} is relatively large, it is possible to achieve substantial model performance improvements, but it may also cause training instability. Therefore, we choose a relatively low α_{MI} ($1e-4$) to obtain stable performance improvements. For α_{bal}, the results may vary across different models. For example, on DeepSeekMoE, using a smaller balance loss can further improve model performance, while on OLMoE, a moderate balance loss is preferred. To achieve stable performance and ensure the magnitude is consistent with the balance loss of the soft routing baseline, we choose $1e-3$ as the weight for the balance loss.

Table S9. Ablation on loss alpha ($\text{SMoES}_{\text{attention-soft}}$).

Method	MSI	Multi-Modal	Language	Overall
VLM based on DeepSeekMoE (A3B/16B, top-6/64 experts)				
No Specialization	.177	100%	100%	100%
α_{MI}				
1e-1	.533	**+3.0%**	**+16.1%**	**+8.6%**
1e-2	.716	-11.9%	+1.0%	-6.3%
1e-3	.610	-1.0%	+4.9%	+1.5%
1e-4 (default)	.487	+1.8%	+6.2%	+3.5%
1e-5	.417	-0.0%	+4.0%	+1.7%
α_{bal}				
1e-1	.290	-43.2%	-48.4%	-45.3%
1e-2	.410	-7.1%	-17.8%	-11.3%
1e-3 (default)	.487	+1.8%	+6.2%	+3.5%
1e-4	.578	**+3.9%**	**+8.1%**	**+5.2%**
1e-5	.550	+3.9%	+6.0%	+4.5%
VLM based on OLMoE (A1B/7B, top-8/64 experts)				
No Specialization	.205	100%	100%	100%
α_{MI}				
1e-1	.684	+1.8%	+4.6%	+2.8%
1e-2	.871	**+2.0%**	**+8.3%**	**+4.3%**
1e-3	.982	-0.2%	+2.2%	+0.7%
1e-4 (default)	.620	+0.5%	+6.7%	+2.9%
1e-5	.340	+0.3%	+3.9%	+1.7%
α_{bal}				
1e-1	.738	-15.4%	-27.9%	-20.1%
1e-2	.796	-0.3%	-1.8%	-0.9%
1e-3 (default)	.620	**+0.5%**	**+6.7%**	**+2.9%**
1e-4	.654	+0.4%	+4.9%	+2.1%
1e-5	.702	+0.3%	+6.0%	+2.6%

2.4. Efficiency Analysis

To evaluate the practical advantages of our SMoES method in real-world scenarios, we conduct edge-side inference experiments comparing our SMoES against baseline models without modality specialization. These experiments demonstrate the communication efficiency gains achieved through expert specialization, particularly in resource-constrained edge deployment settings.

We deploy our experiments on a dual-NVIDIA-Orin GPU setup, which represents a typical edge-side scenario

Table S10. Cross-GPU EP transfer ratio at prefill and decode stages for OLMoE. V: Vision tokens; T: Text tokens. *(PV : PT : DT)*: token count ratio for prefill-vision, prefill-text and decode-text. DeDup: de-duplication of tokens routed to experts on the same device in top-K routing.

Method	Top-K w/o DeDup				Top-K w/ DeDup			
	Prefill (P)			**Decode (D)**	**Prefill (P)**			**Decode (D)**
	V	**T**	**V+T**	**T**	**V**	**T**	**V+T**	**T**
MMMU (PV : PT : DT = 79% : 19% : 2%)								
Baseline (No Specialization)	42.7%	70.4%	48.0%	37.4%	97.7%	99.5%	98.0%	86.5%
$\text{SMoES}_{\text{attention-soft}}$	8.8% ↓ 79.4%	71.7% ↑ 1.0%	20.7% ↓ 56.9%	34.3% ↓ 8.3%	33.6% ↓ 65.6%	98.9% ↓ 0.6%	46.1% ↓ 53.0%	88.9% ↑ 2.8%
$\text{SMoES}_{\text{gaussian-soft}}$	3.3% ↓ 92.3%	76.7% ↑ 8.9%	17.3% ↓ 63.9%	16.0% ↓ 57.3%	15.0% ↓ 84.6%	99.3% ↓ 0.3%	31.1% ↓ 68.3%	43.3% ↓ 49.9%
SQA-IMG (PV : PT : DT = 65% : 30% : 5%)								
Baseline (No Specialization)	42.5%	74.1%	52.6%	49.0%	97.5%	99.7%	98.2%	94.9%
$\text{SMoES}_{\text{attention-soft}}$	8.4% ↓80.3%	76.3% ↑ 2.9%	30.1% ↓ 42.9%	41.7% ↓15.0%	33.2% ↓ 65.9%	99.3% ↓ 0.4%	54.3% ↓ 44.7%	90.0% ↓ 5.2%
$\text{SMoES}_{\text{gaussian-soft}}$	2.7% ↓ 93.6%	80.3% ↑ 8.2%	27.5% ↓ 47.7%	17.7% ↓ 63.9%	13.0% ↓ 86.6%	99.2% ↓ 0.5%	40.6% ↓ 58.7%	49.9% ↓ 47.4%
POPE (PV : PT : DT = 86% : 11% : 3%)								
Baseline (No Specialization)	42.9%	85.7%	47.8%	37.7%	97.8%	100%	98.1%	87.8%
$\text{SMoES}_{\text{attention-soft}}$	8.0% ↓ 81.4%	88.2% ↑ 2.9%	17.1% ↓ 64.1%	31.4% ↓ 16.7%	33.4% ↓ 65.9%	99.9% ↓ 0.1%	41.0% ↓ 58.2%	84.4% ↓ 3.8%
$\text{SMoES}_{\text{gaussian-soft}}$	1.8% ↓ 95.8%	91.5% ↑ 6.8%	12.0% ↓ 74.8%	14.5% ↓ 61.5%	9.0% ↓ 90.8%	99.8% ↓ 0.2%	19.4% ↓ 80.2%	40.8% ↓ 53.6%
GQA (PV : PT : DT = 86% : 11% : 3%)								
Baseline (No Specialization)	43.0%	85.5%	47.9%	38.3%	97.9%	100%	98.1%	89.2%
$\text{SMoES}_{\text{attention-soft}}$	8.2% ↓ 80.9%	87.7% ↑ 2.6%	17.4% ↓ 63.6%	32.1% ↓ 16.2%	34.0% ↓ 65.2%	99.9% ↓ 0.1%	41.7% ↓ 57.5%	84.8% ↓ 5.0%
$\text{SMoES}_{\text{gaussian-soft}}$	1.9% ↓ 95.7%	90.9% ↑ 6.4%	12.2% ↓ 74.5%	16.2% ↓ 57.7%	9.3% ↓ 90.5%	99.8% ↓ 0.2%	19.8% ↓ 79.8%	42.4% ↓ 52.5%
TextVQA (PV : PT : DT = 80% : 17% : 3%)								
baseline	43.6%	72.5%	48.6%	43.1%	97.9%	99.1%	98.1%	90.7%
$\text{SMoES}_{\text{attention-soft}}$	8.9% ↓ 79.5%	72.0% ↓ 0.8%	19.9% ↓ 59.0%	34.5% ↓ 19.9%	35.8% ↓ 63.5%	97.1% ↓ 2.1%	46.5% ↓ 52.6%	82.6% ↓ 8.9%
$\text{SMoES}_{\text{gaussian-soft}}$	2.6% ↓ 93.9%	77.2% ↑ 6.4%	15.6% ↓ 67.8%	21.0% ↓ 51.1%	12.6% ↓ 87.1%	97.5% ↓ 1.6%	27.4% ↓ 72.0%	47.1% ↓ 48.0%

for automotive applications. Given the high efficiency requirements for edge-side deployment, we test with a small-scale OLMoE-based VLM model with 7B parameters. Two NVIDIA Orin GPUs are connected via 10Gb Ethernet and deployed with Expert Parallelism (EP). Memory capacity is one of the bottlenecks for edge-side resources. Compared to Tensor Parallel (TP) and Data Parallel (DP) strategies, this approach has no redundancy in either weights or KV Cache. Compared to Attention-FFN separation with the same memory resource redundancy, this deployment has lower token transfer time, and memory space usage is relatively balanced. For the baseline deployment, since the experts are balanced, asynchronous transmission cannot give performance benefits and may even reduce efficiency, so synchronous transmission is adopted. For our SMoES with modality specialization, since there are more local experts with longer computation time, we adopted asynchronous transmission, allowing network transmission and expert computation to execute in parallel. Additionally, the optimized deployment is similar to PD separation, where

Table S11. TTFT (Time to First Token, Prefill) and TPOT (Time Per Output Token, Decode) speed improvement of $\text{SMoES}_{\text{attention-soft}}$ compared to soft-routing baseline. Δ: speedup percentage.

Benchmark	Method	Batch Size=1		Batch Size=2		Batch Size=4		Batch Size=8		Batch Size=16		Batch Size=32	
		TTFT(s)	TPOT(s)	TTFT(s)	TPOT(s)	TTFT(s)	TPOT(s)	TTFT(s)	TPOT(s)	TTFT(s)	TPOT(s)	TTFT(s)	TPOT(s)
MMMU	baseline	2.810	0.786	3.819	0.853	5.100	0.981	7.949	1.414	11.562	2.078	17.626	2.552
	SMoES	**2.519**	**0.703**	**3.303**	**0.767**	**4.302**	**0.890**	**6.203**	**1.287**	**8.804**	**1.899**	**13.638**	**2.309**
	Δ	↓10.3%	↓10.5%	↓13.5%	↓10.1%	↓15.7%	↓9.3%	↓22.0%	↓9.0%	↓23.9%	↓8.6%	↓22.6%	↓9.6%
SQA-IMG	baseline	1.493	0.766	2.940	0.858	4.217	0.951	5.824	1.278	9.759	2.100	15.036	2.721
	SMoES	**1.356**	**0.692**	**2.558**	**0.775**	**3.656**	**0.858**	**4.859**	**1.134**	**7.571**	**1.885**	**12.023**	**2.431**
	Δ	↓9.2%	↓9.7%	↓13.0%	↓9.7%	↓13.3%	↓9.8%	↓16.6%	↓11.3%	↓22.4%	↓10.3%	↓20.0%	↓10.6%

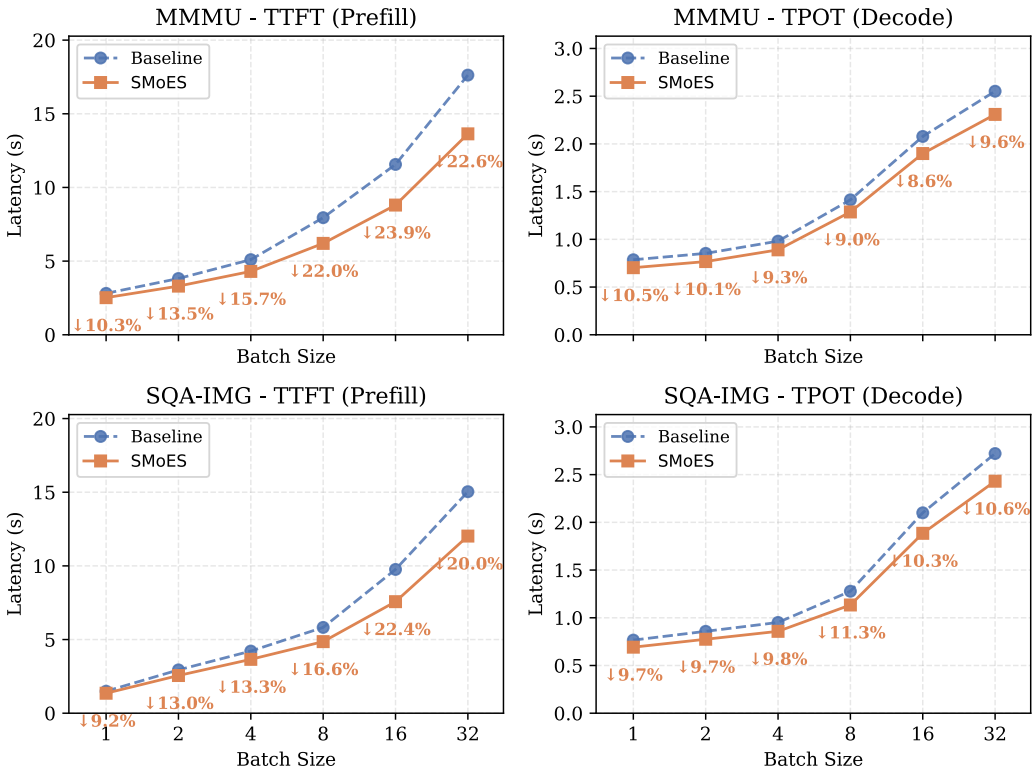

Figure S1. Latency decrease of SMoES compared to soft routing baseline at different batch sizes on edge-side deployment. The deployed VLM is based on OLMoE with 7B parameters. TTFT: Time to First Token (prefill stage); TPOT: Time Per Output Token (decode stage).

KV Cache transmission can be transferred asynchronously and be overlapped. The baseline uses a sequential order to assign experts, which is equivalent to a random order since there are no order constraints during training.

Based on the previous discussion, we can see that experts exhibit significant specialization. During actual deployment and data transmission, the data goes through DeDup (deduplication) to avoid transmitting duplicate tokens routed to experts on the same device in top-K routing. As shown in Tab. S10, we report the cross-GPU EP transfer ratio for vision and text tokens separately, since the number of vision and text tokens differs between prefill and decode phases. We also provide the overall proportion of transmitted tokens (V+T) at prefill stage. The table compares the transfer ratios with and without DeDup, demonstrating that our specialized methods (SMoES) significantly reduce the communication overhead for most of the cases, especially for vision tokens during the prefill stage. The conclusion is consistent among different benchmark datasets, indicating that our specialized methods (SMoES) are effective in reducing the communication overhead for most of the vision-language tasks.

The inference TTFT (Time to First Token) (Prefill) time and TPOT (Time Per Output Token) (Decode) time are reported in Tab. S11 and visualized in Fig. S1. The performance improvement mainly comes from the reduction in

communication time and the parallel execution of expert computation and communication, as evidenced by the latency breakdown in Fig. S1. For Prefill, as the batch size increases, the proportion of communication time increases, resulting in significant optimization performance, which aligns with the widening gaps observed in both Tab. S11 and Fig. S1. For Decode, when the Batch Size is small, due to the limited number of tokens, fewer experts are activated. Since the activated experts are proportional to the network transmission volume to a certain extent, the proportion of network transmission time remains constant, and thus the improvement ratio also remains constant.

2.4.1. Larger-Scale EP

Edge-side VLM/VLA deployment is crucial for autonomous driving and robotics. Scaling SMoES to cloud environments presents complex optimization challenges—where scheduling must account for multidimensional variables such as query modality ratios, expert specialization/redundancy, and scale-up/out bandwidth—yet it also unlocks greater efficiency potential. With proper optimization, SMoES's specialized expert groups allow converting costly cross-node communication into local computation, significantly reducing overhead compared to modality-agnostic baselines. While fully solving this complex scheduling is non-trivial, we evaluate a 16-GPU (2-node) scenario using a simple modality-based arrangement to reveal the substantial potential of SMoES. Without deep optimization, SMoES still reduces the inference time by 5.6% compared to the baseline (Tab. S12).

Batch size	1	2	4	8	16	32
Baseline	4.3	7.2	13.2	24.1	47.3	93.4
SMoES	4.1	6.7	12.1	23.1	45.4	88.2
Δ	-4.7%	-6.9%	-8.3%	-4.2%	-4.0%	-5.6%

Table S12. Inference time (s) of Qwen3-MoE (16 GPUs on 2 nodes).

2.4.2. Training Overhead

The training overhead is shown in Tab. S13. SMoES does not affect model parameters, except that the Gaussian Estimator adds a small buffer of approximately 60M. TFLOPS and peak GPU memory usage are almost unchanged. SMoES introduces a slight increase in training time, with the Gaussian Estimator contributing more due to global feature synchronization. But it becomes less noticeable as the model scales up. Note that SMoES introduces no additional overhead during inference.

2.5. Visualizations

We show routing distributions for SMoES based on the four backbones in Figs. S2 to S4. We can see that at most

Model	Params (B)	TFLOPS	Step Time (s)	Mem. (GB)
32 experts	1.19	1.82	0.74	22.3
+inter-bin MI-loss	1.19	1.82	0.80	22.3
+Attention Estimator	1.19	1.82	0.81	22.3
+Gaussian Estimator	1.25	1.83	0.93	22.3
64 experts	1.59	1.90	0.82	31.4
+inter-bin MI-loss	1.59	1.90	0.90	31.4
+Attention Estimator	1.59	1.90	0.91	31.4
+Gaussian Estimator	1.65	1.91	0.99	31.4

Table S13. Training costs on OLMoE for one GPU (batch 2, EP.8).

layers, there is one or several bins that are dedicated to vision tokens. In vision-language tasks, vision tokens typically dominate in quantity but have lower information density. By allocating dedicated experts to handle these abundant yet information-sparse vision tokens, other experts are freed up to process more complex text information. This is why SMoES achieves significantly better performance on text-only tasks compared to other methods.

However, vision tokens are not always low in information density; some vision tokens carry critical information. Fortunately, our modality specialization method is soft and dynamic, allowing the model to learn to route different vision tokens to different experts. This enables important vision tokens to be assigned to specialized experts for processing, preventing information loss. This is why our method outperforms manually pre-defined hard-routed and hybrid-routed approaches.

Except for Moonlight-MoE, which shows a relatively flat MSI layer-wise curve, most models exhibit the characteristic that shallow layers have higher modality specialization (MSI) while deeper layers show lower specialization, indicating a gradual fusion of modalities as the network goes deeper. For Moonlight-MoE, the additional router score bias makes its routing distribution less uniform, which slightly reduces its overall modality specialization level compared to other backbones.

We show the evolution of expert specialization during training on the four backbones in Figs. S5 to S9. At the early stage of training, the baseline and SMoES exhibit similar expert tendencies, with specialization scores roughly evenly spread across the spectrum. However, as training progresses, baseline experts tend to collapse toward the middle, meaning most experts become mixed-modal and handle both vision and text tokens simultaneously. In contrast, SMoES maintains pronounced modality specialization: one subset of experts separates to process vision-dominant tokens, another focuses on text-dominant tokens, and the remaining experts stay mixed to handle multimodal inputs. This specialized pattern is especially evident in shallow layers where vision and text tokens remain far apart in representation space; deeper layers, after repeated cross-modal

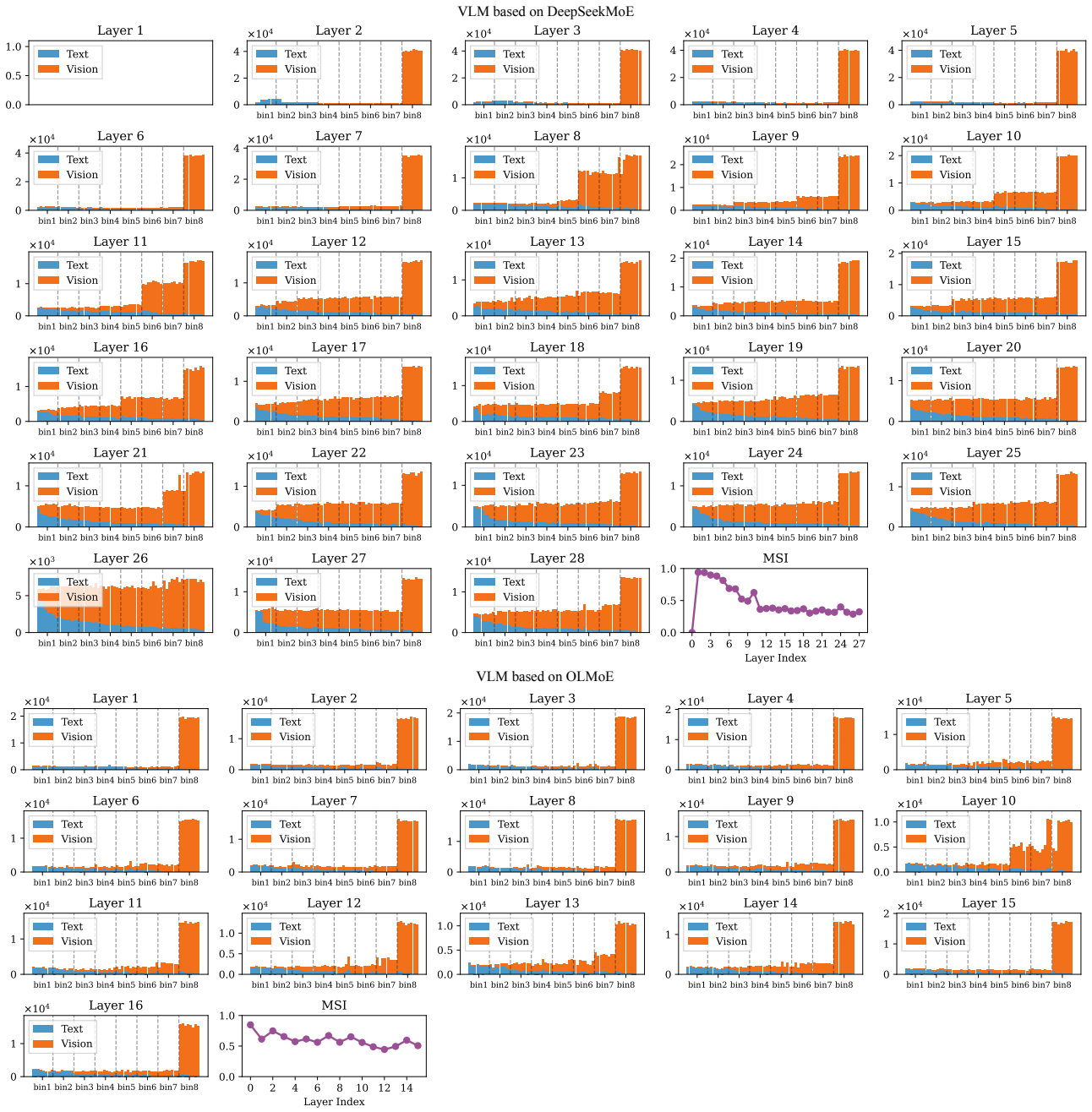

Figure S2. Routing distribution of tokens to experts in DeepSeekMoE and OLMoE. Horizontal axis: all experts grouped into eight expert bins. Vertical axis: number of tokens routed to each expert.

interactions, naturally become more multimodal, which explains the reduced but still observable specialization gap.

We show the modality fusion patterns for SMoES$_{\text{attention-soft}}$ in the four backbones in Figs. S10 to S12. Across models, shallow layers tend to produce soft modality scores near the two extremes (pure vision or pure text), and the scores gradually converge toward the center as depth increases. The convergence rate and shape vary with the backbone: DeepSeekMoE and Qwen3-MoE often split vision tokens into multiple peaks, Moonlight-MoE keeps a single peak. These differences indicate that backbone architecture and pretraining induce distinct token characteristics, while also showing that SMoES adapts to diverse MoE designs.

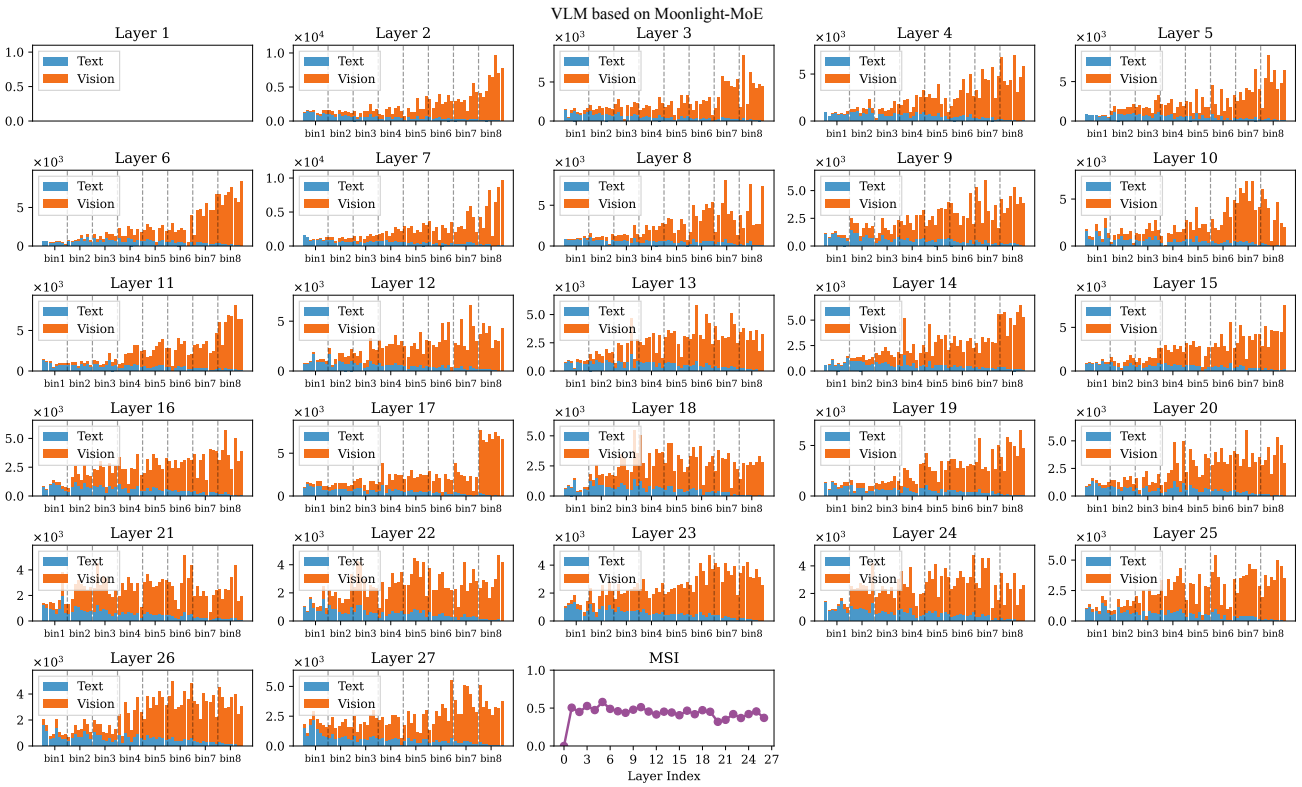

Figure S3. Routing distribution of tokens to experts in Moonlight-MoE. Horizontal axis: all experts grouped into eight expert bins. Vertical axis: number of tokens routed to each expert.

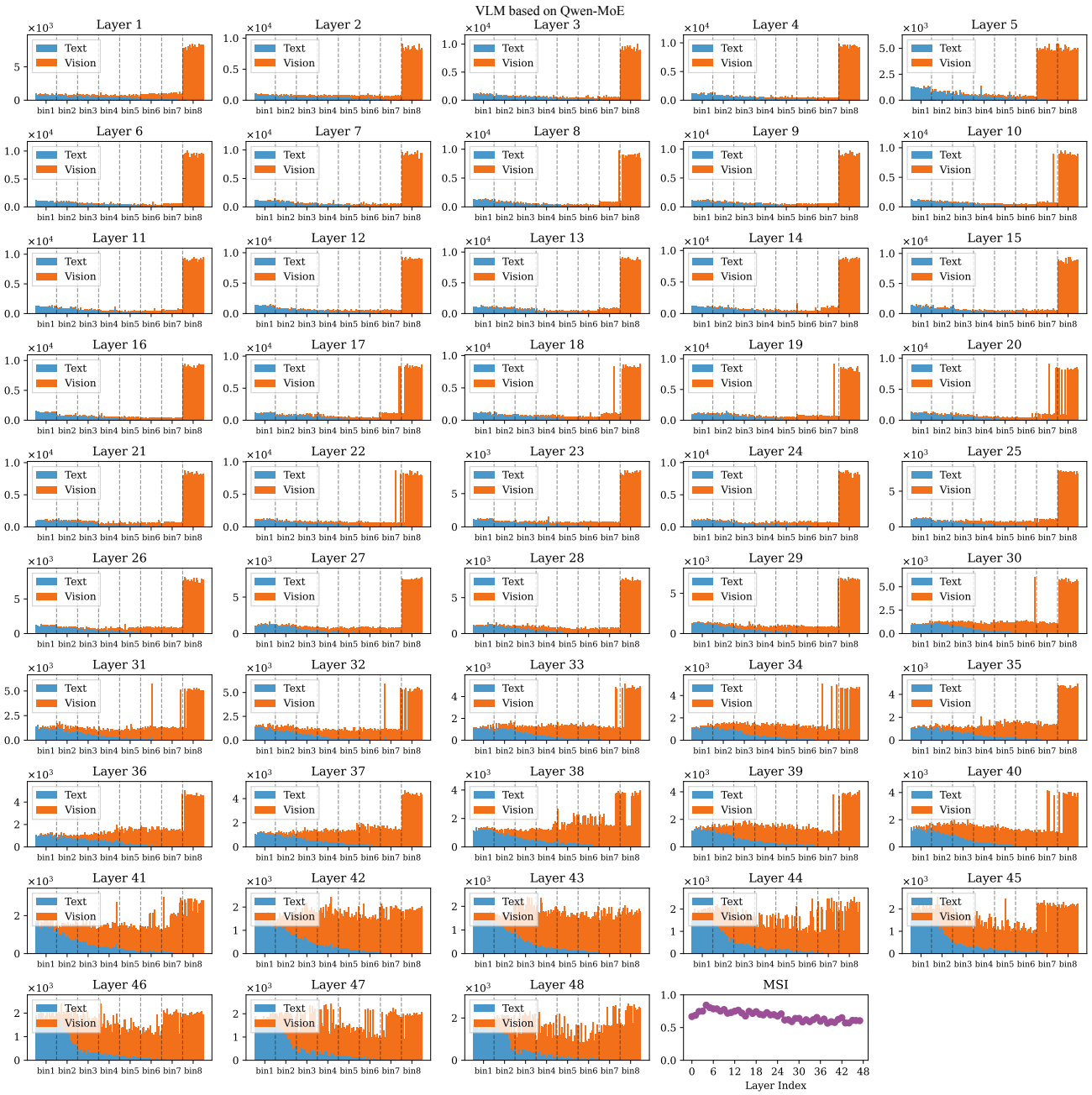

Figure S4. Routing distribution of tokens to experts in Qwen3-MoE. Horizontal axis: all experts grouped into eight expert bins. Vertical axis: number of tokens routed to each expert.

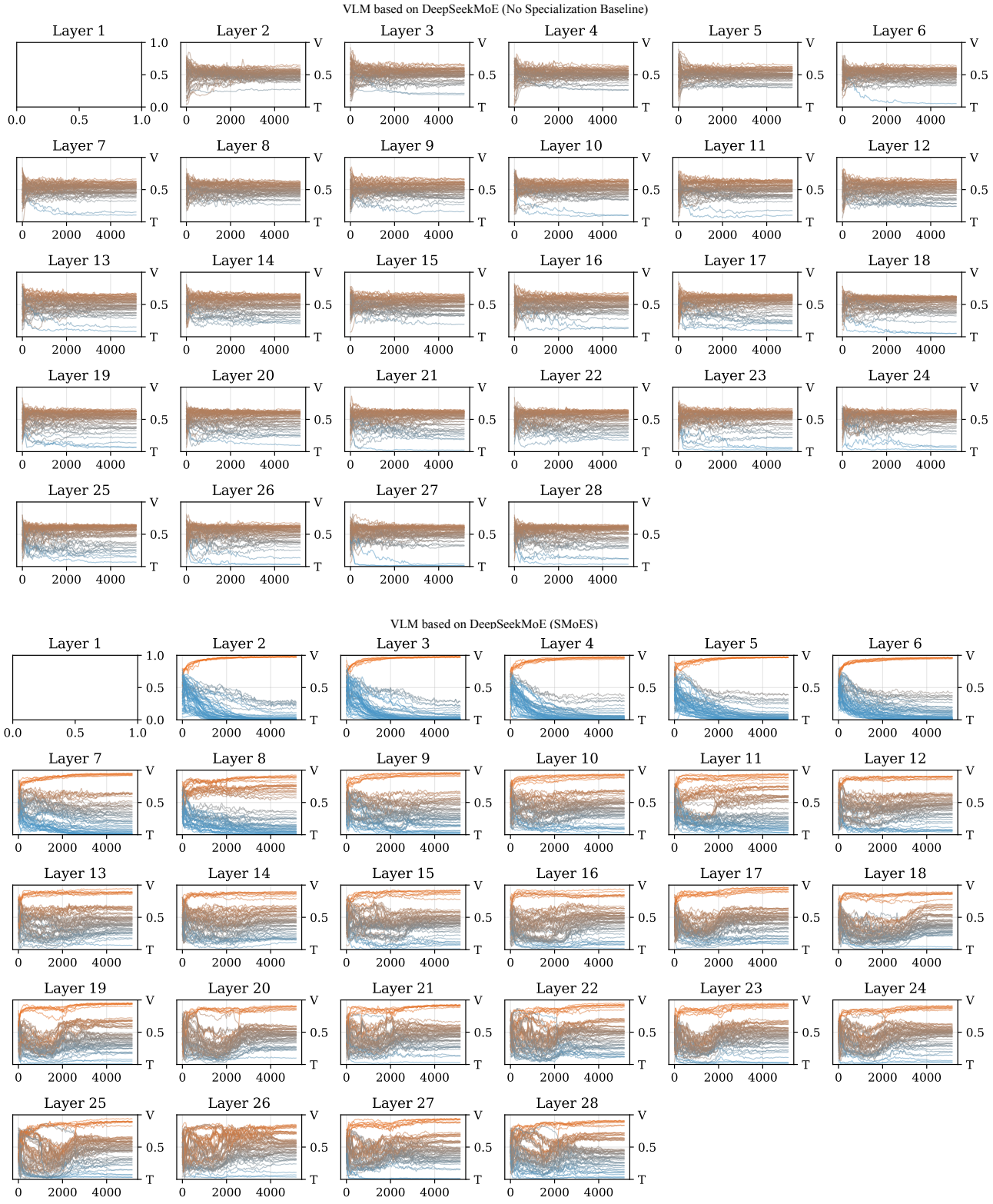

Figure S5. Evolution of expert specialization during training on DeepSeekMoE. Each curve represents an expert. Horizontal axis: training steps. Vertical axis: expert specialization score (symmetric expansion of MSI). V: vision specialization; T: text specialization.

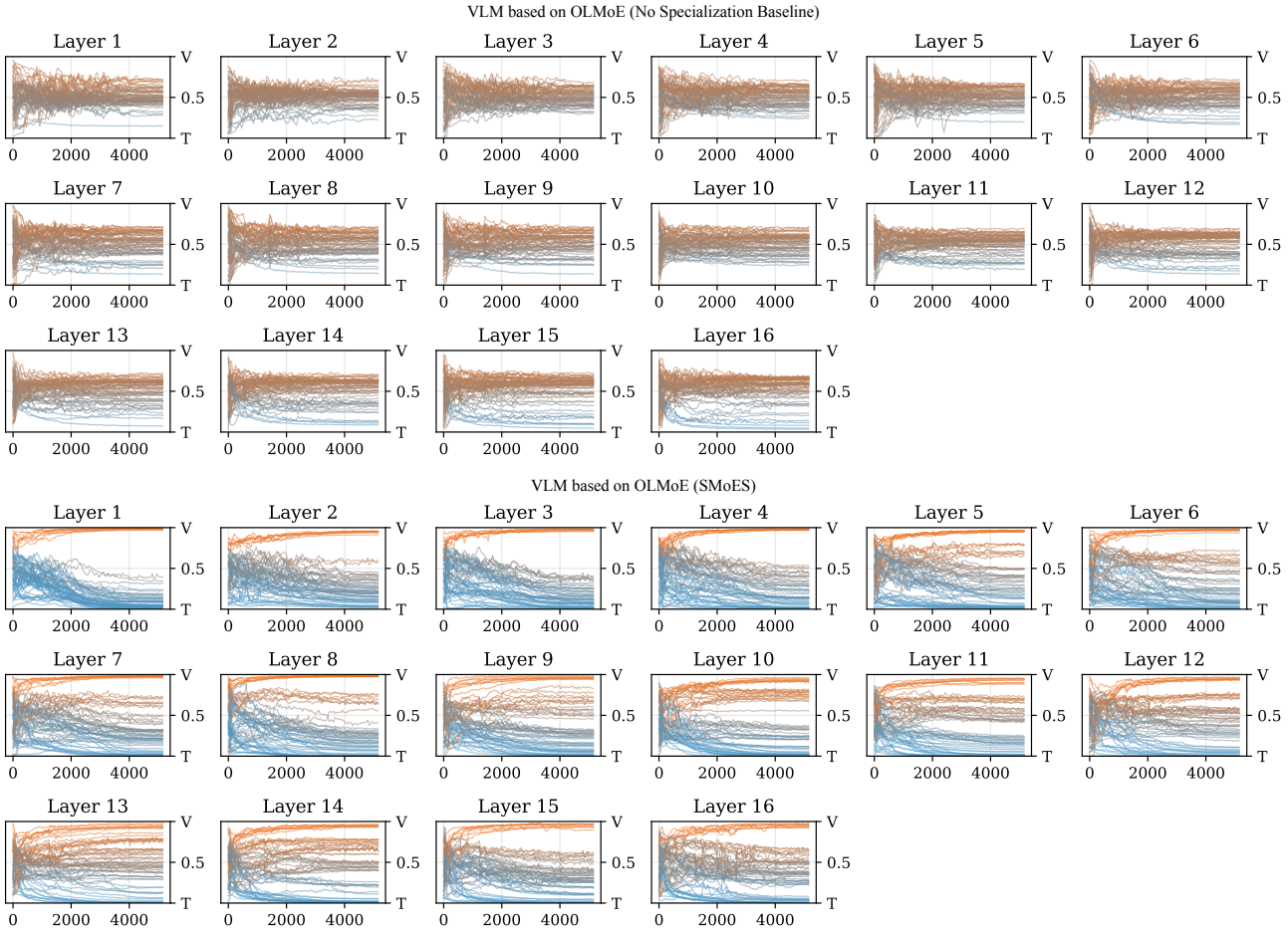

Figure S6. Evolution of expert specialization during training on OLMoE. Each curve represents an expert. Horizontal axis: training steps. Vertical axis: expert specialization score (symmetric expansion of MSI). V: vision specialization; T: text specialization.

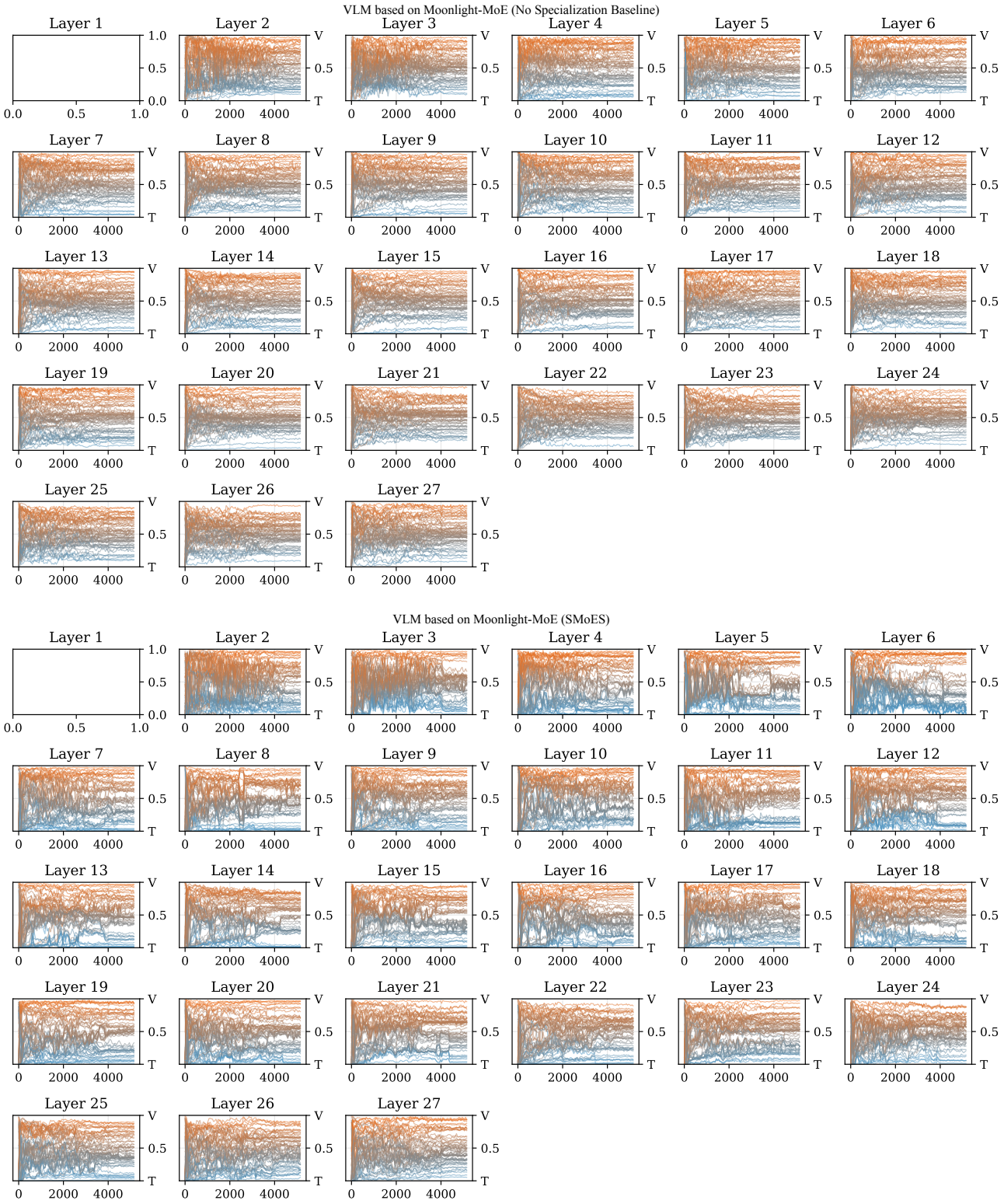

Figure S7. Evolution of expert specialization during training on Moonlight-MoE. Each curve represents an expert. Horizontal axis: training steps. Vertical axis: expert specialization score (symmetric expansion of MSI). V: vision specialization; T: text specialization.

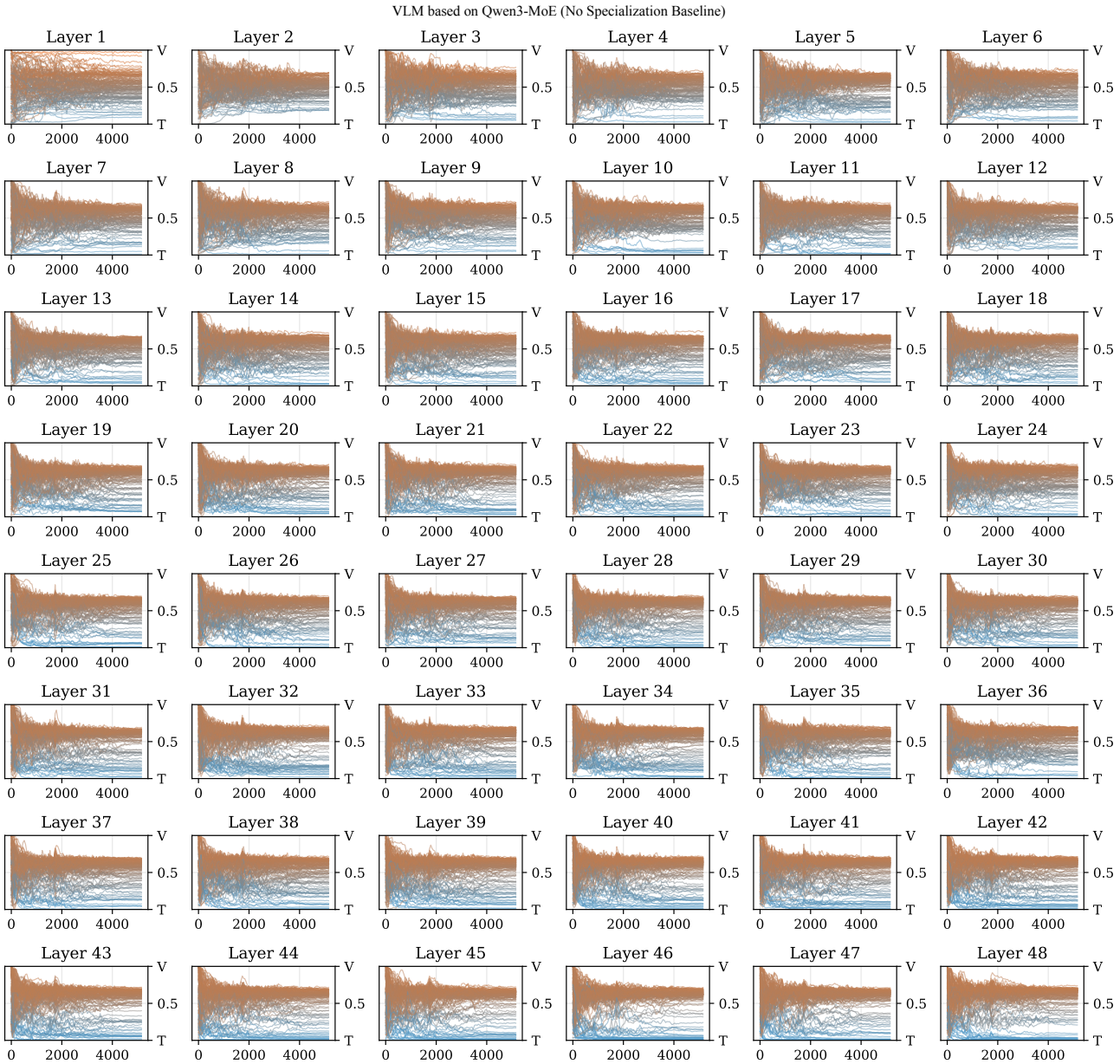

Figure S8. Evolution of expert specialization during training on Qwen3-MoE with no specialization (baseline). Each curve represents an expert. Horizontal axis: training steps. Vertical axis: expert specialization score (symmetric expansion of MSI). V: vision specialization; T: text specialization.

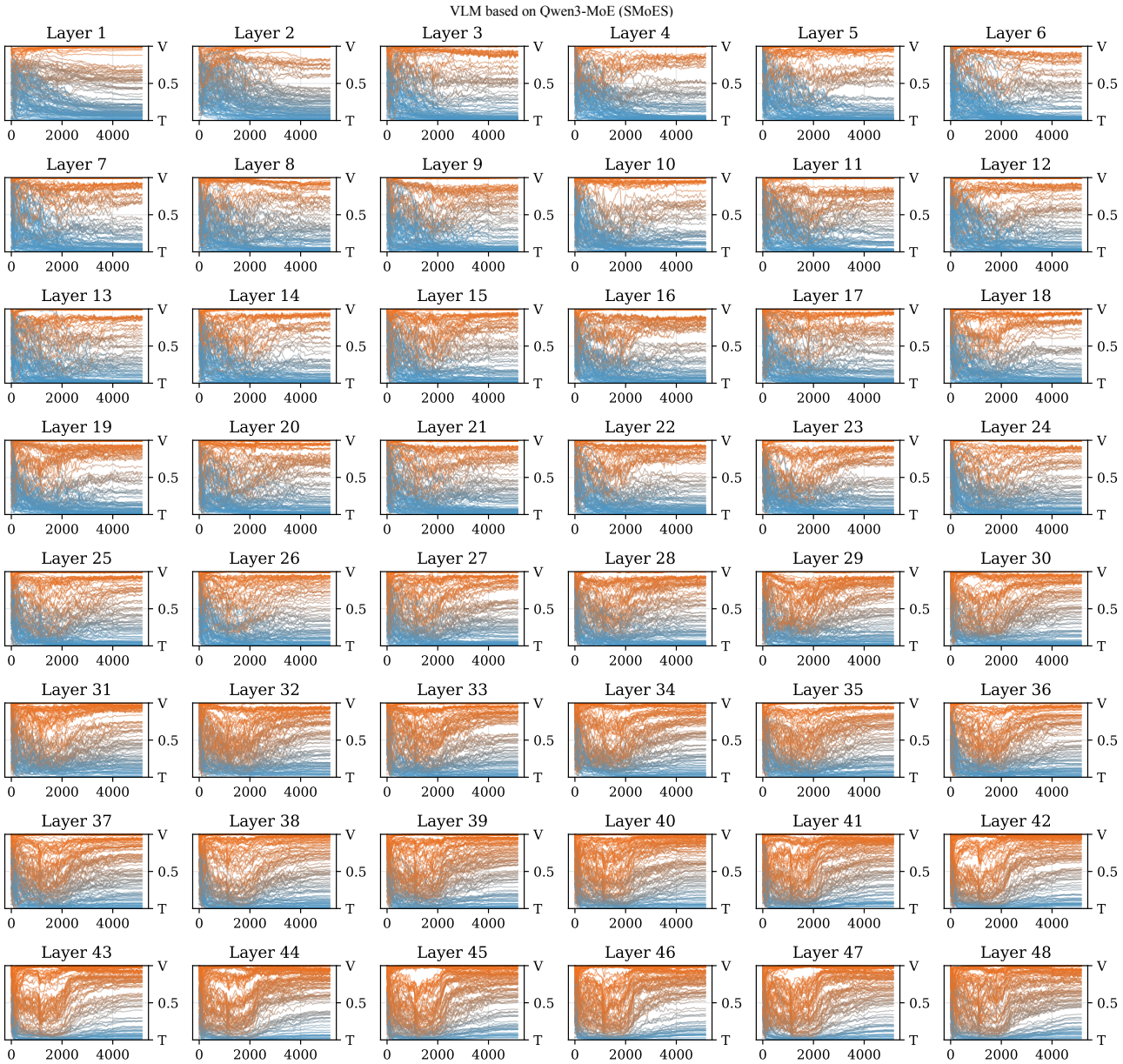

Figure S9. Evolution of expert specialization during training on Qwen3-MoE with SMoES. Each curve represents an expert. Horizontal axis: training steps. Vertical axis: expert specialization score (symmetric expansion of MSI). V: vision specialization; T: text specialization.

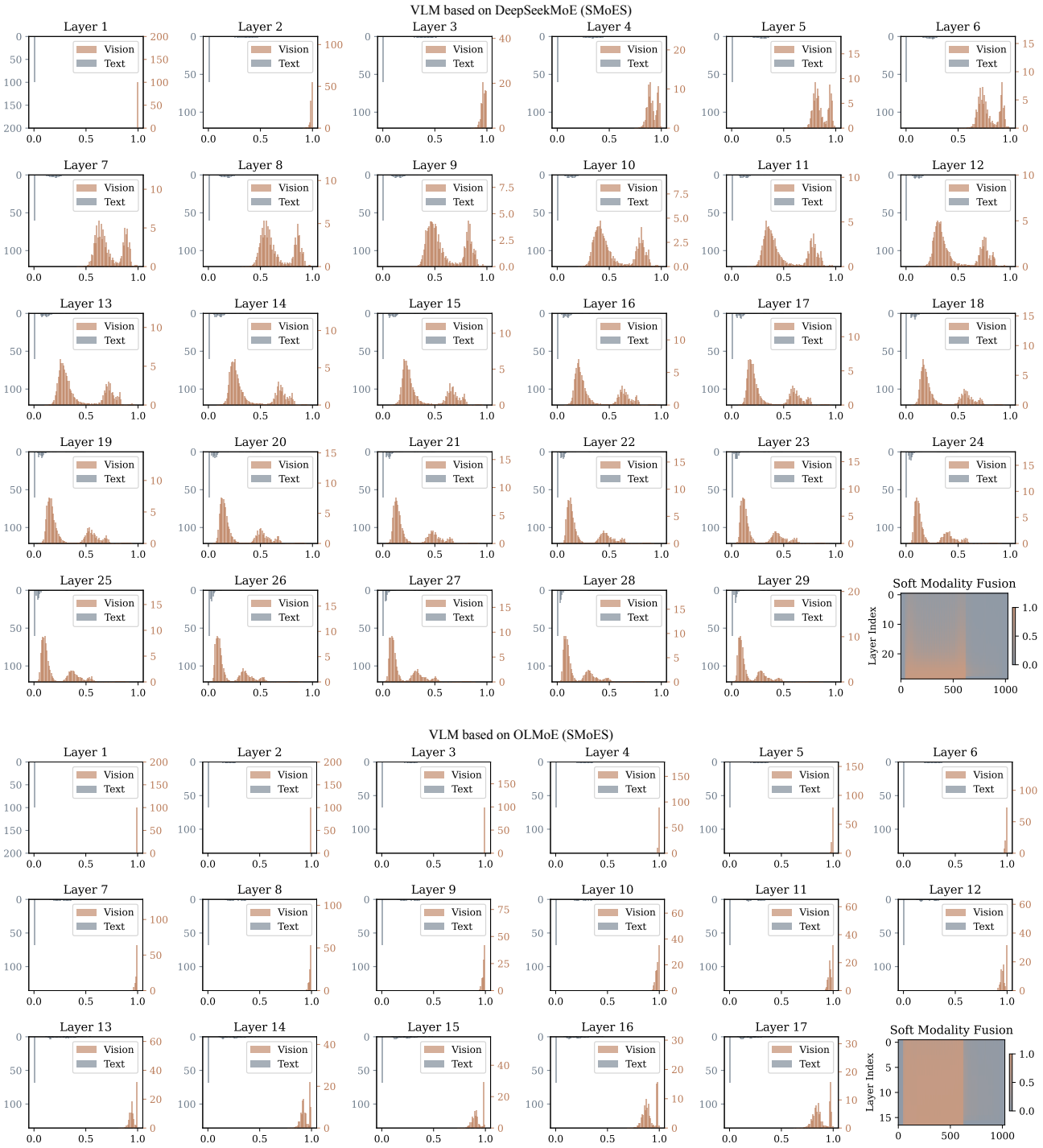

Figure S10. Modality fusion patterns for $\text{SMoES}_{\text{attention-soft}}$ in DeepSeekMoE and OLMoE. Layer subplots: horizontal axis is the soft modality score, and vertical axis is the number of token distribution. Soft modality fusion subplot: horizontal axis is the token id in a sample sequence, and vertical axis is the layer index.

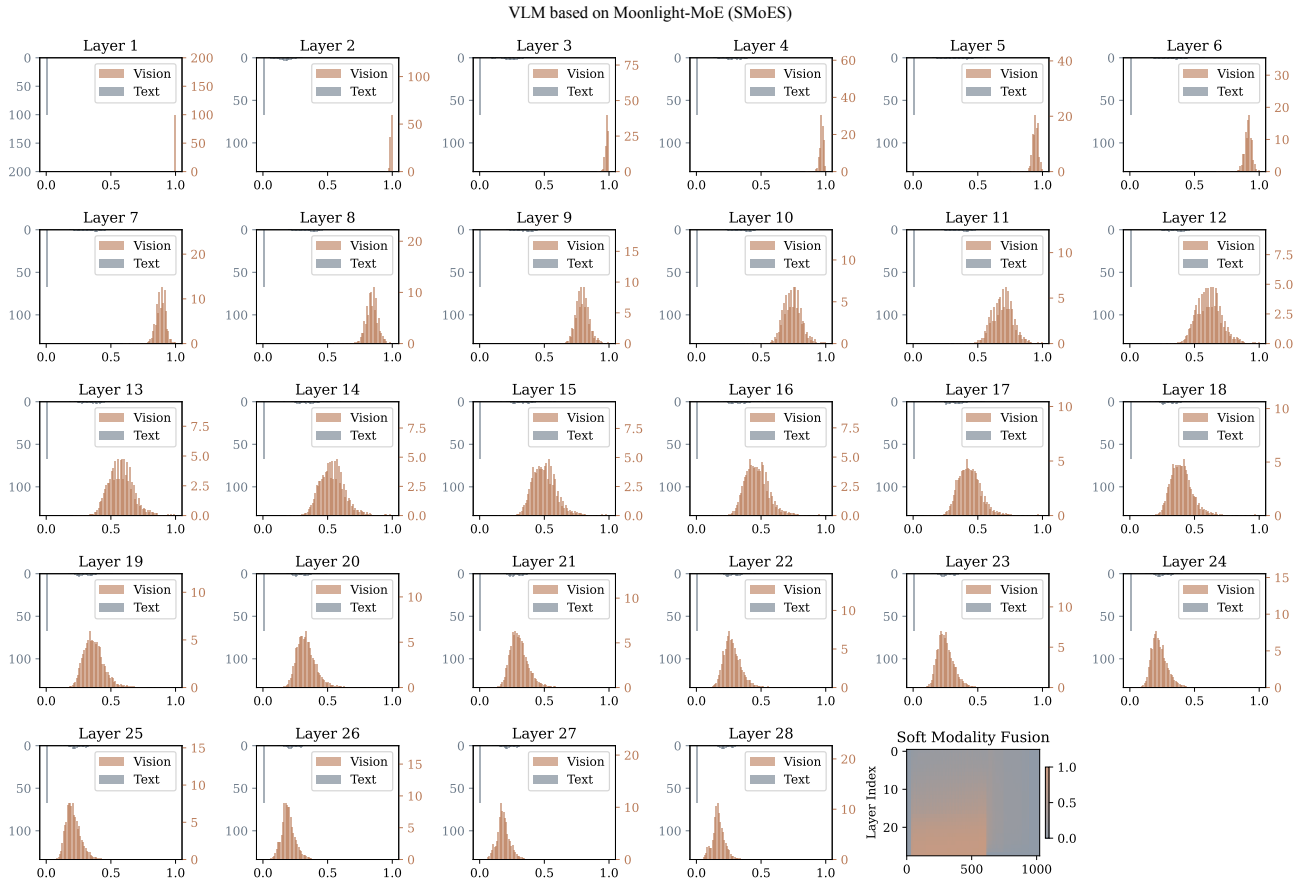

Figure S11. Modality fusion patterns for $\text{SMoES}_{\text{attention-soft}}$ in Moonlight-MoE. Layer subplots: horizontal axis is the soft modality score, and vertical axis is the number of token distribution. Soft modality fusion subplot: horizontal axis is the token id in a sample sequence, and vertical axis is the layer index.

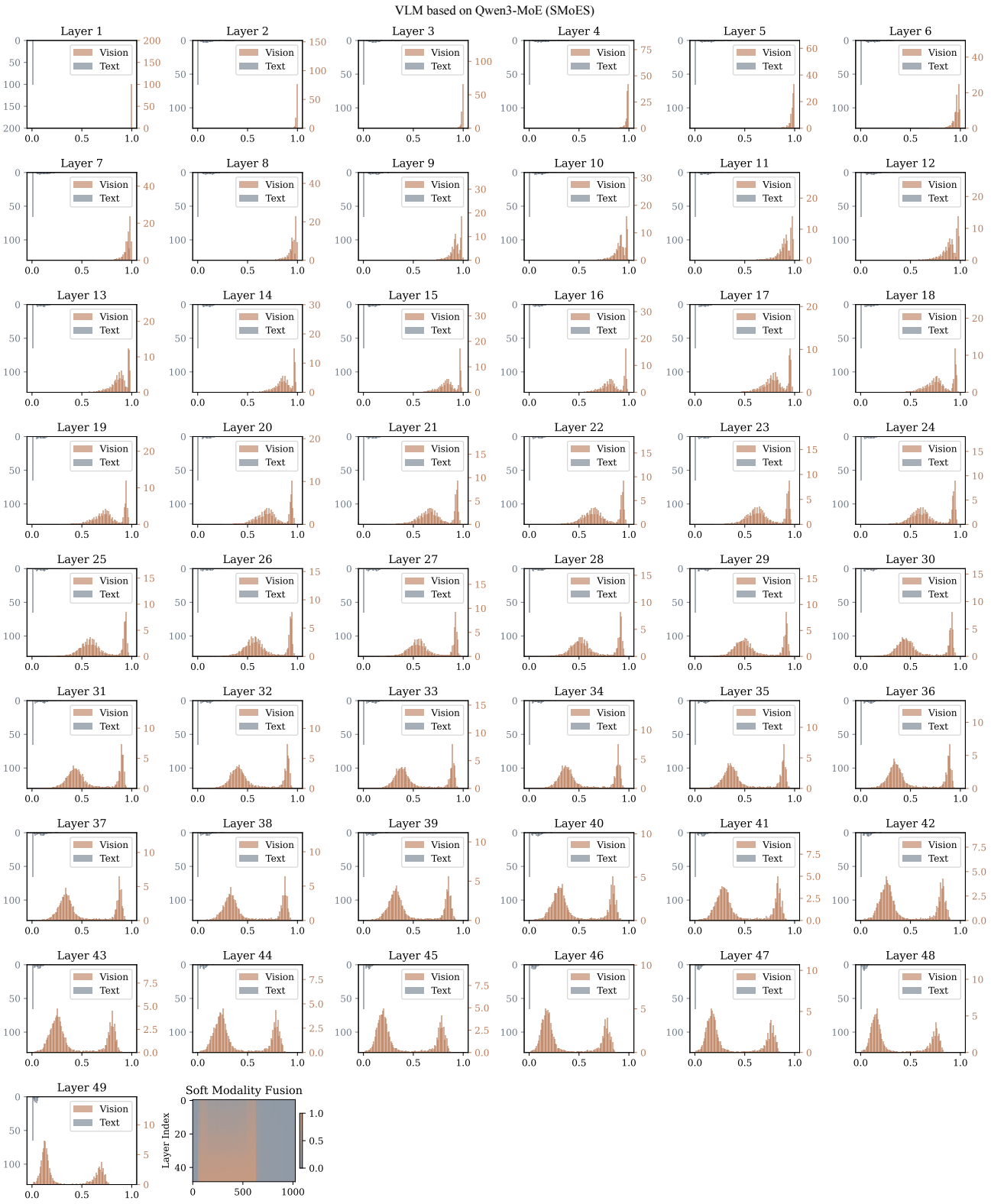

Figure S12. Modality fusion patterns for $\text{SMoES}_{\text{attention-soft}}$ in Qwen3-MoE. Layer subplots: horizontal axis is the soft modality score, and vertical axis is the number of token distribution. Soft modality fusion subplot: horizontal axis is the token id in a sample sequence, and vertical axis is the layer index.