SAGE: SUBSPACE ALIGNMENT FOR CLASSIFIER-FREE GUIDANCE IN MIXTURE-OF-EXPERTS DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Transformers with Mixture-of-Experts (MoE) routing are a leading recipe for scaling generative models. Classifier-Free Guidance (CFG) is essential for generation quality, yet excessively high guidance scales trigger collapse. We identify a previously unreported failure mode in their combination: the two CFG branches route independently, so their realized activations occupy different subspaces. The unconditional write then leaves the conditional subspace, and CFG amplifies that residual linearly in the guidance scale. We propose SAGE, a training-time regularizer that aligns unconditional MoE activations to the conditional subspace without restricting routing diversity, at zero inference cost. Toy experiments show that SAGE dramatically suppresses extreme drift by $9.2\times$. When scaled to a 1B-parameter text-to-image model, SAGE significantly improves generation quality, delivering a 9.3% boost in peak DPG-Bench performance. Extensive experiments demonstrate that SAGE consistently outperforms the baseline.

1 INTRODUCTION

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) and their flow-matching counterparts (Lipman et al., 2023; Liu et al., 2023) have become the dominant paradigm for high-fidelity generation (Seedance, 2026; Happyhorse, 2026). Following the success of sparse experts in language modeling (Fedus et al., 2022; Lepikhin et al., 2021), recent work scales DiTs via Mixture-of-Experts (MoE) feed-forward blocks, including DiT-MoE (Fei et al., 2024), EC-DiT (Sun et al., 2025), Race-DiT (Yuan et al., 2025), and others (Cheng et al., 2025; Zheng et al., 2025). In parallel, Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) remains the indispensable inference knob that trades diversity for fidelity by linearly combining conditional and unconditional velocity fields. Virtually every modern text-to-image system ships with CFG enabled, and practitioners routinely tune the guidance scale s to reach the operating point they want. These two ingredients, sparse experts for capacity and CFG for controllability, are therefore combined in almost every large-scale generative model being built today.

Despite the popularity of both MoE scaling and CFG conditioning, we find that their combination harbors a previously overlooked pathology. When CFG is applied to an MoE-DiT, sample quality degrades far more severely than in a parameter-matched dense model as the guidance scale increases. Because the conditional and unconditional forward passes produce different hidden states, the routers select different expert subsets for the two branches, and the CFG linear combination mixes outputs from incompatible subspaces, injecting a spurious error that grows with the guidance scale. This routing divergence is intrinsic to MoE and vanishes identically in dense architectures.

We trace this failure to a geometric mechanism. Different hidden states $h_c \neq h_u$ cause the Top-k routers to select different expert subsets $S_c \neq S_u$. Write D_{route} for this routing misalignment, and let $\mathcal{V}_{S_c}$ be the subspace spanned by the realized conditional activations. CFG then injects a leak $\delta(s)$ outside $\mathcal{V}_{S_c}$ that grows with the guidance scale s. In a dense model, $D_{\text{route}} \equiv 0$, so the failure is MoE-intrinsic.

The mechanism suggests an obvious fix: forcing the two branches onto the same routes, for instance by penalizing the divergence between their routing distributions. We find this approach to be fun-

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.

damentally flawed. It removes the mismatch, but it also eliminates the specialization that motivates MoE. Experiments confirm that constraining routing freedom degrades performance.

Our starting point is that CFG is closed in any common subspace: if both writes lie in a span $\mathcal{V}$, then so does v_{cfg}, regardless of which experts produced them. It is therefore enough to align the unconditional activations to the conditional subspace $\mathcal{V}_{S_c}$, while leaving the routers free. This observation motivates **SAGE**, **S**ubspace **A**lignment for CF**G** in Mo**E** Diffusion Models, a training-time regularizer that penalizes only the component of the unconditional MoE output that lies outside the stacked conditional activations. SAGE requires no architectural change, adds one loss during training, preserves routing diversity, and incurs zero inference cost.

Empirically, the gains are substantial. In toy experiments, SAGE cuts the worst-case drift at high guidance by $9.2\times$ and largely restores mode purity where the baseline has collapsed. We further scale SAGE up to a 1B-parameter MoE-DiT on the text-to-image task, where it improves the peak DPG-Bench score by 9.3% and keeps that quality as guidance grows. The same pattern holds on GenEval, where SAGE lifts the peak score by 2.7 pp, and the qualitative comparisons show coherent structure at high guidance where the baseline has already broken down.

We summarize our contributions as follows:

- We uncover a novel failure mode specific to MoE-DiTs, termed routing-induced subspace leakage: routing misalignment forces conditional and unconditional branches into disjoint subspaces, which CFG subsequently amplifies linearly with the guidance scale. We show this structural leak is unique to sparse models and vanishes in dense architectures.
- We propose SAGE, a training-time regularizer that aligns unconditional activations back into the conditional subspace while maintaining expert routing diversity and incurring zero inference overhead. Crucially, we demonstrate that the intuitive alternative of forcing hard routing agreement is markedly sub-optimal.
- Across extensive experiments on large-scale text-to-image MoE-DiTs, SAGE consistently outperforms baseline models across various CFG scales. Controlled toy experiments and ablation studies validate the underlying mechanism and design choices of SAGE.

2 Related Work

2.1 Mixture-of-Experts in Diffusion Models

Sparse mixture-of-experts (MoE) architectures (Lepikhin et al., 2021; Fedus et al., 2022) have emerged as a dominant recipe for scaling diffusion transformers (Peebles & Xie, 2023) beyond the capacity of dense feed-forward networks (FFNs). RAPHAEL (Xue et al., 2023) pioneered the use of space-MoE and time-MoE layers for text-to-image diffusion, demonstrating that routing along spatial and temporal axes yields strong artistic generation. DiT-MoE (Fei et al., 2024) scales sparse DiTs to 16.5B parameters and achieves state-of-the-art performance. EC-DiT (Sun et al., 2025) replaces token-choice routing with expert-choice routing (Zhou et al., 2022) to allocate compute adaptively across image patches, while Diff-MoE (Cheng et al., 2025) and Race-DiT (Yuan et al., 2025) further explore time-aware and space-adaptive expert selection and flexible per-token expert counts. Dense2MoE (Zheng et al., 2025) takes a post-hoc route, restructuring pre-trained FLUX-style (Black Forest Labs, 2024) dense DiTs into sparse MoE backbones with 60% fewer activated parameters. Almost all of these works evaluate their models with classifier-free guidance (Ho & Salimans, 2022), yet none of them study how MoE routing interacts with the dual conditional/unconditional forward passes that CFG requires. Most recently, ProMoE (Wei et al., 2026) explicitly addresses the sub-optimal interaction between MoE routing and CFG by partitioning tokens into conditional and unconditional subsets via a hard first-stage router and assigning dedicated experts to the unconditional branch. While this design provides partial relief analogous to that offered by the shared-expert mechanism analyzed in section 3.2, it does not align the realized activations of the remaining routed experts and sacrifices one expert's capacity for conditional generation.

2.2 Classifier-Free Guidance

Classifier-free guidance has become the mainstream inference-time mechanism for conditional diffusion, as it improves generation results and enables better semantic control. It relies on jointly

trained conditional and unconditional predictions, which are combined as $(1+s)\,v_c - s\,v_u$ at inference. Improving CFG has motivated a line of corrective methods (Hong et al., 2023; Zheng & Lan, 2024; Ahn et al., 2024; Kynkäänniemi et al., 2024; Sadat et al., 2025b). GLIDE (Nichol et al., 2022) empirically compared CFG to CLIP guidance (Radford et al., 2021) for text-to-image synthesis, and subsequent work has proposed CFG rescaling, dynamic schedules, and spatial-inconsistency analyses to mitigate artifacts at high s (Shen et al., 2024). CADS (Sadat et al., 2024) anneals the conditioning to restore diversity, APG (Sadat et al., 2025a) decomposes the guidance term to remove the oversaturation and artifacts that appear at large scales, and CFG-Zero* (Fan et al., 2025) rescales and zero-initializes the guidance to correct inaccurate velocity estimates. CFG++ (Chung et al., 2025) attributes guidance failures to iterates drifting off the data manifold and addresses this by reformulating CFG as a manifold-constrained inverse problem solved at sampling time. Another line of work replaces the null-condition branch altogether: PAG (Ahn et al., 2024) perturbs self-attention to synthesize a deliberately degraded prediction as the guidance baseline, in the spirit of self-attention guidance (Hong et al., 2023), and Karras et al. (2024) propose autoguidance, which improves performance by additionally training an auxiliary model. Li et al. (2025) investigates how guidance amplifies the top singular directions of the conditional score. Our analysis complements theirs by revealing a qualitatively different failure mode unique to sparse architectures: routing misalignment places the two CFG branches in different realized subspaces, and CFG then scales the residual of the unconditional write outside the conditional subspace. Unlike autoguidance or guidance-distillation approaches (Meng et al., 2023; Jensen & Sadat, 2025), which modify or bypass the unconditional branch, we resolve this structural mismatch during training via an architecture-aware subspace constraint that incurs zero inference cost.

3 SAGE: Subspace Alignment for Classifier-Free Guidance

3.1 Preliminaries

Flow Matching. Flow matching (Lipman et al., 2023; Liu et al., 2023) learns a velocity field $v_\theta(x, t, c) : \mathbb{R}^d \times [0, 1] \times \mathcal{C} \to \mathbb{R}^d$ transporting $\mathcal{N}(0, I_d)$ to $p_{\text{data}}(\cdot \mid c)$. With the flow interpolant $x_t = t\,x_1 + (1-t)\,\varepsilon$, the loss is

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x_1,\varepsilon,t,c}\Big[\big\|v_\theta(x_t, t, c) - (x_1 - \varepsilon)\big\|_2^2\Big]. \tag{1}$$

Classifier-Free Guidance. Classifier-free guidance (Ho & Salimans, 2022) uses a single model to handle both conditional and unconditional regimes by replacing c with a null token $\varnothing$ at random during training. At inference, the two branches are combined as

$$v_{\text{cfg}}(x, t, c, s) = (1 + s)\,v_\theta(x, t, c) - s\,v_\theta(x, t, \varnothing). \tag{2}$$

MoE Layer. An MoE layer (Lepikhin et al., 2021; Fedus et al., 2022) replaces the FFN with N parallel experts and a sparse router. Given a hidden state $h \in \mathbb{R}^{d_h}$, the router selects a top-k active set $S(h) \subseteq \{1, \ldots, N\}$ with gating weights $g_i(h)$, and each expert E_i is a two-layer FFN with output projection $W_2^{(i)} \in \mathbb{R}^{d_h \times d_f}$:

$$f_{\text{MoE}}(h) = \sum_{i \in S(h)} g_i(h)\,E_i(h). \tag{3}$$

3.2 Routing Misalignment and Subspace Alignment

We write $v_c = f_{\text{MoE}}(h_c)$ and $v_u = f_{\text{MoE}}(h_u)$ for the MoE outputs under condition c and the null token $\varnothing$, respectively, with active sets S_c and S_u. Because the conditioning signal is mixed into earlier layers, $h_c \neq h_u$, so the routers select different experts: $S_c \neq S_u$ in general. We quantify this with

$$D_{\text{route}} := 1 - \frac{|S_c \cap S_u|}{k} \in [0, 1]. \tag{4}$$

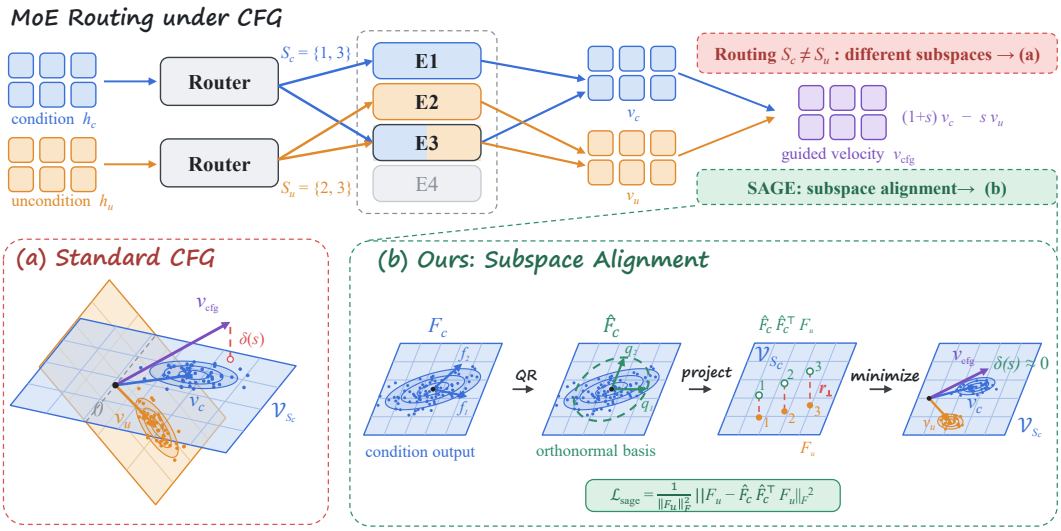


Figure 1: **Subspace alignment. Top:** Conditional and unconditional hidden states h_c and h_u are routed independently, so the realized activations occupy different subspaces. **(a)** Standard CFG injects a leak $\delta(s)$. **(b)** SAGE uses thin QR of the stacked conditional outputs F_c to yield an orthonormal basis Q_c of $\mathcal{V}_{S_c}$. Projecting F_u onto this subspace and penalizing the residual with $\mathcal{L}_{\text{sage}}$ enforces $v_u \in \mathcal{V}_{S_c}$ at training time.

As illustrated in fig. 1 (a), v_c and v_u live in different subspaces, so the CFG combination (2) mixes incompatible directions. Decomposing this combination into shared and exclusive parts gives:

$$v_{\text{cfg}} = \underbrace{\sum_{i \in S_c \cap S_u} \left[(1+s)\, g_i^c E_i(h_c) - s\, g_i^u E_i(h_u)\right]}_{v_{\text{shared}}} + (1+s) \sum_{i \in S_c \setminus S_u} g_i^c E_i(h_c) \;-\; s \sum_{j \in S_u \setminus S_c} g_j^u E_j(h_u). \tag{5}$$

The exclusive unconditional term is a write that the conditional branch did not produce in this pass. Let $\mathcal{V}_{S_c}$ be the rank-m section of stacked conditional activations defined below, with projector P_c, and define the leak

$$\delta(s) \;:=\; -s\,(I - P_c)\, v_u. \tag{6}$$

Then $v_{\text{cfg}} = (1+s)v_c - s\,P_c v_u + \delta(s)$, and $\|\delta(s)\| = s\,\|(I - P_c)v_u\|$. The leak is exactly linear in s and vanishes if and only if $v_u \in \mathcal{V}_{S_c}$. Exclusive experts are the MoE-specific source of this residual (Appendix A); a dense FFN has $D_{\text{route}} \equiv 0$ and no exclusive experts.

If $v_c, v_u \in \mathcal{V}$ for any subspace $\mathcal{V}$, then $v_{\text{cfg}} \in \mathcal{V}$: CFG is closed. We need neither $S_c = S_u$ nor a shared global basis. As in fig. 1, it is enough that v_u lie in the conditional subspace. SAGE therefore penalizes only the component of the unconditional write outside $\mathcal{V}_{S_c}$,

$$\mathcal{L}_{\text{sage}}(\theta) := \mathbb{E}\left[\left\|(I - P_c)\, f_{\text{MoE}}(h_u)\right\|_2^2\right]. \tag{7}$$

In a minibatch of B samples with T tokens per sample, stack the layer outputs into $F_c, F_u \in \mathbb{R}^{d_h \times n}$ with $n = BT$. Full-span QR of F_c is vacuous whenever $n \geq d_h$. We therefore draw a column subset Ω of size

$$m = \min\left(n, \lfloor d_h / k_{\text{split}} \rfloor\right), \tag{8}$$

where the integer k_{split} is chosen so that $\mathcal{V}_{S_c}$ remains a proper subspace whenever n is large. Stop-gradient F_c and take a thin QR of the section $F_{c,\Omega} = \hat{F}_c R$:

$$\mathcal{L}_{\text{sage}} = \frac{1}{\|F_u\|_F^2} \left\|F_u - \hat{F}_c \hat{F}_c^\top F_u\right\|_F^2. \tag{9}$$

Algorithm 1 MoE-DiT Training with SAGE

Require: Dataset $\mathcal{D}$, MoE-DiT v_θ, SAGE weight λ, CFG dropout p_{drop}, split factor k_{split}

for each training iteration **do**
  Sample minibatch $\{(x_1, c)\} \sim \mathcal{D}$, $t \sim \mathcal{U}[0,1]$, $\varepsilon \sim \mathcal{N}(0, I)$
  $x_t \leftarrow t\, x_1 + (1-t)\,\varepsilon$; $y \leftarrow x_1 - \varepsilon$
  Conditional pass: $v_c, \{f^{(\ell)}_{\text{MoE}}(h_c)\}_\ell \leftarrow \text{Forward}(v_\theta, x_t, t, c)$
  Unconditional pass: $v_u, \{f^{(\ell)}_{\text{MoE}}(h_u)\}_\ell \leftarrow \text{Forward}(v_\theta, x_t, t, \varnothing)$
  Draw $u \sim \text{Bernoulli}(p_{\text{drop}})$; $\mathcal{L}_{\text{FM}} \leftarrow \|v_u - y\|^2$ if u else $\|v_c - y\|^2$
  for each MoE layer ℓ **do**
    $F_c \leftarrow \text{stack}(f^{(\ell)}_{\text{MoE}}(h_c))$; $F_u \leftarrow \text{stack}(f^{(\ell)}_{\text{MoE}}(h_u))$ $\{F_c, F_u \in \mathbb{R}^{d_h \times n}, n = BT\}$
    $m \leftarrow \min(n, \lfloor d_h / k_{\text{split}} \rfloor)$; sample m columns Ω; $\hat{F}_c, _ \leftarrow \text{thinQR}(\text{sg}(F_{c,\Omega}))$
    $\mathcal{L}^{(\ell)}_{\text{sage}} \leftarrow \frac{1}{\|F_u\|_F^2} \|F_u - \hat{F}_c \hat{F}_c^\top F_u\|_F^2$
  end for
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{FM}} + \lambda \sum_\ell \mathcal{L}^{(\ell)}_{\text{sage}}$
  $\theta \leftarrow \theta - \text{lr} \cdot \nabla_\theta \mathcal{L}_{\text{total}}$
end for

The cost is $\mathcal{O}(d_h m^2 + d_h m n)$ per layer, and the total objective is

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{FM}}(\theta) + \lambda \sum_\ell \mathcal{L}^{(\ell)}_{\text{sage}}(\theta), \qquad \lambda > 0. \tag{10}$$

The batch leak energy is then $s\,\varepsilon\,\|F_u\|_F$ (proposition A.3), where ε is the training residual. SAGE needs a paired unconditional forward on the same x_t in order to form (F_c, F_u). The flow-matching term still follows standard CFG dropout: with probability p_{drop} one supervises v_u, otherwise v_c. Inference is unchanged. The procedure is algorithm 1.

4 Experiments

We validate SAGE in two complementary settings. First, we use a 2-D Gaussian mixture toy experiment (section 4.1) to empirically verify the leak arising from CFG in MoE-DiTs. In this setting, the leak $\|\delta(s)\|$ is directly measurable, and the true likelihood is available in closed form. Subsequently, we demonstrate the effectiveness of SAGE on a large-scale, 1B-parameter MoE-DiT for text-to-image generation (section 4.2), showing that our method successfully scales to high-dimensional data.

4.1 Toy Experiment

We construct a 2-D class-conditional Gaussian mixture with K=8 isotropic modes (r=5, σ=0.3; 80,000 training samples). The backbone is a flow-matching (Lipman et al., 2023) multilayer perceptron (MLP) with a single MoE layer (N=8 experts, top-k=2, d_h=64, d_f=16). A dense FFN of width 32 (matching the number of active parameters) serves as a control. With T=1, SAGE uses every available conditional write, so k_{split} is not needed. We compare three configurations: Dense, MoE, and MoE + SAGE (λ=1), all trained with AdamW (Loshchilov & Hutter, 2019) for 20,000 steps and sampled using a 50-step Euler ODE solver.

In this toy setup, conditional generation corresponds to class-specific point clouds, while the unconditional counterpart spans the remaining empty regions. Thus, applying CFG essentially concentrates the points of each class toward their respective centers. In theory, a larger guidance scale should drive the empirical cluster centers closer to the ground-truth centers. However, due to the inherent CFG bias, an excessively large guidance scale often causes the generated clusters to deviate from the ground-truth centers.

Figure 2 visualizes inference results for each model across four CFG scales, with per-class excess drift Δd relative to s=1 shown as bar charts below each panel; the annotation in each bar chart reports the maximum Δd. Table 1 quantifies two complementary metrics: the mean Δd averaged

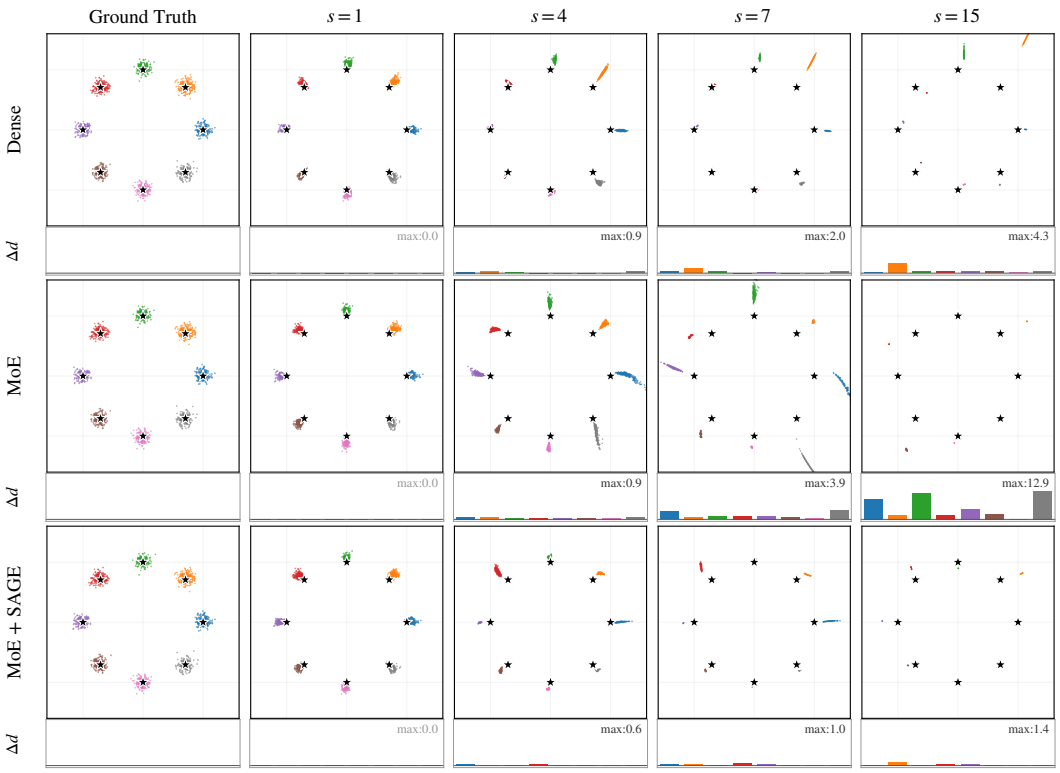


Figure 2: **Toy experiment sample grids** for Dense, MoE, and MoE+SAGE across $s \in \{1, 4, 7, 15\}$, with per-class center drift (Δd; lower is better). The baseline MoE exhibits increasing drift at large s; SAGE keeps the drift substantially smaller.

over all eight classes (with the per-class maximum in parentheses, matching the bar-chart annotations in fig. 2) and mode purity. At s=7, the maximum per-class excess drift for MoE reaches 3.9, over 1.9× that of Dense, while MoE + SAGE stays at only 1.0. At s=15, the gap widens dramatically: the maximum per-class excess drift for MoE reaches 12.9, while MoE + SAGE remains at 1.4 (9.2× down). Mode purity tells the same story: MoE collapses to 0.121 at s=7 while MoE + SAGE maintains 0.781 (a 6.5× improvement). These results confirm that the subspace leak from routing misalignment is MoE-specific, and that SAGE effectively neutralizes it.

Table 1: **Quantitative results on the toy experiment.** Excess drift Δd reports the mean per-class distance increase relative to s=1 (matching the bar charts in fig. 2); the per-class maximum is shown in parentheses for cross-reference with the bar-chart annotations. Mode purity (higher is better) measures separation across the eight modes. MoE drift explodes at high guidance while MoE + SAGE stays close to Dense.

	Excess Drift Δd (↓)			Mode Purity (↑)				
Method	s=4	s=7	s=15	s=1	s=2	s=4	s=7	s=15
Dense	0.33 (0.9)	0.50 (2.0)	1.06 (4.3)	.993	.968	.794	.586	.501
MoE	0.72 (0.9)	1.83 (3.9)	5.59 (12.9)	.989	.929	.394	.121	.125
MoE + SAGE	0.34 (0.6)	0.39 (1.0)	0.39 (1.4)	.997	.989	.917	.781	.666

4.2 Large-Scale Text-to-Image Generation

4.2.1 Setup

We train a 1B-parameter MoE-DiT consisting of 12 Transformer layers (Vaswani et al., 2017) with hidden dimension d_h=1536 and 12 attention heads (128 dimensions per head). Each layer contains 8 routed experts and 1 shared expert, with top-2 routing applied to the routed experts. The intermediate

Table 2: **Overall scores on DPG-Bench and GenEval across CFG scales.** SAGE consistently outperforms the Baseline at all $s \geq 3$ and degrades more gracefully under high guidance, while the KL routing constraint hurts performance across the board. Δ (peak→10) measures the change from the peak score to the score at s=10; Δ_{base} shows the difference from the Baseline's peak score. **Bold** denotes the best score.

		CFG guidance scale s						
Benchmark	Method	1	3	5	7	10	Δ (peak→10)	Δ_{base}
DPG-Bench	Baseline	**54.77**	61.76	56.33	52.38	48.56	−13.20	–
	SAGE (λ=1)	52.30	**67.49**	**65.63**	**63.36**	**59.55**	−7.94	+5.73
	KL (λ_{kl}=1)	40.95	50.65	46.82	44.38	41.08	−9.57	−11.11
GenEval	Baseline	**30.3**	57.1	55.5	53.2	47.1	−10.0	–
	SAGE (λ=1)	26.5	**59.8**	**59.3**	**57.1**	**52.5**	−7.3	+2.7
	KL (λ_{kl}=1)	13.8	36.0	34.5	32.2	29.5	−6.5	−21.1

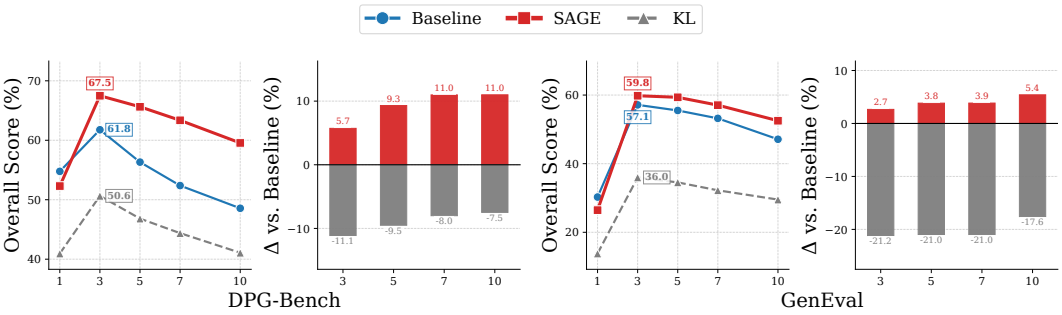


Figure 3: **Performance improvements of SAGE across CFG scales.** SAGE (red) maintains high performance across the full guidance range, while the Baseline (blue) peaks at s=3 and declines sharply. The KL routing constraint (gray) consistently underperforms both methods, confirming that naive distributional alignment of routing decisions is counterproductive.

dimensions of the routed and shared experts are 1024 and 2048, respectively. SAGE is applied with k_{split}=2. Text conditioning is provided by a frozen Qwen2.5-VL-7B-Instruct encoder (Bai et al., 2025).

We curate a 100M-image subset of LAION-5B (Schuhmann et al., 2022), re-caption it with Qwen2.5-VL 72B, and resize and crop all images to a resolution of approximately 256×256. We use AdamW with a learning rate of 10^{-4}, a constant learning-rate schedule following a 1,000-step warmup, and a global batch size of 512 on 32 NVIDIA H200 GPUs. We set p_{drop}=0.1, the MoE auxiliary load-balancing loss coefficient (Fedus et al., 2022) to α_{aux}=0.001, and the router z-loss coefficient (Zoph et al., 2022) to α_z=0.001. Three configurations are compared: Baseline (λ=0), SAGE (λ=1), and KL Routing Constraint (λ_{kl}=1), which penalizes the KL divergence between the conditional and unconditional routing distributions to force $S_c \approx S_u$. All models are trained for the same 200,000 steps to ensure a fair comparison.

We use two benchmarks: (a) DPG-Bench (Hu et al., 2024), evaluated with an mPLUG backbone (Li et al., 2022), for which we report the overall score and a per-category breakdown; and (b) GenEval (Ghosh et al., 2023), with 553 prompts and 4 seeds per prompt. CFG scales are swept over $s \in \{1.0, 3.0, 5.0, 7.0, 10.0\}$.

4.2.2 Main Results

Quantitative comparison. The ordering SAGE > Baseline > KL holds consistently at all $s \geq 3$ on both benchmarks (table 2 and fig. 3). SAGE significantly enhances generation quality, boosting peak DPG-Bench performance by 9.3%. Meanwhile, SAGE's robustness is particularly notable: the DPG-Bench score drops by only 7.94 points from its peak to s=10, versus 13.20 for the Baseline; moreover, SAGE at s=7 still exceeds the Baseline's peak. The DPG-Bench evaluation protocol



Figure 4: **Qualitative comparison across CFG scales.** Each panel shows the same prompt rendered by Baseline (top) and SAGE (bottom) at $s \in \{1, 3, 7, 10, 15\}$ from left to right. At s=10 and 15, the Baseline exhibits over-saturation, white patches, and structural collapse, while SAGE preserves coherent composition and realistic textures.

and category-level breakdown are provided in Appendix B.1. On GenEval, the pattern is the same: SAGE retains a $+5.4$ pp advantage at s=10 (52.5% vs. 47.1%) and loses only 7.3 pp from its peak, compared with 10.0 pp for the Baseline. The largest per-task improvement is on color-attribute binding ($+11.5$ pp; full breakdown in Appendix B.2), a compositional task that relies heavily on precise conditional control, exactly where a write outside the conditional subspace is most damaging. To confirm that the CFG-robustness benefit of SAGE also holds for distributional image fidelity, we report FID (Heusel et al., 2017) across CFG scales in Appendix B.5.

The KL routing constraint, which penalizes $D_{\mathrm{KL}}(\pi_c \| \pi_u)$ to force consistent routing, is counterproductive on every metric. By collapsing the routing distributions, it destroys expert specialization: the model can no longer assign different experts to different semantic attributes, resulting in a -21 pp drop on GenEval relative to the Baseline at s=3. This provides strong empirical evidence for the theoretical insight of section 3.2: the correct fix is subspace alignment, not routing agreement. In contrast to the KL constraint, SAGE leaves the router essentially untouched: its conditional and unconditional branches stay as routing-diverse as the Baseline at every guidance scale, so subspace alignment is achieved without sacrificing expert specialization (Appendix B.4).

Qualitative comparison. Figure 4 presents several prompts across the full CFG sweep. At moderate guidance ($s \leq 7$), both methods produce plausible images; as s increases to 10 and 15, the Baseline develops over-saturated colors, white-out patches, and structural repetition, all symptomatic of the subspace leak $\delta(s)$ in (6). SAGE preserves coherent composition, fine-grained texture, and realistic color balance even at s=15. The quantitative and qualitative trends therefore agree: the benefit of SAGE is not a benchmark artifact but a visible improvement in image structure at high guidance.

4.2.3 Ablation Study

Keeping the setup of section 4.2.1 fixed and evaluating at guidance scales $s \in \{3, 5\}$, we ablate the two most consequential design choices of SAGE: *when* the SAGE loss is applied and *how strongly* it is weighted. For both ablations, each configuration is evaluated on DPG-Bench and GenEval at each scale. Two further ablations, covering *how often* the loss needs to be activated and *how much*

Table 3: **Training timing and loss weight analysis. Left:** comparison of from-scratch SAGE and post-hoc fine-tuning of a pre-trained baseline for 5k or 10k steps, with λ=1 for all SAGE variants. **Right:** effect of the SAGE loss weight λ; the default λ=1 performs best among the tested values on both benchmarks at both guidance scales.

	s=3		s=5	
Method	DPG	GenEval	DPG	GenEval
Baseline	61.76	57.1	56.33	55.5
SAGE (from scratch)	67.49	59.8	65.63	59.3
Post-hoc SAGE (5k)	61.32	56.3	56.23	54.8
Post-hoc SAGE (10k)	60.20	55.3	55.73	53.5

	s=3		s=5	
λ	DPG	GenEval	DPG	GenEval
0.5	65.31	58.8	62.07	57.8
1	67.49	59.8	65.63	59.3
2.0	64.85	58.6	63.33	58.2
5.0	59.94	56.2	58.69	55.8

of its effect a shared expert already provides for free, are deferred to Appendix B.3 and use the same two guidance scales.

Training timing. A natural question is whether SAGE can be applied post-hoc to an already-trained baseline by fine-tuning with $\mathcal{L}_{\text{sage}}$ for a small number of additional steps. Table 3 compares SAGE trained from scratch with post-hoc fine-tuning for 5k and 10k steps, all with λ=1. Applying SAGE from the start yields better performance, whereas post-hoc fine-tuning fails to improve on the baseline. We hypothesize that once the two branches' activation clouds have specialized apart during pre-training, a short SAGE fine-tune cannot pull F_u into the conditional section without harming the flow-matching fit: the residual $\text{range}(F_u) \not\subseteq \mathcal{V}_{S_c}$ has already crystallized, and rotating the writes into containment requires training from scratch.

Loss weight sensitivity. To assess sensitivity to the loss weight at scale, we vary $\lambda \in \{0.5, 1, 2.0, 5.0\}$. table 3 shows that the default weight λ=1 performs best on both benchmarks at both guidance scales. Relative to the baseline in the left panel, reducing the weight to 0.5 yields only small gains, while increasing it to 2.0 or 5.0 lowers all four scores below the baseline. This peaked response is consistent with a trade-off between subspace alignment and flow-matching fit: an excessively large SAGE loss weight may restrict the representations needed for accurate velocity prediction. proposition A.3 identifies leak energy with the SAGE residual; it does not imply that benchmark scores improve monotonically with λ.

Other ablations. Appendix B.3 reports further ablations on activation frequency and the shared expert, together with a comparison against ProMoE (Wei et al., 2026), at $s \in \{3, 5\}$. Activating SAGE every second or fifth training step retains most of the gains from applying it every step, indicating that continuous activation is unnecessary. Removing the shared expert lowers baseline performance, but SAGE still improves both benchmarks, supporting the view that a shared expert alone does not replace explicit subspace alignment. ProMoE improves over the baseline but remains below the default SAGE configuration at both guidance scales.

5 Conclusion and Future Work

We identified a subspace leak in MoE diffusion models. Routing misalignment lets the two CFG branches realize different subspaces; exclusive experts let the unconditional write leave the conditional subspace, and CFG amplifies that residual linearly in s. We proposed SAGE, which penalizes the component of the unconditional MoE write outside $\mathcal{V}_{S_c}$ at training time, requires no architectural change, and incurs zero inference cost. From a controlled toy microscope to a 1B text-to-image model, SAGE consistently improves CFG stability.

Our large-scale evaluation covers models up to the 1B-parameter scale. Scaling SAGE to 100B+ models, analyzing how errors compound across layers in deeper architectures, and exploring video generation remain important directions for future work. Furthermore, CFG mechanisms in most generative models are considerably more complex; for instance, text-to-audio-video (T2AV) and reference-to-video (R2V) tasks often employ double CFG and offer promising settings for future investigation.

AI DISCLOSURE

In this work, we used generative AI tools to polish and improve the readability of the manuscript, assist in writing portions of the experimental code, and help create scientific figures. We use Qwen2.5-VL-72B to recaption training data. We have not used generative AI tools for generating synthetic datasets, developing theoretical models, proposing hypotheses, designing research methodology, interpreting results, or any other tasks requiring mandatory disclosure beyond those stated above.

All AI-assisted outputs were rigorously reviewed by the authors. AI-generated code was manually verified, tested for correctness, and integrated into our pipeline; AI-assisted figures were checked against raw data for accuracy; and all AI-polished text was carefully reviewed to ensure it faithfully represents our original ideas and claims. We take full responsibility for the final content of this work, including all text, claims, code, mathematical proofs, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, section 3 specifies the SAGE objective and its implementation, with training pseudocode in algorithm 1. The definitions, conditions, and algebraic derivations underlying our analysis are given in Appendix A. Section 4.1 describes the synthetic data construction and the toy model, training, and sampling settings, while section 4.2.1 documents the text-to-image architecture, data sources and preprocessing, optimization hyperparameters, hardware, and benchmark setup. Additional benchmark evaluation details and score breakdowns are provided in Appendix B.1 and Appendix B.2; Appendix B.5 specifies the FID reference-image preprocessing and paired sampling protocol. Ablation configurations are reported in section 4.2.3 and Appendix B.3, and the routing-diversity and training-health diagnostics are described in Appendix B.4 and Appendix C.

REFERENCES

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *Proceedings of the European Conference on Computer Vision*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Black Forest Labs. Announcing Black Forest Labs, 2024. URL `https://bfl.ai/announcing-black-forest-labs/`.

Kun Cheng, Xiao He, Lei Yu, Zhijun Tu, Mingrui Zhu, Nannan Wang, Xinbo Gao, and Jie Hu. Diff-MoE: Diffusion transformer with time-aware and space-adaptive experts. In *Proceedings of the International Conference on Machine Learning*, 2025.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, 2025.

Weichen Fan, Chenyang Si, Ziqi Liu, and Ziwei Liu. CFG-Zero*: Improved classifier-free guidance for flow matching models. *arXiv preprint arXiv:2503.18886*, 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

Team Happyhorse. Happyhorse 1.0, 2026. URL `https://www.happyhorse.com/`.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Cristian Perez Jensen and Seyedmorteza Sadat. Efficient distillation of classifier-free guidance using adapters. *Transactions on Machine Learning Research*, 2025.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, 2024.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, 2024.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. mPLUG: Effective and efficient vision-language learning by cross-modal skip-connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Xiang Li, Rongrong Wang, and Qing Qu. Towards understanding the mechanisms of classifier-free guidance. In *Advances in Neural Information Processing Systems*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations*, 2024.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2025a.

Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M. Weber. No training, no problem: Rethinking classifier-free guidance for diffusion models. In *International Conference on Learning Representations*, 2025b.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, 2022.

Team Seedance. Seedance 2.0: Advancing video generation for world complexity, 2026. URL https://arxiv.org/abs/2604.14148.

Dazhong Shen, Guanglu Song, Zeyue Xue, Fu-Yun Wang, and Yu Liu. Rethinking the spatial inconsistency in classifier-free diffusion guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Haotian Sun, Tao Lei, Bowen Zhang, Yanghao Li, Haoshuo Huang, Ruoming Pang, Bo Dai, and Nan Du. EC-DIT: Scaling diffusion transformers with adaptive expert-choice routing. In *International Conference on Learning Representations*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yingya Zhang, Yu Liu, and Hongming Shan. Routing matters in MoE: Scaling diffusion transformers with explicit routing guidance. In *International Conference on Learning Representations*, 2026.

Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. RAPHAEL: Text-to-image generation via large mixture of diffusion paths. In *Advances in Neural Information Processing Systems*, 2023.

Yike Yuan, Ziyu Wang, Zihao Huang, Defa Zhu, Xun Zhou, Jingyi Yu, and Qiyang Min. Expert race: A flexible routing strategy for scaling diffusion transformer with mixture of experts. In *Proceedings of the International Conference on Machine Learning*, 2025.

Candi Zheng and Yuan Lan. Characteristic guidance: Non-linear correction for diffusion model at large guidance scale. In *Proceedings of the International Conference on Machine Learning*, 2024.

Youwei Zheng, Yuxi Ren, Xin Xia, Xuefeng Xiao, and Xiaohua Xie. Dense2MoE: Restructuring diffusion transformer to MoE for efficient text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, 2022.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A Theoretical Analysis

This section records the identities used in section 3 of the main paper. CFG would mix the layer writes $v_c = f_{\mathrm{MoE}}(h_c)$ and $v_u = f_{\mathrm{MoE}}(h_u)$. Notation follows the main text: $S_c = \operatorname{supp} g(h_c)$, $S_u = \operatorname{supp} g(h_u)$, $D_{\mathrm{route}} = 1 - |S_c \cap S_u|/k$, and

$$E_i(h) = W_2^{(i)}\sigma(W_1^{(i)}h + b_1^{(i)}) + b_2^{(i)}, \quad W_2^{(i)} \in \mathbb{R}^{d_h \times d_f}. \tag{11}$$

When a shared expert is present it is included in f_{MoE} and in the stacked activations below.

A.1 Independent routing, different subspaces

The two CFG branches route independently, so S_c and S_u need not coincide. Stack the minibatch writes into $F_c, F_u \in \mathbb{R}^{d_h \times n}$, $n = BT$. Let Ω be a set of $m = \min(n, \lfloor d_h/k_{\mathrm{split}} \rfloor)$ column indices (Eq. (8) of the main paper), and define the conditional activation subspace

$$\mathcal{V}_{S_c} \;:=\; \operatorname{span}(F_{c,\Omega}), \qquad P_c \;:=\; \hat{F}_c \hat{F}_c^{\top}, \tag{12}$$

where $\hat{F}_c$ is an orthonormal basis of $\mathcal{V}_{S_c}$ from thin QR of $F_{c,\Omega}$ (stop-gradient on F_c, as in Eq. (9) of the main paper). Then $\dim \mathcal{V}_{S_c} \le m$, so $\mathcal{V}_{S_c}$ is a proper subspace whenever $m < d_h$. It is *not* claimed to contain every column of F_c when $n > m$. The object SAGE uses is this realized subspace $\mathcal{V}_{S_c}$.

Lemma A.1 (CFG closure)**.** *If $v_c, v_u \in \mathcal{A}$ for any subspace $\mathcal{A} \subseteq \mathbb{R}^{d_h}$, then $v_{\mathrm{cfg}} = (1{+}s)v_c - s v_u \in \mathcal{A}$.*

Definition A.2 (Subspace leak)**.** Let $v_{\mathrm{ideal}} := (1{+}s)v_c - s\,P_c v_u$. The leak is the difference

$$\delta(s) \;:=\; v_{\mathrm{cfg}} - v_{\mathrm{ideal}} \;=\; -s\,(I - P_c)\,v_u. \tag{13}$$

Equivalently,

$$v_{\mathrm{cfg}} = (1{+}s)v_c - s\,P_c v_u + \delta(s), \qquad \|\delta(s)\| = s\,\|(I - P_c)v_u\|. \tag{14}$$

So $\delta(s) = 0$ if and only if $v_u \in \mathcal{V}_{S_c}$, and the leak, when present, is exactly linear in s. Independent routing makes $v_u \notin \mathcal{V}_{S_c}$ typical: the unconditional branch can write in directions the conditional activations did not realize. SAGE drives $\delta \to 0$; it does not claim $v_{\mathrm{cfg}} \in \mathcal{V}_{S_c}$ when $n > m$, because tokens of F_c outside Ω need not lie in $\mathcal{V}_{S_c}$. Closure (lemma A.1) is the motivation for aligning v_u to a realized conditional subspace, not a claim that the implemented section contains the whole batch.

A.2 What SAGE minimizes

The implemented loss Eq. (9) of the main paper is the relative energy of F_u outside $\mathcal{V}_{S_c}$. Writing Δ for the matrix whose columns are the per-token leaks (13), the identity

$$\Delta = -s\,(I - P_c)F_u, \qquad \|\Delta\|_F = s\,\|(I - P_c)F_u\|_F \tag{15}$$

holds at every parameter value. With

$$\mathcal{L}_{\mathrm{sage}} = \frac{1}{\|F_u\|_F^2}\|(I - P_c)F_u\|_F^2, \tag{16}$$

one has $\|\Delta\|_F = s\sqrt{\mathcal{L}_{\mathrm{sage}}}\,\|F_u\|_F$.

Proposition A.3 (SAGE controls leak energy)**.** *At any parameter value, $\|\Delta\|_F = s\,\|(I - P_c)F_u\|_F$. In particular, if $\mathcal{L}_{\mathrm{sage}}(\theta^*) = \varepsilon^2$, then*

$$\|\Delta\|_F = s\,\varepsilon\,\|F_u\|_F. \tag{17}$$

No Markov bound and no $\lambda^{-1/2}$ rate are claimed: λ only trades $\mathcal{L}_{\mathrm{sage}}$ against $\mathcal{L}_{\mathrm{FM}}$ during optimization (section 4.2.3 of the main paper). Equation (15) is the identity that Eq. (9) of the main paper drives to zero.

A.3 EXCLUSIVE EXPERTS

Split $v_u = v_u^{\mathrm{sh}} + v_u^{\mathrm{ex}}$ into experts in $S_c \cap S_u$ and in $S_u \setminus S_c$:

$$v_u^{\mathrm{ex}} = \sum_{j \in S_u \setminus S_c} g_j^u \, E_j(h_u). \tag{18}$$

This extra write is MoE-specific: a dense FFN has $S_c = S_u = \{1\}$, hence $D_{\mathrm{route}} \equiv 0$ and $v_u^{\mathrm{ex}} = 0$. The split does *not* bound $\|(I - P_c)v_u\|$. Shared-set writes $E_i(h_u)$ need not lie in $\mathcal{V}_{S_c}$ (which is spanned by $E_i(h_c)$ and other tokens), and exclusive writes need not be orthogonal to $\mathcal{V}_{S_c}$. Independent routing only makes a residual outside $\mathcal{V}_{S_c}$ typical. Dense CFG can still fail for other reasons; SAGE targets the component of v_u outside $\mathcal{V}_{S_c}$.

A.4 SHARED EXPERT

A shared expert E_0 is evaluated on both branches and does not enter D_{route}. It reduces exclusive routed mass in v_u^{ex}, but it does not put v_u in $\mathcal{V}_{S_c}$: $E_0(h_u)$ still differs from $E_0(h_c)$, and exclusive routed writes remain. Our 1B implementation applies Eq. (9) of the main paper to the full MoE output (shared plus routed). Both experiments minimize the same loss: the toy model has $T = 1$ and uses every available conditional write, so k_{split} is not needed; the 1B model has $n = BT \gg d_h$ and uses k_{split}=2, so only the rank-m section is a proper subspace.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 DPG-BENCH EVALUATION AND CATEGORY BREAKDOWN

We evaluate the Baseline, SAGE (λ=1), and the KL routing constraint (λ_{kl}=1) on DPG-Bench (Hu et al., 2024) at CFG scales (Ho & Salimans, 2022) $s \in \{1, 3, 5, 7, 10\}$, using mPLUG (Li et al., 2022) for visual question answering (VQA). For each prompt, we generate four 320×320 images using 50 sampling steps.

Table 4 reports the five top-level categories: Global, Entity, Attribute, Relation, and Other. SAGE exceeds the Baseline in all five categories at every tested $s \geq 3$. At s=3, the largest gains are in Other (+6.0 pp) and Entity (+4.7 pp); at s=10, the gains reach +10.8 pp for Entity and +8.1 pp for Relation. Thus, the advantage under stronger guidance extends across object presence, attributes, and relations, rather than being confined to one category.

Table 5 gives all 13 second-level categories at s=3 and s=5. SAGE improves over the Baseline in every subcategory at both scales. The gains in counting are +6.4 and +12.4 pp, respectively, while whole-entity presence improves by +5.1 and +8.6 pp. Size, texture, and spatial relations also improve at both scales, showing that the benefits cover multiple aspects of compositional generation.

B.2 GENEVAL PER-TASK BREAKDOWN

Table 6 reports the full per-task breakdown on GenEval (Ghosh et al., 2023) for all three methods at every guidance scale $s \in \{1, 3, 5, 7, 10\}$; the "Overall" column is the mean over the six tasks and reproduces the GenEval rows of table 2 of the main paper. Focusing on the shared operating point s=3, SAGE improves four of the six categories, with the largest gain by far in color-attribute binding (+11.5 pp), followed by two-object composition (+3.5 pp) and single-object presence (+3.4 pp); it shows small decreases in counting (−1.9 pp) and colors (−2.4 pp). The gains therefore concentrate exactly on the compositional categories that depend on precise conditional control, which is where a write outside the conditional subspace is most damaging. Across the whole sweep, SAGE leads the Baseline on the Overall score at every $s \geq 3$, while the KL routing constraint degrades every category at every scale, consistent with the loss of expert specialization discussed in section 4.2.2 of the main paper.

B.3 ADDITIONAL ABLATIONS

The two ablations below complement those reported in section 4.2.3 of the main paper, use the same setup as in section 4.2.1 of the main paper, and are likewise evaluated at guidance scales $s \in \{3, 5\}$.

Table 4: **DPG-Bench category scores (%) across CFG scales.** Category scores aggregate raw per-question answers over all four images per prompt. Overall uses dependency-aware scoring and matches table 2 of the main paper; it is not the category average. Higher is better.

Method	s	Global	Entity	Attribute	Relation	Other	Overall
Baseline	1	70.06	68.04	73.16	84.98	43.00	54.77
	3	73.10	73.87	81.00	85.71	57.40	61.76
	5	65.43	69.66	77.44	82.28	54.60	56.33
	7	61.02	66.10	75.00	79.96	52.00	52.38
	10	59.65	62.59	71.66	77.90	50.60	48.56
SAGE (λ=1)	1	69.45	65.94	71.04	84.91	42.30	52.30
	3	76.44	78.58	83.36	88.65	63.40	67.49
	5	73.40	77.70	81.54	86.95	65.10	65.63
	7	70.44	76.10	80.20	86.60	63.90	63.36
	10	66.49	73.34	77.21	85.96	62.00	59.55
KL (λ_{kl}=1)	1	63.30	55.26	66.73	78.74	29.30	40.95
	3	67.63	64.66	77.45	80.32	48.90	50.65
	5	64.44	61.86	74.45	77.10	45.60	46.82
	7	60.56	59.56	72.40	74.76	44.10	44.38
	10	59.80	56.13	69.33	73.60	42.20	41.08

Table 5: **Fine-grained DPG-Bench scores (%) at $s \in \{3, 5\}$.** All 13 second-level categories use the same four-image, pre-dependency aggregation as table 4. SAGE and KL use λ=1 and λ_{kl}=1, respectively. Higher is better.

		s=3			s=5		
Category	Subcategory	Baseline	SAGE	KL	Baseline	SAGE	KL
Global	–	73.10	76.44	67.63	65.43	73.40	64.44
Entity	Whole	73.89	79.03	63.72	69.43	78.02	60.59
	Part	77.25	80.08	71.58	75.44	81.01	71.09
	State	71.91	75.53	65.63	67.66	74.29	63.20
Attribute	Color	87.90	88.52	84.71	85.32	87.87	82.35
	Shape	78.93	80.57	72.93	79.59	80.57	78.17
	Size	62.71	66.32	55.58	56.20	64.57	52.38
	Texture	77.59	81.61	75.29	73.15	78.60	70.46
	Other	77.47	80.43	72.39	73.00	77.72	69.26
Relation	Spatial	86.10	89.17	80.72	82.82	87.46	77.67
	Non-spatial	79.72	80.66	74.21	74.06	79.25	68.40
Other	Count	53.88	60.25	43.38	49.63	62.00	39.13
	Text	71.50	76.00	71.00	74.50	77.50	71.50

Both the activation-frequency ablation (table 7) and the shared-expert and ProMoE (Wei et al., 2026) comparison (table 8) report DPG-Bench and GenEval scores at each scale.

Activation frequency. We investigate whether SAGE needs to be active at every training step or whether intermittent activation suffices. Because $\mathcal{L}_{\text{sage}}$ is a minibatch subspace constraint rather than a per-token identity, it can be amortized across steps: intermittent application still reshapes how the two branches write. The table shows that activating SAGE every second or every fifth step retains nearly all of the from-scratch gain, which is consistent with a batch-level section rather than a per-token constraint. Table 7 compares activation at every step with activation at every second and every fifth step.

Shared expert as an implicit bridge. Two factors explain why existing MoE-DiTs (Fei et al., 2024) produce acceptable images despite the subspace leak. The first is that moderate CFG hides the problem: at typical operating points (s=3 to 5), the leak is present but not catastrophic, so the

Table 6: **GenEval per-task accuracy (%) across CFG scales.** Per-category GenEval scores for all three methods at every guidance scale s; the "Overall" column is the mean over the six tasks and matches the GenEval rows of table 2 of the main paper.

Method	s	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attr.	Overall
Baseline	1	50.0	18.2	20.3	44.1	25.3	23.8	30.3
	3	79.4	46.0	40.0	79.3	51.5	46.8	57.1
	5	82.5	43.9	40.3	75.8	48.8	41.8	55.5
	7	79.4	43.9	40.3	68.6	49.8	37.3	53.2
	10	73.1	42.4	33.1	58.2	42.5	33.3	47.1
SAGE (λ=1)	1	42.5	17.2	18.1	36.7	20.3	24.0	26.5
	3	82.8	49.5	38.1	76.9	53.3	58.3	59.8
	5	80.0	48.7	45.3	75.3	52.8	54.0	59.3
	7	76.9	50.8	39.4	72.3	53.5	49.5	57.1
	10	74.7	51.3	39.4	63.3	46.3	40.3	52.5
KL (λ_{kl}=1)	1	23.8	6.8	10.6	23.4	8.0	10.3	13.8
	3	60.3	23.5	23.8	55.6	27.3	25.5	36.0
	5	55.0	24.2	27.8	49.2	28.0	23.0	34.5
	7	55.6	20.2	25.9	44.1	29.0	18.5	32.2
	10	56.3	19.2	21.9	39.9	23.8	16.3	29.5

Table 7: **Ablation: SAGE activation frequency ($s \in \{3, 5\}$).** DPG-Bench and GenEval scores for SAGE activation at every step, every second step, and every fifth step.

	s=3		s=5	
Interval	DPG	GenEval	DPG	GenEval
Every step	67.49	59.8	65.63	59.3
Every 2 steps	67.21	59.9	65.84	59.1
Every 5 steps	66.73	59.4	65.12	58.8

degradation without SAGE is tolerable, even though the Baseline still benefits substantially from SAGE at these scales (+5.73 DPG at s=3; table 2 of the main paper); the gap widens dramatically at s=7 to 10, which matters increasingly as applications demand high guidance (double-CFG, style transfer, compositional prompting). The second is that the shared expert is a dense-style path evaluated on both branches, which reduces exclusive routed mass (section A.4) without putting v_u in $\mathcal{V}_{S_c}$. We isolate this effect in table 8 by removing the shared expert entirely and by comparing with ProMoE (which fixes one expert for the unconditional branch).

Table 8: **Ablation: shared expert and ProMoE ($s \in \{3, 5\}$).** Comparison of MoE configurations with and without SAGE against ProMoE at both guidance scales.

	s=3		s=5	
Configuration	DPG	GenEval	DPG	GenEval
8E + 1 shared, no SAGE	61.76	57.1	56.33	55.5
8E + 1 shared, SAGE (λ=1)	67.49	59.8	65.63	59.3
8E, no shared, no SAGE	57.82	53.0	52.41	51.4
8E, no shared, SAGE (λ=1)	64.05	56.4	62.18	55.6
ProMoE	63.12	58.4	57.95	56.8

The results show the expected pattern: removing the shared expert worsens baseline performance (more exclusive routed mass is exposed), while adding SAGE still recovers most of the gap. ProMoE, which reserves one expert for the unconditional branch and thus plays a role analogous to the shared expert without aligning the remaining routed activations, provides partial relief but does not match SAGE.

B.4 SAGE Preserves Routing Diversity

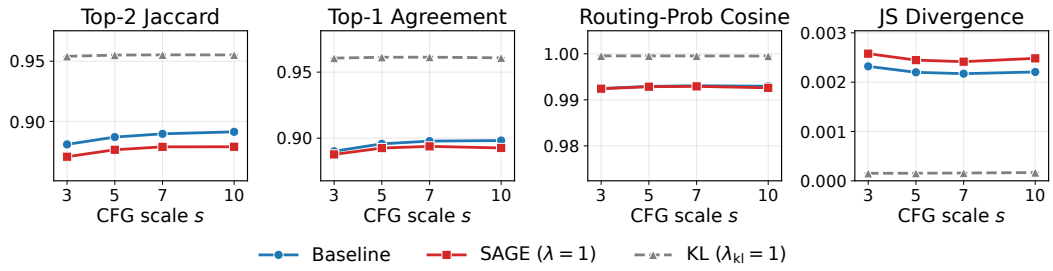


Figure 5: **Conditional vs. unconditional router similarity across CFG scales.** SAGE (red) tracks the Baseline (blue) almost exactly on all four measures, whereas the KL routing constraint (gray, dashed) forces the two branches to agree (higher Jaccard/agreement, near-zero Jensen–Shannon (JS) divergence). Higher values indicate greater similarity for Top-2 Jaccard, Top-1 agreement, and routing-probability cosine similarity; lower values indicate greater similarity for JS divergence. Error bars are standard errors over 500 prompts and are smaller than the markers.

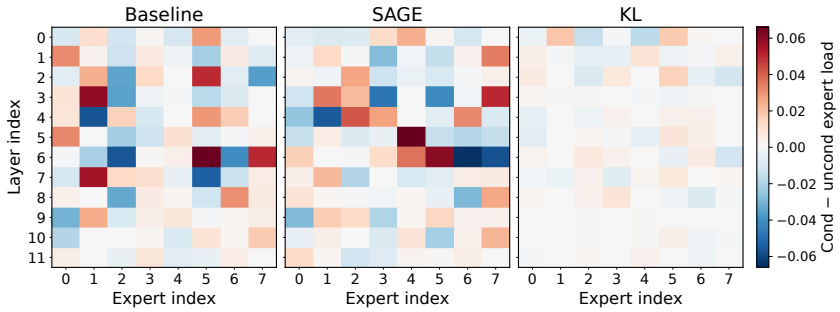


Figure 6: **Conditional minus unconditional expert load per layer and expert at** s=3, averaged over 500 prompts and shown on a shared color scale. Baseline and SAGE exhibit comparably large conditional/unconditional load differences (routing diversity preserved), whereas the KL constraint suppresses them almost entirely (routing forced to agree).

A central design goal of SAGE is to fix the subspace leak *without* touching the router: unlike the KL routing constraint, which explicitly forces the two branches to route alike, SAGE only aligns unconditional activations to the conditional subspace (section 3.2 of the main paper). Here we verify empirically that SAGE indeed leaves routing behavior essentially unchanged. Using the 1B MoE-DiT of section 4.2.1 of the main paper, we run each model on 500 prompts across four guidance scales and record, at every denoising step and layer, the conditional and unconditional Top-2 expert selections and routing distributions. We evaluated various values for λ_{kl} (0.01, 0.1, and 1) and observed consistent results across all settings. Decreasing λ_{kl} did not yield improvements, as the routing health and benchmark scores still fell short of the baseline.

Conditional/unconditional routing similarity is unchanged. Figure 5 reports four similarity measures between the conditional and unconditional branches as a function of the guidance scale. Across all scales, SAGE is statistically indistinguishable from the Baseline on every measure: Top-2 Jaccard, Top-1 agreement, and routing-probability cosine similarity all coincide within noise, and the JS divergence between the two routing distributions is, if anything, marginally *higher* for SAGE than for the Baseline, indicating that SAGE does not push the branches toward common experts. The KL constraint behaves oppositely, driving Jaccard and agreement up toward 0.95 and the JS divergence down toward zero, that is, collapsing the two branches onto nearly identical routes.

Per-expert load divergence is preserved. Figure 6 visualizes the per-layer, per-expert difference between the conditional and unconditional expert loads at s=3, on a shared color scale. The Baseline and SAGE panels are equally vivid, with mean absolute load differences of 0.0144 and 0.0148,

Table 9: **FID across CFG scales.** Lower is better. Δ is Baseline $-$ SAGE, positive values indicate a SAGE improvement.

Method	s=1	s=3	s=5	s=7	s=10
Baseline	13.90	18.54	26.11	33.11	45.38
SAGE (λ=1)	13.18	14.80	20.28	25.54	37.12
Δ (Base $-$ SAGE)	$+0.72$	$+3.74$	$+5.83$	$+7.57$	$+8.26$

respectively, confirming that the two branches continue to recruit visibly different experts under SAGE. By contrast, the KL panel is almost uniformly white (mean absolute difference 0.0034, roughly $4\times$ smaller), showing that routing agreement has been enforced at the cost of expert specialization. Together, the two figures confirm the claim of section 4.2.2 of the main paper: SAGE reduces the residual outside $\mathcal{V}_{S_c}$ while leaving routing diversity intact, which is precisely why it avoids the specialization collapse that makes the KL constraint counterproductive.

B.5 FID Across CFG Scales

To confirm that the CFG-robustness benefit of SAGE also holds for distributional image fidelity, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017) as a function of the guidance scale $s \in \{1, 3, 5, 7, 10\}$.

We sample a fixed set of $10{,}000$ prompts from the in-domain T2I test set and pair each prompt with its ground-truth image to form an in-domain reference set. Reference images are decoded, EXIF-oriented, center-cropped to a square, and bicubic-resized to 320×320 to match the generation resolution; each model then generates one 320×320 image per prompt with 50 sampling steps and a fixed per-prompt seed. FID is computed from standard Inception-V3 features (2048-d `pool3`, bicubic resize to 299), using the identical $10{,}000$ prompt IDs for the reference set and for every model and scale, so all conditions are strictly paired. The Baseline and SAGE checkpoints are trained 200,000 steps.

Table 9 mirrors the trend of the alignment metrics. At s=1 (no guidance) SAGE performs better than baseline (13.90 vs. 13.18). As the guidance scale grows, the Baseline FID degrades steeply ($+31.5$ from s=1 to s=10), whereas SAGE degrades markedly more slowly ($+23.94$); SAGE therefore attains a better FID at every tested scale.

C Training Dynamics and Health

We further compare the optimization dynamics of the Baseline and SAGE over a $200{,}000$-step training schedule. The purpose of this analysis is twofold: to verify that the additional alignment objective does not compromise the primary flow-matching objective or the health of MoE routing, and to confirm that it directly reduces the subspace residual targeted by SAGE.

Figure 7(a) reports the primary flow-matching objective $\mathcal{L}_{\mathrm{FM}}$ defined in Eq. (1) of the main paper. Both methods exhibit the same rapid initial decrease and converge to nearly identical values. Because the two curves overlap at the scale of the full plot, we additionally show a $4{,}000$-step magnified interval. The zoomed view confirms that their differences remain negligible.

Figure 7(b) shows the standard MoE auxiliary loss used to encourage balanced expert utilization. A lower and stable value indicates that routing does not concentrate excessively on a small subset of experts. The two methods follow closely matched trajectories throughout training, with SAGE maintaining a slightly lower auxiliary loss than the Baseline. This indicates that SAGE does not interfere with the existing load-balancing mechanism or induce routing collapse.

Dead expert ratio is measured in Figure 7(c). Let E denote the number of experts, N the total number of routed tokens in the current batch, and C_e the number of tokens assigned to expert e. We classify expert e as dead when its load is below 10% of the uniform expected load N/E, and

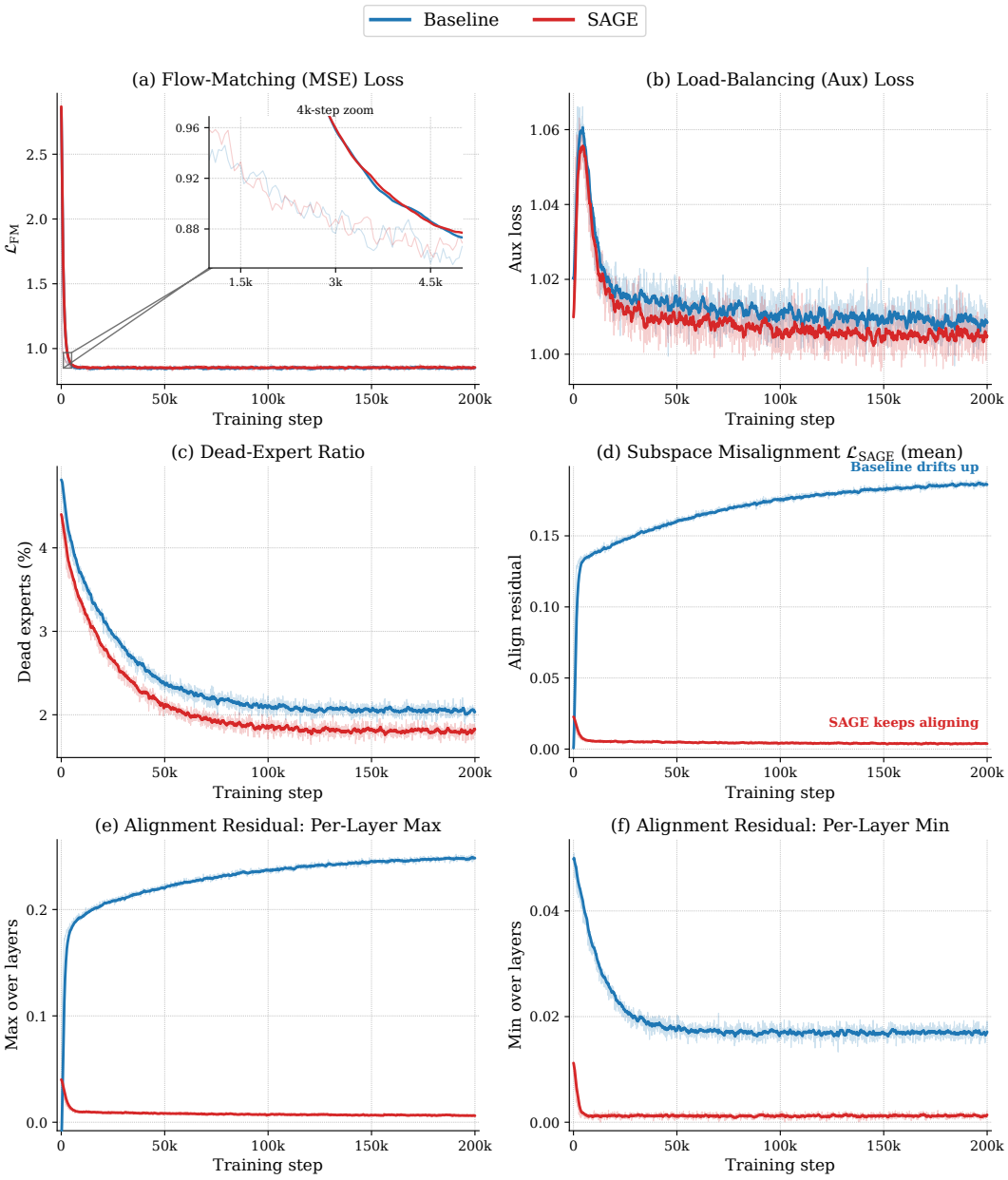


Figure 7: **Training dynamics of the Baseline and SAGE. (a)** Flow-matching MSE loss; the inset magnifies a 4,000-step interval and shows that the two runs remain closely matched. **(b)** MoE load-balancing auxiliary loss. **(c)** Percentage of dead experts. **(d–f)** Mean, per-layer maximum, and per-layer minimum of the subspace residual, respectively. SAGE preserves the primary optimization and routing-health statistics while consistently reducing the alignment residual.

compute

$$\text{dead_expert_pct} = \frac{1}{E}\sum_{e=1}^{E}\mathbb{I}\left(C_e < 0.1\,\frac{N}{E}\right), \tag{19}$$

where $\mathbb{I}(\cdot)$ is the indicator function. For both methods, the dead-expert percentage remains below 5% and decreases as training proceeds. SAGE closely tracks the Baseline and does not increase the number of under-utilized experts, further confirming that its alignment constraint does not waste MoE capacity.

Figure 7(d–f) report three complementary summaries of the layer-wise alignment residual: its mean across MoE layers, its maximum, and its minimum. The Baseline residual grows over training in

both the mean and worst-layer views, indicating that unconditional writes drift outside the conditional subspace $\mathcal{V}_{S_c}$ as experts specialize. In contrast, SAGE rapidly decreases the residual and keeps its mean, maximum, and minimum close to zero throughout training. The per-layer maximum is especially informative because it rules out the possibility that a small set of poorly aligned layers is hidden by averaging.