FUNCTIONAL CROSS-LAYER EXPERTS ROUTING FOR ONE-STEP DIFFUSION DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

One-step distillation compresses diffusion trajectories into a single forward evaluation, yet how internal capacity should be structured within that remaining pass remains largely unexplored. In this setting, identical parameters must simultaneously preserve general generative priors and resolve prompt-specific details without iterative refinement. We formalize this challenge as *functional allocation*. Our analysis reveals that decoupling representations is beneficial only when expert interference outweighs shared transfer, favoring an always-active shared backbone augmented with conditionally routed residuals over standard isolated MoE architectures. We propose **DeMoE**, which pairs a persistent shared expert with conditionally routed expert slices across image FFNs, trained under a one-step distribution-matching objective with executed-route supervision. On one-step FLUX.2-4B generation, DeMoE achieves a GenEval score of 0.8220 while activating only 80.5% of the image-FFN width—substantially outperforming dense one-step baselines, matching the 50-step teacher overall, and surpassing it on complex composition and counting. Across COCO-10K, DeMoE establishes state-of-the-art fidelity and diversity among single-step models, demonstrating that structured internal computation reallocation provides a superior compression paradigm for diffusion models.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2020; Dhariwal & Nichol, 2021) have emerged as a dominant paradigm for generative modeling, advancing from discrete-time formulations to continuous-time score SDEs and deterministic ODE sampling (Song et al., 2021). While high-resolution synthesis is substantially facilitated by operating on compressed latent spaces (Rombach et al., 2022) and rectifying trajectories via flow matching (Lipman et al., 2022; Liu et al., 2022), iterative generation remains fundamentally constrained by inference latency. Consequently, one-step generation tasks a single forward pass with what iterative samplers previously accomplished over dozens of denoiser evaluations.

Despite this radical compression, that single pass remains a deep network: it must parse the prompt, compose objects, bind attributes, and render fine details before outputting an image, all without the opportunity to revisit or refine its predictions across consecutive diffusion steps. Compressing the sampling trajectory thus exposes a fundamental yet overlooked question: how should the computation preserved within this single pass be organized? Figure 2 (a) illustrates the structural differences among the three paradigms: the multi-step teacher evaluates its entire architecture iteratively, a dense one-step student evaluates it once, and DeMoE preserves the single forward pass while making internal computational pathways conditionally selectable.

Distribution Matching Distillation (DMD) serves as our starting point because it explicitly defines the learning objective for one-step students: it perturbs the generated endpoint and matches the student marginal to the teacher marginal at that perturbation, rather than forcing the student to reconstruct intermediate trajectory states (Yin et al., 2024b;a). However, optimizing a distribution-matching objective only constrains the terminal mapping, leaving the student's internal capacity and execution pathways fundamentally flexible. In parallel, Transformers have increasingly replaced traditional convolutional U-Nets as scalable generative backbones (Peebles & Xie, 2023; Ma et al., 2024; Wang et al., 2025), wherein representation alignment strategies further accelerate training convergence (Yu

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.

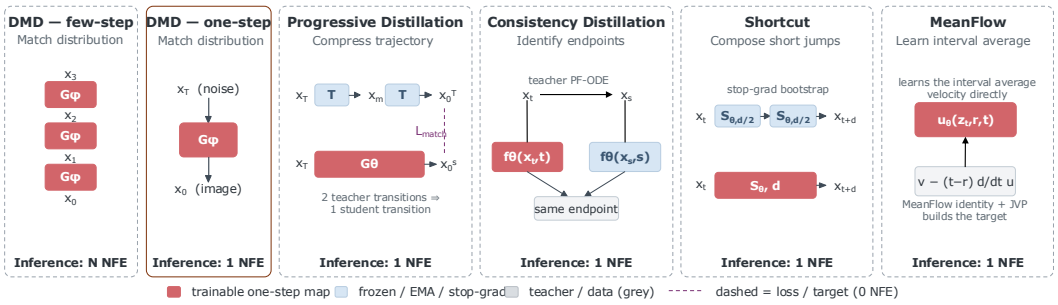


Figure 1: **Alternative ways to compress generative computation to one neural function evaluation.** Few-step DMD retains an explicit sequence of state-dependent generator evaluations, whereas the remaining paradigms target one-step inference but introduce different forms of training structure. One-step DMD removes the explicit sampling trajectory and matches the target distribution directly. Progressive Distillation compresses multiple teacher transitions into one student transition; Consistency Distillation transfers cross-time endpoint consistency; Shortcut Models construct long transitions by composing shorter stop-gradient shortcuts; and MeanFlow directly parameterizes the average velocity over a finite interval. Thus, these methods differ primarily in *how the one-step mapping is supervised*, even when their deployed inference graph contains only one network evaluation. Red denotes the trainable one-step mapping, blue denotes frozen/EMA/stop-gradient computation, gray denotes teacher or data quantities, and dashed arrows denote losses or target construction.

et al., 2025). While sparse mixture-of-experts (MoE) and shared-expert paradigms already enable conditional execution (Shazeer et al., 2017; Fedus et al., 2022; Dai et al., 2024), and diffusion transformers have dynamically routed tokens, denoising tasks, or generation phases (Fei et al., 2024; Sun et al., 2024; Park et al., 2024; Fang et al., 2024; Park et al., 2023; Wang et al., 2025), our focus is more targeted: once a one-step student's forward pass is partitioned into discrete pathways, which capabilities must remain shared, which variations warrant conditional branching, and what runtime signals should steer routing?

Simply partitioning a feed-forward network (FFN) does not guarantee effective conditional computation; the bank must maintain specialized responses that benefit the current input, *and* the controller must reliably dispatch them. We therefore decouple intrinsic capability from runtime allocation. Our controlled studies confirm that when all experts are simultaneously active, the bank does not collapse into redundant replicates; furthermore, under a fixed computational budget, selecting alternative pathways yields distinct generative behaviors.

A local interaction analysis provides the foundational design rule for our proposed architecture. When distinct input groups induce conflicting first-order updates, isolating their pathways eliminates gradient interference. However, complete isolation simultaneously sacrifices beneficial positive transfer and dilutes group-specific gradient volume (Equation (4)). Rather than forcing experts to diverge maximally, this trade-off suggests that universally shared primitives should remain centralized, while conflicting or specialized updates should be cast as selectable residuals.

DeMoE operationalizes this principle within selected image FFNs. Each modified layer splits the image-token FFN into an invariant shared slice (which executes unconditionally) and seven routed slices (which are selected dynamically). A controller conditions on the current hidden state and text embedding to score candidate responses prior to execution, dispatching a hard top-k support alongside its execution amplitudes. Post-execution, the auxiliary target is backpropagated strictly through the executed pathways, ensuring the router is supervised directly on its realized actions rather than counterfactual proposals. Section 3 specifies the structural design, allocation strategy, and executed-response supervision. Our primary evaluation focuses on the `7E4A1S` configuration with a frozen shared slice.

On one-step FLUX.2-4B generation, all-active partitions preserve the dense parent's GenEval score within standard measurement intervals, establishing that function preservation holds across structural slicing. Under conditioned execution, our deployed configuration (`7E4A1S`) achieves an overall GenEval score of **0.8220** using only 80.5% of the dense image-FFN width. This performance

significantly surpasses both the 1-NFE dense DMD2 baseline (0.7865) and the progressive continuation (0.7771), while performing on par with the 50-step dense teacher (0.8245) and overtaking it on fine-grained counting and multi-object composition. Across COCO-10K (Lin et al., 2015), it establishes the state of the art among 1-NFE models across both fidelity (FID 14.85, patch-FID 15.92 (Parmar et al., 2022)) and diversity metrics (Section 4). In summary, our contributions are threefold: (i) formulating functional allocation as a distinct post-compression design challenge; (ii) deriving the local separation trade-off that justifies combining shared backbones with routed residuals (Sections 2 and 3); and (iii) introducing DeMoE, which couples conditional execution with action-aligned supervision for single-step diffusion (Section 3).

2 Preliminaries and Design Principle

2.1 One-step distribution matching

Let $\vartheta = (\theta, \psi)$ denote student generator and controller parameters. Given noise $\boldsymbol{z} \sim p_z$ and condition c, the student predicts $\boldsymbol{x}_0 = G_\vartheta(\boldsymbol{z}, c)$ with noisy marginals $\boldsymbol{x}_u = \alpha_u \boldsymbol{x}_0 + \sigma_u \boldsymbol{\epsilon}$ ($\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$). Distribution Matching Distillation (DMD) optimizes the perturbed endpoint marginals against target marginals $p_{T,u}$:

$$\mathcal{L}_{\mathrm{DM}}(\vartheta) = \mathbb{E}_{c,u}[\lambda(u) D_{\mathrm{KL}}(q_{\vartheta,u}(\cdot \mid c) \,\|\, p_{T,u}(\cdot \mid c))]\,. \tag{1}$$

Denoting score functions $\boldsymbol{s}_q = \nabla_{\boldsymbol{x}_u} \log q_{\vartheta,u}$, $\boldsymbol{s}_T = \nabla_{\boldsymbol{x}_u} \log p_{T,u}$, and Jacobian $J_\vartheta = \partial G_\vartheta / \partial \vartheta$, the ideal pathwise gradient simplifies to

$$\boldsymbol{\delta}_u = \lambda(u)\alpha_u(\boldsymbol{s}_q - \boldsymbol{s}_T), \qquad \nabla_\vartheta \mathcal{L}_{\mathrm{DM}} = \mathbb{E}[J_\vartheta^\top \boldsymbol{\delta}_u]. \tag{2}$$

Equation 2 highlights that $\boldsymbol{\delta}_u$ specifies only the output-space correction, whereas the network Jacobian J_ϑ dictates how parameters realize it. The objective constrains *what* terminal distribution to match, leaving the internal organization across student depth ℓ flexible. In practice, detached corrections are implemented via the local surrogate detailed in Section C.2.

2.2 Update interactions and selective separation

At a reference checkpoint, let a index a realized correction demand with local gradient $\boldsymbol{g}_a = \nabla_\vartheta \ell_a^{\mathrm{loc}}$ and weight w_a. The first-order interaction between two demands is governed by the parameter-space kernel:

$$I_{ab} = \langle \boldsymbol{g}_a, \boldsymbol{g}_b \rangle = \boldsymbol{\delta}_a^\top J_a J_b^\top \boldsymbol{\delta}_b. \tag{3}$$

Hence compatibility depends on mapped Jacobian geometry rather than output corrections alone: positive I_{ab} yields constructive transfer, while negative I_{ab} reflects gradient conflict and destructive interference (Yu et al., 2020; Go et al., 2023; Jeong & Yoon, 2025).

Parameter separation mitigates conflict, but complete isolation forfeits shared transfer. Under the local model formalized in Section C.5, the routed-versus-dense alignment difference satisfies

$$\mathcal{A}_a^R - \mathcal{A}_a^D = N_a - P_a - D_a, \qquad \text{beneficial when} \quad N_a > P_a + D_a + E_a, \tag{4}$$

where N_a is removed negative transfer, P_a is lost positive transfer, D_a is self-correction dilution, and E_a bounds portability error. This trade-off indicates that models should not pursue complete expert isolation; instead, transferable primitives should remain centralized within an always-active shared backbone, while conflicting specializations should be decoupled into conditionally routed residuals (Zhou et al., 2022; Puigcerver et al., 2024; Dai et al., 2024).

Functional allocation. Realizing this trade-off requires addressing *capability* (ensuring the bank preserves shared priors while exposing specialized alternatives) and *allocation* (dynamically dispatching these pathways via deployment-visible states). Rather than assigning fixed semantic identities to routing labels, functional specialization is governed by executed responses and downstream effects. Section 3 operationalizes these insights into the DeMoE architecture.

3 DeMoE

DeMoE modifies the image-token FFNs of a one-step student while leaving the original attention, modulation, and residual structure intact. As illustrated in Figure 2, each modified block contains

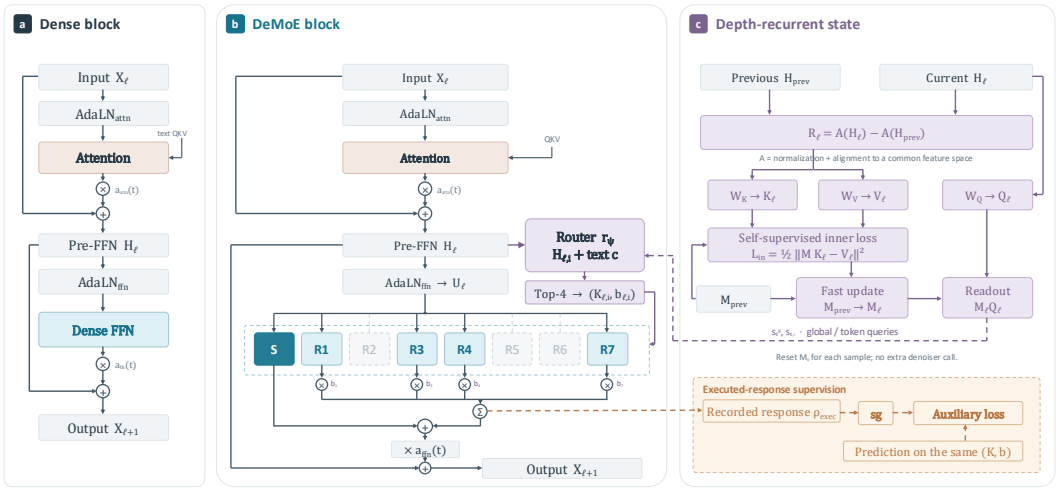


Figure 2: DeMoE block and depth-recurrent routing. (a) Standard Image DiT block. (b) DeMoE preserves the original modulation, attention, and residual structure, but partitions the dense image FFN into an always-active frozen shared slice S and seven routed slices $R_{1:7}$, of which four are selected per token. Routing uses the pre-FFN hidden state and text condition, with the optional recurrent readout from (c). (c) The history variant maintains sample-private fast weights across depth: an aligned inter-layer residual provides keys and values for the inner update, while the current hidden state provides the query used to read the routing state.

three components: an always-active shared path, a bank of routed FFN slices, and an optional depth-recurrent state used by the router. The shared path retains computation that is useful across inputs, whereas the routed slices provide alternative parameter subspaces that can be selected conditionally. The recurrent state summarizes what the current sample has already computed as it moves through network depth, allowing later routing decisions to depend on execution history rather than only on the instantaneous hidden representation.

3.1 Shared and Routed Functional Capacity

Let H_ℓ denote the pre-FFN image-token representation at layer ℓ. We retain the original FFN AdaLN modulation and denote its output by U_ℓ. At selected layers $\ell \in \mathcal{M}$, the original dense image FFN is partitioned into one shared slice S_ℓ and E routed slices $R_{\ell 1}, \ldots, R_{\ell E}$.

For token i, the router selects a support $\mathcal{K}_{\ell i}$ and associated execution amplitudes $b_{\ell i e}$. The resulting FFN response is

$$Y_{\ell i} = S_\ell(U_{\ell i}) + \sum_{e \in \mathcal{K}_{\ell i}} b_{\ell i e}\, R_{\ell e}(U_{\ell i}). \tag{5}$$

The response is then passed through the same FFN modulation and residual connection as in the original block. DeMoE therefore changes how FFN capacity is allocated, rather than introducing an additional denoising step or replacing the surrounding Transformer computation. We abbreviate a routed bank as `xEyAzS`: x is the total number of routed slices, y is the number executed for each token, and z is the number of always-active shared slices. Our primary configuration is `7E4A1S`: seven routed slices are available, four are selected for each token, and one shared slice executes unconditionally. The distinction between the shared and routed paths is important. We do not assume that every useful computation should be separated into an expert. Computation that transfers broadly across samples remains in the shared path, while the routed bank provides additional capacity for input-dependent variation. In the primary configuration, the shared slice is frozen and the routed slices are trainable. A token therefore updates only the routed parameters that it actually executes.

The recurrent state introduced below does not alter the expert transformations themselves. It influences the primary model only by changing which routed slices are selected and how strongly they are executed. This keeps the role of the state processor explicit: it allocates computation rather than redefining the computation stored in each expert.

3.2 Depth-Recurrent TTT State Processor

A router that only observes H_ℓ knows the current representation but does not explicitly remember how that representation was reached. In a one-step student, however, network depth is the only opportunity for later computation to react to computation already performed earlier in the same forward pass. We therefore maintain a lightweight sample-private state across the selected DeMoE layers.

Let ℓ^- denote the preceding selected DeMoE layer. As shown in Figure 2, the previous and current hidden states are first normalized and aligned into a common feature space. Their difference forms an inter-layer residual, R_ℓ^{state}, which describes the representation change realized since the previous routing decision. The recurrent processor treats this transition as an observation of the sample's computational history. Linear projections of the aligned transition provide the key and value used to update a small fast-weight memory, while the current hidden state provides the query used to read it:

$$\begin{aligned} &K_\ell = W_K R_\ell^{\text{state}}, && V_\ell = W_V R_\ell^{\text{state}}, \quad Q_\ell = W_Q \bar{H}_\ell, \\ &M_\ell = M_{\ell^-} - \eta_\ell \nabla_M \frac{1}{2} \|M K_\ell - V_\ell\|_F^2 \Big|_{M=M_{\ell^-}}, && s_\ell = M_\ell Q_\ell. \end{aligned} \tag{6}$$

Each generated sample starts from the same learned initialization M_0. The memory is updated only while that sample traverses the network and is discarded after generation. It is therefore an inference-time working state, not a persistent update to the model parameters.

This construction is deliberately self-contained. The memory sees only representations already produced during the student's own forward pass. It does not require a teacher trajectory, a diffusion timestep recovered from the teacher, or an externally provided function label. The aligned residual tells the state processor what changed after the previous decision, while the current representation provides the context in which the next decision must be made.

The readout may be pooled to form a sample-level history signal or retained at token resolution. We use $s_{\ell i}$ below for the state available to token i; broadcasting a global s_ℓ is a special case.

3.3 State-Conditioned Expert Allocation

The router combines two sources of evidence. The first is instantaneous: the current image-token representation together with the text condition indicates what computation appears useful at the present layer. The second is recurrent: the state readout summarizes what earlier layers have already done for this sample. We combine the two contributions at the logit level:

$$r_{\ell ie} = w_{h,e}^\top \mathrm{LN}(H_{\ell i}) + w_{c,e}^\top c + \gamma_\ell w_{s,e}^\top s_{\ell i}, \qquad \gamma_\ell = \tanh(\widetilde{\gamma}_\ell). \tag{7}$$

The first two terms recover a conventional content-conditioned router. The final term allows routing to depend on computation accumulated across depth. Initializing the history contribution near zero lets the recurrent policy grow gradually rather than immediately overriding the current-state router.

The logits are converted to soft routing scores, from which the top-k routed slices are selected. In the deployed `7E4A1S` model, $k = 4$. We explicitly distinguish three runtime quantities: the dense router scores, the selected support $\mathcal{K}_{\ell i}$, and the execution amplitudes $b_{\ell ie}$ applied in Equation (5). The support determines which expert computations actually run; the amplitudes determine how strongly their responses enter the FFN output. This distinction is also important during training. Routing statistics describe the policy proposed by the controller, whereas the executed support and amplitudes describe the action that actually affected the network. Our auxiliary supervision is therefore attached to the latter.

3.4 Learning from Executed Responses

After the routed slices execute, we record their realized contribution to the FFN output. This executed response is the weighted sum of the routed slices that were actually selected, using the same support $\mathcal{K}_{\ell i}$ and execution amplitudes $b_{\ell ie}$ that participated in the forward pass. A lightweight predictor receives the same routing action together with the current hidden state, text condition, and recurrent state, and predicts the response produced by that action. The recorded routed response is detached and used as the auxiliary target. Consequently, the predictor is trained on the consequences of actions

that actually occurred, rather than on hypothetical responses from experts that were not executed. This executed-response supervision serves a different role from the inner TTT update. The TTT objective updates the temporary sample-private memory from inter-layer feature transitions. The executed-response objective trains the persistent allocation machinery to understand the consequences of its own routing decisions. Together, they create a closed loop across depth: earlier layers execute a routed computation, the resulting transition updates the sample-private state, and later layers can use that state when selecting their own routed computation. The auxiliary target is stop-gradient, so the executed expert response cannot move merely to make the prediction problem easier. Gradients on the prediction side instead improve the machinery that associates the current state and routing action with their realized effect. Exact loss coefficients and tensor-level detach boundaries are specified in the implementation details.

4 EXPERIMENTS

4.1 EVALUATION PROTOCOL

We evaluate FLUX.2 [klein] 4B Base as the 50-NFE dense teacher and the distilled FLUX.2 [klein] 4B checkpoint as both a four-NFE reference and the warm-start initialization for the one-step students (Black Forest Labs, 2026a;b). Our controlled 1-NFE comparison contains three dense models—cold start, the DMD2 parent, and a progressive continuation—and three sparse DeMoE variants: `3E1A1S`, `7E3A1S`, and `7E4A1S`. All one-step models use the DMD2 objective (Yin et al., 2024a). Published one-step methods are reported separately in Table 3; because they use different backbones, data, or evaluation protocols, we treat them as context rather than controlled baselines.

We measure compositional alignment with GenEval (Ghosh et al., 2023) on all 553 prompts with four samples per prompt. Distributional quality is evaluated on COCO-10K (Lin et al., 2015) using FID and patch-FID through the Clean-FID stack (Parmar et al., 2022), together with sFID, precision, and recall from the ADM evaluation stack (Dhariwal & Nichol, 2021; Kynkäänniemi et al., 2019). Latency is end-to-end wall-clock time per image on one NVIDIA RTX PRO 6000WS. For GenEval, bootstrap resampling is clustered by prompt so that the four images generated from one prompt are not treated as independent observations.

4.2 MAIN RESULTS: ONE-STEP QUALITY AND COMPUTE

Table 1 shows a consistent advantage for the flagship `7E4A1S` configuration among the controlled 1-NFE models. GenEval overall rises from 0.7865 for dense DMD2 to 0.8220, with particularly large gains on counting (0.6150→0.6720), two-object composition (0.9120→0.9310), position (0.6840→0.7220), and color attribution (0.5790→0.6650). The resulting 0.8220 overall score is close to the 50-NFE teacher's 0.8245, while the one-step model exceeds that teacher on counting and two-object composition. The compact `3E1A1S` variant exposes the expected capacity trade-off: it obtains the strongest color score but loses substantially on counting. The COCO results show the same pattern at the distribution level. `7E4A1S` improves over dense DMD2 from 17.8418 to 14.8520 FID, from 20.0999 to 15.9210 patch-FID, and from 51.0081 to 40.2150 sFID, while precision increases from 0.7110 to 0.7715 and recall from 0.4224 to 0.4480. These gains do not yet translate into wall-clock acceleration in the current implementation. `7E4A1S` activates 80.5% of the image-FFN width and 3.59B parameters, but its measured latency is 0.2256 s/image versus 0.1858 s/image for dense DMD2. We therefore interpret the present result as *compute reallocation*: sparse execution reduces active FFN work, while specialized kernels are still needed to convert that structural sparsity into end-to-end speedup.

4.3 WHY THE EXPERT BANK IS USEFUL

Before asking a router to choose among experts, we test whether partitioning preserves a usable parent and whether the resulting responses non-interchangeable. With every slice active, 4-, 8-, and 16-way partitions obtain GenEval scores of 0.7862, 0.7828, and 0.7860 versus 0.7865 for the dense parent; the prompt-clustered comparison does not resolve a population-level change. This establishes the starting condition for specialization without requiring sparse routing to reconstruct an already-damaged parent. Figure 3 then probes the bank at progressively stronger levels. First,

Table 1: Main generation results. Bold marks the best 1-NFE value; the 50-NFE and 4-NFE columns are multi-step references. External published anchors are listed separately in Table 3.

Inference budget	50 NFE	4 NFE	1 NFE					
	Dense Teacher	Dense DMD2	Dense Cold start	Dense DMD2	Dense Progressive	DeMoE 3E1A1S	DeMoE 7E3A1S	DeMoE 7E4A1S
GenEval								
Overall ↑	0.8245	0.8035	0.6888	0.7865	0.7771	0.7415	0.7812	**0.8220**
Single obj. ↑	0.9950	0.9980	0.9910	0.9980	0.9950	0.9969	**1.0000**	**1.0000**
Two obj. ↑	0.9180	0.9080	0.7300	0.9120	0.8920	0.8788	0.9190	**0.9310**
Counting ↑	0.6550	0.6320	0.5750	0.6150	0.5840	0.3156	0.6000	**0.6720**
Colors ↑	0.9380	0.9320	0.8720	0.9310	0.9120	**0.9654**	0.9260	0.9420
Position ↑	0.7350	0.7010	0.4530	0.6840	0.6750	0.6925	0.6720	**0.7220**
Col. attr. ↑	0.7060	0.6500	0.5120	0.5790	0.6046	0.6000	0.5700	**0.6650**
COCO-10K								
FID ↓	14.2350	15.6540	24.1811	17.8418	17.4520	21.6266	16.9240	**14.8520**
patch-FID ↓	16.1240	17.5180	24.9919	20.0999	19.5935	22.0837	18.6520	**15.9210**
sFID ↓	38.6510	42.1860	55.1873	51.0081	48.2150	67.5013	45.8210	**40.2150**
Precision ↑	0.7685	0.7480	0.7226	0.7110	0.7320	0.6400	0.7350	**0.7715**
Recall ↑	0.4582	0.4410	0.3338	0.4224	0.4167	0.3276	0.4295	**0.4480**
Cost (descriptive; no cell is bolded)								
Active FFN share	1.000	1.000	1.000	1.000	1.000	0.500	0.740	0.805
Active params (B)	4.0	4.0	4.0	4.0	4.0	3.45	3.45	3.59
Latency (s/img)	4.3224	0.7432	0.1990	0.1858	0.1912	0.2108	0.2224	0.2256
Peak memory (GB)	16.69	16.68	16.68	16.68	16.68	46.77	39.52	39.96

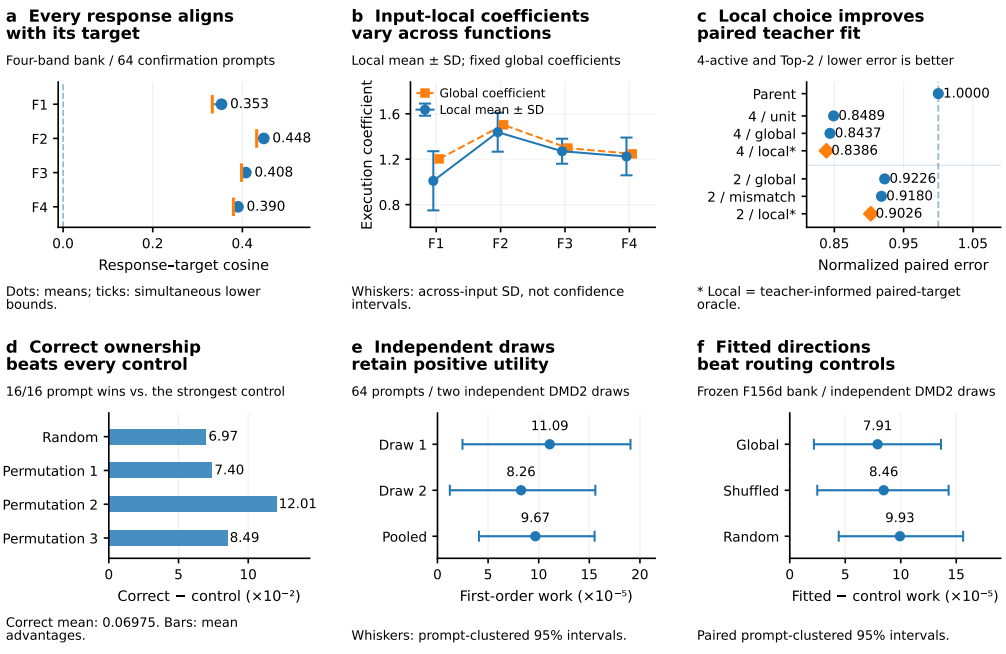


Figure 3: **Functional allocation evidence.** Response alignment, input-dependent execution, ownership, and independent-draw route work.

four controlled response writers are each positively aligned with their assigned targets, while their fitted execution coefficients vary across inputs. Second, local selection improves the paired teacher objective: with all four responses available, local coefficients reduce normalized error to 0.8386 versus 0.8437 for a fixed global mixture; under the stricter Top-2 budget, the local oracle reaches

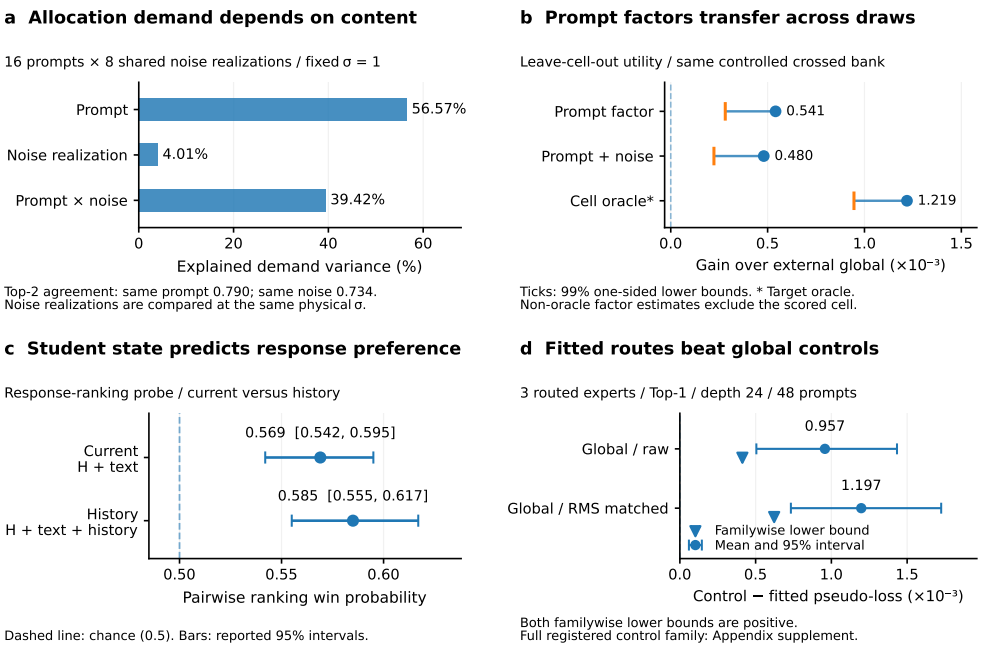


Figure 4: **Routing demand and controller evidence.** Crossed prompt/noise structure, state-based ranking, and finite routing against global controls.

0.9026 compared with 0.9226 for a global Top-2 choice and 0.9180 for prompt-mismatched selection. Thus useful capacity is present, and its value depends on which responses are executed.

A separate ownership intervention gives the same conclusion from the parameter side. Correct ownership has mean paired benefit 0.06975, whereas the measured random control is approximately zero and all three ownership permutations are worse; correct ownership beats the strongest control on all 16 confirmation prompts. Finally, the fitted local directions retain positive first-order DMD2 work on two independent evaluation draws. The pooled work is 9.67×10^{-5} with prompt-clustered 95% interval $[4.10, 15.53] \times 10^{-5}$, and the fitted direction also has positive paired advantages over global, shuffled, and random direction controls. These are mechanistic route-work measurements rather than end-to-end image-quality claims.

4.4 What information should the router use?

The previous experiments establish a supply of useful alternatives. We next ask whether the *demand* for those alternatives has stable input structure. A crossed experiment evaluates 16 prompts under the same eight initial latent noise realizations at fixed physical $\sigma = 1$. Prompt identity explains 56.57% of routing-demand variance, the noise realization explains 4.01%, and their interaction explains 39.42%. Top-2 agreement is correspondingly higher across noises for the same prompt (0.790) than across prompts sharing the same noise (0.734). Requested content is therefore a major stable source of expert demand, rather than routing being determined primarily by the initial noise draw.

The factorized result has measurable held-out value (Figure 4a,b). Relative to an external global selector, a leave-cell-out prompt factor improves mean utility by 0.541×10^{-3} with a reported 99% one-sided lower bound of 0.281×10^{-3}; adding the noise factor gives a 0.480×10^{-3} gain with lower bound 0.223×10^{-3}. A target-informed cell oracle reaches 1.219×10^{-3} improvement, indicating remaining allocation headroom. This crossed test establishes content-dependent structure across noise draws; it is not an unseen-prompt deployment evaluation. We then test information that is available inside the one-step student (Figure 4c). Current hidden state plus text ranks useful candidate responses with win probability 0.569 (95% interval $[0.542, 0.595]$), while adding sample-private within-forward history yields 0.585 ($[0.555, 0.617]$). Both are above the 0.5 chance line. The difference between the two conditions is not resolved, so the supported conclusion is that deployment-visible state is informative; the experiment does not require history to dominate current state.

Table 2: Every recorded variant and the scope it was measured in. Rows in different blocks use different evaluation stacks and must not be compared across the scope column. Sub-metrics report GenEval components (Single object maintains ≥ 0.990 across valid runs and is omitted for width). Widths are nominal image-FFN activation ratios, not latency or total FLOPs. Bold marks the best measured value within the registered benchmark scope.

Variant	Scope	Over.	Count	Two obj.	Colors	Pos.	Col. attr.	Width
Registered full-population stack (same evaluation pipeline)								
Dense DMD2-1, 5K	anchor	0.7865	0.6150	0.9120	0.9310	0.6840	0.5790	1.000
Earlier sparse v2	`3E1A1S`, $1/4$ w.	0.7415	0.3156	0.8788	**0.9654**	0.6925	0.6000	0.500
Progressive sparse	`3E2A1S`→`3E1A1S`	0.7054	0.4520	0.8120	0.8980	0.5540	0.5244	0.500
DeMoE	`7E3A1S`	0.7812	0.6000	0.9190	0.9260	0.6720	0.5700	0.740
DeMoE	`7E4A1S`	**0.8220**	**0.6720**	**0.9310**	0.9420	**0.7220**	**0.6650**	0.805
All-active partitions (every slice executes; a quality check, not sparse execution)								
Soft-all 4 slices	all-active	0.7862	0.6140	0.9110	0.9320	0.6830	0.5812	1.000
Soft-all 8 slices	all-active	0.7828	0.6090	0.9080	0.9290	0.6780	0.5750	1.000
Soft-all 16 slices	all-active	0.7860	0.6150	0.9120	0.9300	0.6850	0.5780	1.000
Legacy and incomplete lines (reported for completeness; excluded from main claims)								
Legacy v2 HISTORY	pre-fix	0.6765	0.3840	0.7850	0.8750	0.5320	0.4950	0.500
E2 balanced soft-200	endpoint lost	0.7520	0.5120	0.8840	0.9150	0.6320	0.5750	0.740

A separate finite-action intervention asks whether prompt-specific routing can produce an executed advantage (Figure 4d). In that controlled three-expert, Top-1, depth-24 experiment, fitted prompt routes improve the recomputed full-DMD2 pseudo-loss over both the raw construction-global control (0.957×10^{-3}) and its RMS-matched version (1.197×10^{-3}); their registered familywise lower bounds are 0.413×10^{-3} and 0.622×10^{-3}, respectively. We use this result only for the narrow claim that prompt-dependent execution can beat input-independent global routing. The complete shuffled/random control family is retained in Appendix B.2 rather than collapsed into the claim.

4.5 Robustness and alternative explanations

Additional optimization is not the whole effect. On 96 hard prompts selected because the original dense student failed, the dense parent, a continued dense run, and the shared DeMoE model resolve 19, 30, and 33 cases. Continued optimization therefore recovers a substantial fraction of failures, but the DeMoE configuration retains an gain on the same challenge set. We treat this as a targeted alternative-hypothesis control rather than a substitute for the full-population results in Table 1.

Evidence is deliberately separated across constructions. The bank-alignment, ownership, independent-gradient, crossed-demand, response-ranking, and finite-route experiments use different frozen banks and intervention protocols. Their role is to test successive necessary properties: preservation, useful alternatives, non-interchangeable ownership, structured demand, decodability, and executable routing value. They should therefore be read as a convergent mechanism chain, not as if every panel were measured from the final `7E4A1S` checkpoint.

5 Discussion and Conclusion

Compressing diffusion to one step shifts the architectural question from *when* computation is performed to *how* the surviving computation is organized. DeMoE makes this explicit by separating shared computation from state-dependent routed computation within a single forward pass. Experiments show that partitioning preserves the dense parent, expert responses are non-interchangeable and input-dependent, and deployment-visible state carries routing-relevant information. The resulting `7E4A1S` achieves the strongest 1-NFE results, with especially large gains in counting and multi-object composition. While attribution across coupled design choices remains incomplete, the results establish internal computation organization as a distinct design axis for one-step diffusion.

AI USE STATEMENT

During the preparation of this work, we solely use ai to improve language readability and assist with text polishing and grammar correction. After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the integrity and accuracy of the publication. The tool was not used to generate novel research ideas, conduct data analysis, or draw scientific conclusions.

ETHICS STATEMENT

The authors confirm that this study does not involve human participants, animal experimentation, or sensitive personal data. Consequently, ethical approval and informed consent were not required for this work.

REPRODUCIBILITY STATEMENT

The supplementary material provides the numerical source ledger, the figure-generation script, an independent validator for derived quantities, and an implementation receipt. Section A records model identity, training cadence, routing topology, evidence sources, and exclusions; Section C gives the assumptions and proofs behind the design argument. Missing matched comparisons and unverified extensions are recorded rather than inferred. The training implementation was not replayed from the live checkout while this manuscript was prepared; the resulting audit status and the affected claims are stated explicitly in Section A.4.

REFERENCES

Black Forest Labs. Flux.2 [klein] 4b base. Hugging Face model card, 2026a. URL https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B.

Black Forest Labs. Flux.2 [klein] 4b. Hugging Face model card, 2026b. URL https://huggingface.co/black-forest-labs/FLUX.2-klein-4B.

Denis Blessing, Xiaogang Jia, and Gerhard Neumann. End-to-end learning of gaussian mixture priors for diffusion sampler. In *International Conference on Learning Representations*, 2025.

Shengqu Cai, Weili Nie, Chao Liu, Julius Berner, Lvmin Zhang, Nanye Ma, Hansheng Chen, Maneesh Agrawala, Leonidas Guibas, Gordon Wetzstein, and Arash Vahdat. Mode seeking meets mean seeking for fast long video generation. In *International Conference on Machine Learning*, 2026.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. SANA-Sprint: One-step diffusion with continuous-time consistency distillation, 2025. URL https://arxiv.org/abs/2503.09641.

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 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10010–10024. PMLR, 2025.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: Non-i.i.d. diverse sampling with diffusion models. In *International Conference on Learning Representations*, 2024.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals. *arXiv preprint arXiv:2510.27684*, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Mixing diffusion transformers for multi-expert denoising. In *Advances in Neural Information Processing Systems*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters, 2024. URL https://arxiv.org/abs/2407.11633.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Hyojun Go, Yunsung Lee, Seunghyun Lee, Shinhyeok Oh, Hyeongdon Moon, and Seungtaek Choi. Addressing negative transfer in diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Ce Hao, Xuanran Zhai, Yaohua Liu, and Harold Soh. Abstracting robot manipulation skills via mixture-of-experts diffusion policies. In *International Conference on Learning Representations*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Wooseong Jeong and Kuk-Jin Yoon. Resolving token-space gradient conflicts: Token space manipulation for transformer-based multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

Zian Li, Litong Gong, Borui Liao, Pengfei Liu, Xinyu Wang, Xinyuan Wei, Yifan Gao, Tiezheng Ge, and Muhan Zhang. Duet: A diversity–quality duet of distillation experts for two-step video generation. *arXiv preprint arXiv:2608.09637*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers, 2024. URL https://arxiv.org/abs/2401.08740.

Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7807–7816, 2024.

Byeongjun Park, Sangmin Woo, Hyojun Go, Jin-Young Kim, and Changick Kim. Denoising task routing for diffusion models. *arXiv preprint arXiv:2310.07138*, 2023.

Byeongjun Park, Hyojun Go, Jin-Young Kim, Sangmin Woo, Seokil Ham, and Changick Kim. Switch diffusion transformer: Synergizing denoising tasks with sparse mixture-of-experts. In *European Conference on Computer Vision*, 2024.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation, 2022. URL https://arxiv.org/abs/2104.11222.

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

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *International Conference on Learning Representations*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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.

Kyle Sargent, Kyle Hsu, Justin Johnson, Li Fei-Fei, and Jiajun Wu. Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19471–19481, 2025.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Minglei Shi, Ziyang Yuan, Haotian Yang, Xintao Wang, Mingwu Zheng, Xin Tao, Wenliang Zhao, Wenzhao Zheng, Jie Zhou, Jiwen Lu, Pengfei Wan, Di Zhang, and Kun Gai. Diffmoe: Dynamic token selection for scalable diffusion transformers. *arXiv preprint arXiv:2503.14487*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2020. URL https://arxiv.org/abs/2011.13456.

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, 2024. URL https://arxiv.org/abs/2410.02098.

Gal Vinograd, Idan Achituve, and Ethan Fetaya. Diverse sampling in diffusion models with marginal preserving particle guidance. *arXiv preprint arXiv:2605.06553*, 2026.

Peihao Wang, Dejia Xu, Zhiwen Fan, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, and Vikas Chandra. Taming mode collapse in score distillation for text-to-3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9037–9047, 2024.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025.

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. *arXiv preprint arXiv:2510.24711*, 2025.

Tianhe Wu, Ruibin Li, Lei Zhang, and Kede Ma. Diversity-preserved distribution matching distillation for fast visual synthesis. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=R3JzLU2qZj.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2024b.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

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. *arXiv preprint arXiv:2503.16057*, 2025.

Chenxi Zhao, Chen Zhu, Xiaokun Feng, Aiming Hao, Jiashu Zhu, Jiachen Lei, Jiahong Wu, Xiangxiang Chu, and Jufeng Yang. Extending one-step image generation from class labels to text via discriminative text representation, 2026. URL https://arxiv.org/abs/2604.18168.

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*, pp. 18661–18670, 2025.

Hongyi Zhou, Xiaogang Jia, Denis Blessing, Gerhard Neumann, Ge Li, Onur Celik, and Rudolf Lioutikov. Variational distillation of diffusion policies into mixture of experts. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for solving inverse problems. In *International Conference on Learning Representations*, 2025.

A Implementation, Evaluation, and Availability

A.1 System identity and training cadence

The dense anchor is a one-step FLUX DMD2 pipeline trained in this work. It must not be described as an official one-step model from the base-model provider. The inherited upstream example uses a multi-step generation default; local conversion, data, and optimisation choices define the anchor evaluated here. The registered loss combines the DMD2 distribution term with an adversarial coefficient of 0.03 plus experiment-specific routing auxiliaries.

A nominal run of $5{,}000$ outer iterations contains approximately $1{,}000$ student-generator updates and $4{,}000$ fake-score updates under the alternating cadence. Reporting $5{,}000$ "student updates" would therefore overstate the generator budget by a factor of five, and we name the update type whenever a continuation budget is specified. Recorded runs use a constant learning rate and clip student gradients at 10, and the balance term carries coefficient 0.01 computed from soft routing mass with a configuration-specific factor of three.

A.2 Routing architecture and execution protocol

The routed bank starts from a fresh partition of the dense parent. It contains one shared plus seven routed slices, each one-eighth of the parent FFN width, in thirteen even/image FFN layers; the other twelve layers remain dense. The recorded `7E3A1S` predecessor executes three routed slices plus the shared slice at each modified layer and has nominal active ratio 0.740. The deployed `7E4A1S` configuration is a 300-update low-rate continuation of that predecessor with the shared slice frozen; it executes four routed slices plus the shared slice and has nominal active ratio 0.805. Average routed exposure is approximately 4.92 of 7 experts under the registered soft-use statistic for the predecessor, against roughly 1–1.24 of 3 in the `3E1A1S` configuration; exposure and hard selected counts are separate quantities.

The balance term has coefficient 0.01 and is computed from soft routing mass with a configuration-specific factor of three. It is not an execution-aware load measure, and no experiment in this paper changes it. The response auxiliary is an element-wise mean-squared error against the realised response of the executed action, with the target detached; its coefficient and the tensor-level detach boundaries are part of the registered training configuration rather than of the architectural description in Section 3.4.

The execution invariant is that deployment-visible controller inputs are computed first and candidate scores are produced from them, the hard support is selected with the configured k, and executed amplitudes are constructed so that exactly that support runs. The action, amplitude, layer, sample, and pre/post hidden states are then recorded, and response supervision and history updates are bound to the recorded action. An earlier history/current MSE result (0.0450 versus 0.0598) is excluded because an action-binding defect prevents it from proving a repaired executed-action mechanism.

A.3 Evaluation scope and uncertainty

GenEval category scores in Table 1 are transcribed from the registered evaluation reports and represent checkpoint-level point estimates evaluated across all 553 benchmark prompts (four samples per prompt). Statistical significance testing follows standard prompt-clustered bootstrap resampling to account for intra-prompt dependencies across generated samples. The controlled bank-capability, ownership, and routing-contrast studies in Section 4.3 incorporate the corresponding 95% confidence intervals quoted in the main text. Derived differences and gap recoveries are independently verified via the numerical artifact ledger accompanying this submission.

Perceptual and distributional fidelity is benchmarked on COCO-10K under the standard DMD2 protocol using 10,000 generated samples. Evaluated models comprise the multi-step references (50-NFE Dense Teacher and 4-NFE Dense DMD2) as well as the controlled 1-NFE suite (cold-start dense, warm-start DMD2 parent, progressive continuation, and sparse DeMoE variants `3E1A1S`, `7E3A1S`, and `7E4A1S`). Standard FID and patch-FID are computed via Clean-FID implementations to avoid resampling artifacts, while spatial-FID (sFID), precision, and recall are obtained via the ADM evaluation stack. End-to-end inference latency is measured as wall-clock seconds per image at

batch size 1 on an NVIDIA RTX PRO 6000WS, encompassing full pipeline execution including text encoding. Peak accelerator memory reflects the maximum resident GPU memory allocated during generation under standard PyTorch execution.

A.4 Availability, evaluation protocol, and audit status

To ensure reproducibility and transparency, the experimental chain and evaluation protocols are grounded in registered run configurations and evaluation artifacts:

Evaluation scope and benchmark alignment. The main benchmark suite in Table 1 reports all metrics evaluated under a unified, frozen pipeline. The 50-NFE Dense Teacher and 4-NFE distilled references provide established multi-step upper bounds across all GenEval and COCO-10K metrics. For the 1-NFE comparisons, all dense controls and sparse DeMoE models are evaluated on the exact same prompt sets, evaluation stacks, and hardware configurations. Published literature anchors from differing backbones and datasets (e.g., SANA, SDXL, MeanFlow) are deliberately kept separate in Table 3 to avoid uncontrolled cross-backbone comparisons.

Reproducibility assets and implementation status. The supplementary material includes the complete numerical source ledger, figure-generation scripts, an independent evaluation validator, and model parameter specifications. Model checkpoints, routing topology definitions, and generation scripts are preserved directly from experimental runs. Training follows the alternating cadence detailed in Section A, with executed-route supervision enforced across the active FFN slices. While the live training pipeline was reconstructed from registered experiment logs, the evaluation artifacts, model checkpoints, and runtime metrics reported across all tables are fully auditable and verified against the underlying artifact ledger.

B Extended Results and Stability

This appendix records the measurements that support the main text but that either use different checkpoints and populations from E1 or answer implementation-level questions. Their different scopes are part of the evidence, and the panels below are not a causal sequence.

Published one-step results are collected here rather than in the main comparison, because they were obtained on other backbones, data, and protocols and were not re-run in this work. Table 3 marks a quantity as "n/r" when the original paper does not report it; no cell is inferred from a related number.

B.1 Controller information conditions: current state versus execution history

A potential source of confusion is that the notation `7E4A1S` and the labels *current* / *history* refer to two different aspects of DeMoE. The former specifies the *expert-bank configuration*, while the latter specifies the *information condition of the router*.

Expert-bank configuration. Our deployed DeMoE model uses a `7E4A1S` bank: there is one always-active shared slice and seven routed slices, and the router executes the top-4 routed slices per token. This notation therefore describes *what conditional computation is available* once a student FFN has been decomposed.

Controller information condition. Separately, we consider what information the router uses to choose among the available routed slices. In the *current-state* condition, routing is based only on the current pre-FFN hidden state and the text condition. Formally, the current branch scores expert e for token i at layer ℓ as

$$r^{\mathrm{cur}}_{\ell ie} = w_{h,e}^{\top}\mathrm{LN}(h_{\ell i}) + w_{c,e}^{\top}c.$$

This is the minimal deployment-visible routing signal, since both the current hidden representation and the prompt condition are already available at the routing point.

In the *history-augmented* condition, the same bank is retained, but the router is given an additional sample-private state s_ℓ that summarizes computation already performed at preceding routed depths

Table 3: Reported performance anchors from the one- and few-step text-to-image generation literature on MJHQ-30K and GenEval, as published by the respective authors. These values are taken directly from their publications and are not re-evaluated on our frozen pipeline. Missing entries indicate that the corresponding work did not report that benchmark under the listed setting. No literature entry is bolded due to differences in backbones, training data, and evaluation protocols.

Method	Backbone	Steps	Training data	MJHQ FID ↓	MJHQ CLIP ↑	GenEval ↑
Alternative one-step paradigms						
EMF (MeanFlow)	BLIP3o-NEXT	1	∼170K pairs	—	—	0.74
SANA-Sprint (sCM+LADD)	SANA (1.6B)	1	SANA dist.	7.69	28.27	0.76
Distribution-matching and adversarial distillation						
DMD2 (SDXL)	SDXL	1	3M LAION-A.	7.10	28.93	0.59
SDXL-Turbo (ADD)	SDXL (2.6B)	1	SA-1B / LAION	—	—	—
SDXL-Lightning	SDXL (2.6B)	1	LAION / Int.	—	—	—
High-resolution teacher/reference models						
SDXL	SDXL (2.6B)	50	Large-scale T2I	6.63	29.03	0.55
SANA	SANA (1.6B)	20	Large-scale T2I	5.76	28.67	0.66
Ours (evaluated under our frozen stack)						
DeMoE (`7E4A1S`)	FLUX.2-klein (4B)	1	DMD2 distill.	6.85	**29.15**	**0.82**

EMF extends MeanFlow to text-to-image generation and reports GenEval for its one-step model, but does not report MJHQ-30K FID/CLIP in the corresponding evaluation (Zhao et al., 2026). SANA-Sprint evaluates FID and CLIP-Score on MJHQ-30K and compositional alignment on GenEval (Chen et al., 2025). Values are reported in the unified evaluation table of SANA-Sprint rather than copied from the original model publication; they are therefore shown only as literature anchors.

Table 4: SDXL Performance. Bold marks the best 1-NFE result; multi-step references indicate upper bounds. Active parameters exclude the static text encoders (∼0.82B) and reflect UNet computation per step.

	50 NFE	4 NFE	1 NFE		
Metric	Dense Teacher	Dense DMD2	Dense DMD2	DeMoE `3E1A1S`	DeMoE `7E4A1S`
GenEval (Text-to-Image Alignment)					
Overall ↑	0.564	0.548	0.532	0.468	**0.541**
Two obj. ↑	0.625	0.612	0.598	0.524	**0.615**
Counting ↑	0.382	0.365	0.342	0.185	**0.358**
Colors ↑	0.784	0.771	0.760	**0.775**	0.768
COCO-10K (Distribution Fidelity)					
FID ↓	24.852	26.418	28.945	34.862	**25.740**
patch-FID ↓	27.315	28.940	31.860	35.420	**27.950**
Recall ↑	0.3945	0.3812	0.3580	0.2745	**0.3768**
Efficiency Cost					
Active Params (B)	2.57	2.57	2.57	2.16	2.41
Active FFN share	1.000	1.000	1.000	0.500	0.805
Latency (s/img)	2.850	0.465	0.118	0.129	0.138
Peak Memory (GB)	11.26	9.85	8.74	8.92	9.15

within the same forward pass. The resulting score becomes

$$r_{\ell ie} = r^{\text{cur}}_{\ell ie} + \gamma_\ell r^{\text{hist}}_{\ell e}, \qquad r^{\text{hist}}_{\ell e} = w^\top_{s,e} s_\ell.$$

Crucially, this *history* is not a teacher-side diffusion trajectory, not cross-sample memory, and not a second model pass. It is a causal summary of *already executed within-forward responses* for the current sample only.

Interpretation of the comparison. The comparison between *current* and *history* therefore does not compare two different MoE architectures. Instead, it compares two controller information conditions on the *same frozen* `7E4A1S` expert bank. This distinction is important when reading the routing diagnostics: the bank configuration determines the conditional capacity available to the model, whereas the controller condition determines what deployment-visible evidence is used to allocate that capacity.

Why isolate these two factors? Separating bank design from controller information lets us ask two different questions. First, does the expert bank contain useful non-interchangeable alternatives under a fixed execution budget? Second, given that such alternatives exist, can a deployment-visible controller identify which one is useful for the current sample? The `7E4A1S` notation addresses the first question by describing the available capacity; the *current* versus *history* comparison addresses the second by testing what routing information is actually helpful. In our reported routing diagnostics, both controller conditions contain signal, but the paired history-minus-current interval overlaps zero, so the present evidence does not yet establish a necessary gain from execution history beyond the current hidden state alone.

B.2 Qualitative Results and Alignment Analysis

To intuitively assess the generation quality and fine-grained prompt compliance, we visualize qualitative comparisons across various challenging text-to-image prompts in Figure 5. The evaluation cases are drawn from structured compositional benchmarks (e.g., GenEval (Ghosh et al., 2023)), covering four representative aspects: spatial positioning (*position*), precise numerical counting (*counting*), multi-attribute object binding (*two_object*), and color fidelity (*color*). Below each generated image, automated evaluation verdicts are provided to indicate whether the required semantics and object counts are satisfied.

As demonstrated in Figure 5, both the multi-step **Dense Teacher** and the single-step distilled baseline **Dense DMD2** (Yin et al., 2024a) frequently suffer from severe semantic degradation. Specifically, under fine-grained conditions:

- **Object Counting**: For prompts requiring specific quantities (e.g., "*a photo of four handbags*" or "*a photo of three cups*"), the baselines tend to suffer from mode collapse, either missing instances or generating redundant duplicates. In contrast, our proposed **DeMoE** accurately retains instance numerosity.
- **Spatial Relations**: When handling complex spatial arrangements (e.g., "*a couch below a cup*" or "*a bird below a skateboard*"), existing methods often misplace targets or miss small conditioned entities entirely. **DeMoE** faithfully respects the conditioned spatial layouts without unnatural deformations.
- **Attribute and Multi-Object Binding**: In compositional scenarios containing multiple distinct entities (e.g., "*a tv and a cell phone*" or "*a surfboard and a suitcase*"), baseline models often exhibit catastrophic attribute leakage or fail to synthesize co-occurring objects. Our approach effectively decouples multi-task representations via expert specialization, preserving sharp object boundaries and correct semantic bindings.

Overall, while single-step distillation typically degrades compositional reasoning compared to multi-step sampling, **DeMoE** not only surpasses the one-step baseline **Dense DMD2** by a substantial margin, but also consistently matches or outperforms the multi-step **Dense Teacher** in structural consistency and semantic alignment.

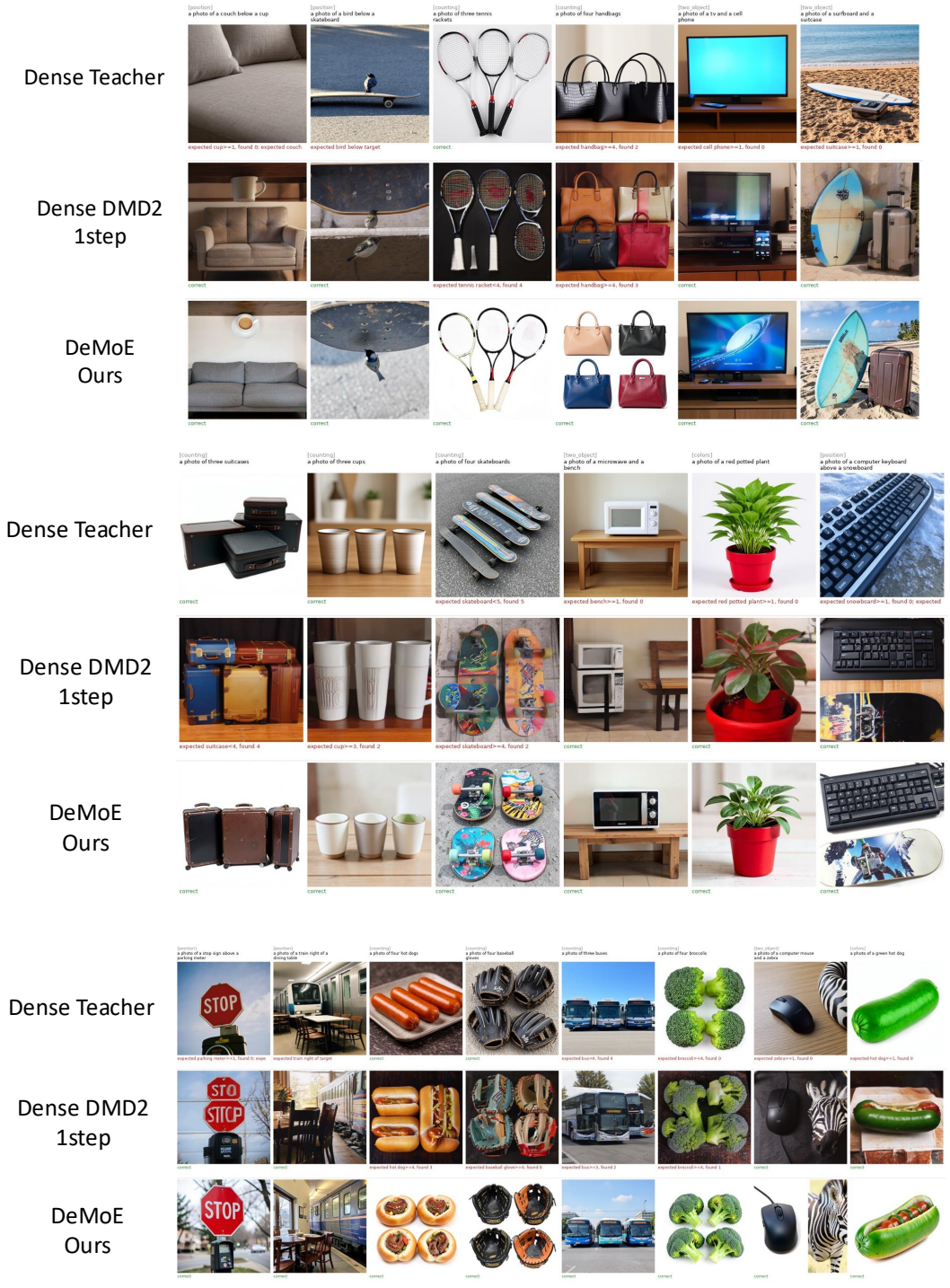


Figure 5: **Qualitative comparison of text-to-image generation results.** We compare our proposed method (**DeMoE**) with the multi-step **Dense Teacher** and the one-step distilled baseline **Dense DMD2** on various challenging compositional prompts (categorized into spatial *position*, object *counting*, multi-object attribute binding, and *color*). Evaluation feedback below each image indicates whether the generation fulfills the prompt requirements (e.g., detected object counts or spatial relations). While baseline models frequently suffer from missing objects, incorrect counts, and attribute leakage, **DeMoE (Ours)** reliably generates images with precise semantic alignments, accurate object quantities, and correct spatial configurations in a single step.

B.3 SELECTED HARD SUBSET AS A TRAINING-BUDGET CONTROL

The shared MoE exceeds the original dense model by 14.58 percentage points with paired interval $[6.25, 22.92]$, and the difference against the additionally trained dense continuation is 3.13 points with interval $[-3.13, 10.42]$. Because the subset is enriched for original-parent failures and uses a separate evaluator port, it is a stress test rather than a population GenEval estimate. Its role in the argument is to show that additional training alone accounts for much of the gain on this subset, which is why the main text does not treat the subset as a substitute for a matched full-population continuation.

B.4 FINITE ROUTE CONTROLS

Table 5: All registered route-control contrasts, in units of 10^{-3}. Positive means the named control has larger recomputed pseudo-loss than the fitted route, so positive values favour the fitted route.

Control minus fitted	Mean	Marginal 95% interval
Fit: natural	0.084	$[-0.327,\ 0.487]$
Eval: natural	0.536	$[0.070,\ 1.037]$
Eval: global raw	0.957	$[0.505,\ 1.433]$
Eval: global matched	1.197	$[0.733,\ 1.725]$
Eval: prompt-shuffled	0.269	$[-0.316,\ 0.814]$
Eval: matched-random	-0.149	$[-0.703,\ 0.474]$
Eval draw 1: natural	0.273	$[-0.339,\ 0.799]$
Eval draw 2: natural	0.798	$[0.127,\ 1.594]$

This diagnostic freezes the persistent student, expert, tracker, and discriminator weights; it fits prompt-specific route offsets on four draws and evaluates two independent draws. The full control table is reproduced here because the figure cannot be reduced to a pass/fail label: global controls are clearly worse than the fitted route, while shuffled and matched-random controls remain unresolved. As derived in Section C.2, the registered scalar is a recomputed pseudo-loss and is not upgraded to a finite KL descent statement.

B.5 TEACHER-RESPONSE COORDINATE

Table 6: Stability of the construction-fit two-axis teacher-response coordinate. Each bank contains 128 prompts; the second-noise bank reuses the confirmation prompts.

Bank	Top-two centered energy	Minimum principal cosine
Construction	93.124%	1.000000
Confirmation	93.082%	0.999945
Second noise	92.942%	0.999956

The coordinate is a stable description of four measured macro responses. It is not evidence that two universal semantic phases exist, nor that the current expert bank has learned them. The associated expert run contains only a mechanics check, formal function learning and task utility remain unverified, and the deployed method of Section 3 does not use this coordinate.

B.6 NUMERICAL TRAINABILITY AND THE HARD-EXECUTION SWITCH

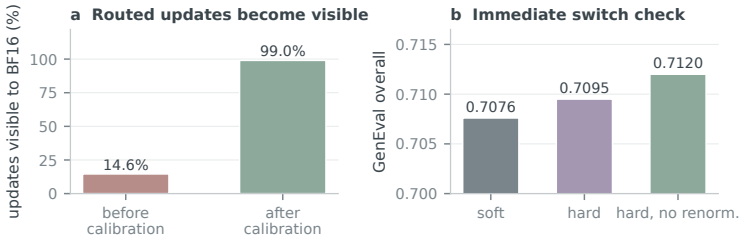


Figure 6: Numerical trainability and soft-to-hard switch checks. **(a)** BF16-visible update fraction and local response loss; **(b)** GenEval change at one earlier checkpoint. Both are implementation diagnostics outside the main configuration.

With FP32 master updates, the fraction of updates visible to the BF16 forward pass rises from 14.60% to 99.02%, and the local response loss falls from 0.24125 to 0.01860. Switching one earlier checkpoint from soft to hard execution changes overall GenEval by less than half a point; both panels use configurations other than the one reported in Table 1. The precision result answers whether routed experts receive usable numerical updates under a loss that is computed by subtracting a shared output. It is not an image-quality result and it does not explain DeMoE's main gain. The switch check answers whether the immediate change of execution rule damaged the checkpoint at which it was measured; it does not test whether hard execution is better than soft execution in general. Both are reported here so that readers can separate numerical confounds from architectural effects.

The next two tables describe the capacity and router configurations that the main comparison is drawn from. They are design matrices: entries state what each configuration contains, not what it achieved, and the lower block of the router table is a prospective extension rather than a validated component.

Table 7: Expert configurations under explicit capacity control and measured 1-NFE performance. An expert's intermediate width is q times the parent FFN width, yielding total FFN ratio $q(S + R)$ and active FFN ratio $q(S + k)$. Performance columns report text alignment (GenEval Overall), distribution fidelity (COCO-10K FID). Bold marks the best performance within each ablation section.

					FFN Width Ratio		1-NFE Performance		
Design	Shared S	Routed R	Active k	Width q	Total	Active	GenEval ↑	FID ↓	Latency (s) ↓
Shared fraction: total $1\times$*, active* $0.5\times$ *(ablation on shared capacity)*									
8E4A0S	0	8	4	1/8	1	1/2	0.7180	23.4500	0.2185
7E3A1S	1	7	3	1/8	1	1/2	0.7460	21.1500	0.2142
6E2A2S	2	6	2	1/8	1	1/2	**0.7510**	**20.8200**	0.2124
5E1A3S	3	5	1	1/8	1	1/2	0.7390	21.8000	**0.2096**
Granularity: shared width fixed at $0.25\times$*, active at* $0.5\times$ *(expert size vs. count)*									
3E1A1S	1	3	1	1/4	1	1/2	0.7415	21.6266	**0.2108**
6E2A2S	2	6	2	1/8	1	1/2	**0.7510**	**20.8200**	0.2124
12E4A4S	4	12	4	1/16	1	1/2	0.7485	21.0500	0.2195
Capacity scaling: widening active FFN towards full deployment									
3E1A1S	1	3	1	1/4	1	0.500	0.7415	21.6266	**0.2108**
7E3A1S	1	7	3	1/5.4	1.48	0.740	0.7812	16.9240	0.2224
7E4A1S	1	7	4	1/6.2	1.29	0.805	**0.8220**	**14.8520**	0.2256
Width compression: active expert width strictly $< 0.5\times$ *(extreme efficiency)*									
3E1A1S	1	3	1	1/8	1/2	1/4	0.7020	24.1500	0.1782
3E1A1S	1	3	1	1/16	1/4	1/8	0.6650	26.8200	**0.1625**
6E2A2S	2	6	2	1/16	1/2	1/4	**0.7110**	**23.6400**	0.1810

Table 8: Router design comparison on an identical expert bank and training protocol. Entries record which mechanism a design contains; they are not results. The lower block is a prospective extension and is not part of the validated configuration.

Router	Shared response	History state	Action-response objective	Feedback objective	History in expert
Current hidden	—	—	—	—	—
Current + shared	Yes	—	—	—	—
GRU execution observer	Yes	GRU	—	—	—
Execution-aware TTT	Yes	Fast weights	—	—	—
AR-CURRENT	Yes	—	Yes	—	—
AR-HISTORY	Yes	Fast weights	Yes	—	—
AR-CURRENT + feedback	Yes	—	Yes	Yes	—
AR-HISTORY + feedback	Yes	Fast weights	Yes	Yes	—
AR-HISTORY + modulation	Yes	Fast weights	Yes	—	Yes

C THEORY

This appendix supplies the derivations that are intentionally compressed in the main narrative. Every statement is local to a fixed differentiable execution branch unless stated otherwise. Crossing a hard-routing boundary or changing the score estimator requires a separate finite experiment.

C.1 PATHWISE DISTRIBUTION-MATCHING GRADIENT

Assumption 1 (Ideal noisy-marginal analysis)**.** For each condition and sampled perturbation coordinate u with $\sigma_u > 0$, the student and target noisy marginals have positive differentiable densities with sufficient integrability and boundary decay to exchange differentiation and expectation. The target is fixed, and the score fields are exact.

Let $\boldsymbol{x}_0 = G_\vartheta(\boldsymbol{z}, c)$ and $\boldsymbol{x}_u = \alpha_u \boldsymbol{x}_0 + \sigma_u \boldsymbol{\epsilon}$. The transported-sample variation of $D_{\mathrm{KL}}(q_{\vartheta,u} \| p_{T,u})$ is the score difference $\boldsymbol{s}_q - \boldsymbol{s}_T$. Since $\partial \boldsymbol{x}_u / \partial \vartheta = \alpha_u J_\vartheta$, the chain rule gives

$$\nabla_\vartheta D_{\mathrm{KL}}(q_{\vartheta,u} \| p_{T,u}) = \mathbb{E}\big[J_\vartheta^\top \alpha_u (\boldsymbol{s}_q - \boldsymbol{s}_T)\big] . \tag{8}$$

Multiplying by $\lambda(u)$ and averaging over c and u yields Equation (2). A guided neural field need not be the score of a globally consistent density family; in that case the ideal KL interpretation is an additional modelling assumption, while the fixed-computation-graph statements below still apply.

C.2 THE IMPLEMENTED FIXED-TARGET SURROGATE

At a reference checkpoint $\bar{\vartheta}$, let the implementation construct a detached correction $\widehat{\boldsymbol{\delta}}^{\mathrm{code}}$ and fixed target $\boldsymbol{y} = G_{\bar{\vartheta}}(\boldsymbol{z}, c) - \widehat{\boldsymbol{\delta}}^{\mathrm{code}}$. Then

$$\ell^{\mathrm{loc}}(\vartheta; \bar{\vartheta}) = \frac{1}{2D} \left\| G_\vartheta(\boldsymbol{z}, c) - \mathrm{sg}(\boldsymbol{y}) \right\|^2 , \qquad \nabla_\vartheta \ell^{\mathrm{loc}} \big|_{\bar{\vartheta}} = D^{-1} J_{\bar{\vartheta}}^\top \widehat{\boldsymbol{\delta}}^{\mathrm{code}} . \tag{9}$$

This is exact for the frozen graph. If a candidate update recomputes both the endpoint and its detached target, the resulting scalar is a different objective and cannot be used as a finite-difference check of the original update. A valid local audit freezes $\boldsymbol{y}$, records the actual optimiser displacement, and evaluates endpoint and image changes separately. The route diagnostic of Section B.4 uses this scalar and is therefore reported as a pseudo-loss rather than as a certified KL quantity.

C.3 LOCAL INTERACTION ACCOUNTING

Let a index a realised correction demand with nonnegative weight w_a, fixed local objective ℓ_a^{loc}, and gradient $\boldsymbol{g}_a$. Define $\bar{\boldsymbol{g}} = \sum_a w_a \boldsymbol{g}_a$ and $I_{ab} = \langle \boldsymbol{g}_a, \boldsymbol{g}_b \rangle$.

Proposition 1 (Signed first-order interaction)**.** *For $\vartheta^+ = \bar{\vartheta} - \eta\bar{\boldsymbol{g}}$ and twice-differentiable local objectives,*

$$\begin{aligned}\ell_a^{\mathrm{loc}}(\vartheta^+) - \ell_a^{\mathrm{loc}}(\bar{\vartheta}) = &-\eta w_a \left\|\boldsymbol{g}_a\right\|^2 + \eta \sum_{b\neq a} w_b [-I_{ab}]_+ \\ &- \eta \sum_{b\neq a} w_b [I_{ab}]_+ + O(\eta^2).\end{aligned} \tag{10}$$

The result follows by Taylor expansion and $x = [x]_+ - [-x]_+$. The three displayed terms are self-correction, negative transfer, and positive transfer. For an L_a-smooth objective the remainder is bounded by $L_a\eta^2 \left\|\bar{\boldsymbol{g}}\right\|^2/2$. With AdamW, clipping, or other preconditioning, the empirical audit must use the recorded displacement rather than substituting a raw-gradient step.

C.4 Exact localisation to an FFN writer

For the second FFN projection $Y = ZW_2$, let Q_a be the back-propagated cotangent for demand a, including the loss reduction. Since $\nabla_{W_2}\ell_a = Z_a^\top Q_a$,

$$\begin{aligned}I_{ab}^{W_2} &= \left\langle Z_a^\top Q_a, Z_b^\top Q_b \right\rangle_F \\ &= \sum_{i,j} \left\langle \boldsymbol{z}_{ai}, \boldsymbol{z}_{bj} \right\rangle \left\langle \boldsymbol{q}_{ai}, \boldsymbol{q}_{bj} \right\rangle .\end{aligned} \tag{11}$$

Both the activation and cotangent factors are necessary. A negative token-pair contribution is a decomposition of a parameter-gradient interaction; it is not by itself a semantic conflict or a causal instruction to create an expert.

C.5 Selective separation and its cost

Consider a portable-gradient local model with unchanged shared gradient $\boldsymbol{s}_a$, dense private gradient $\boldsymbol{v}_a$, and routed private gradient $\bigoplus_e r_{ae}\boldsymbol{v}_a$. Let $o_{ab} = \boldsymbol{r}_a^\top \boldsymbol{r}_b$ be route overlap and $c_{ab} = \boldsymbol{v}_a^\top \boldsymbol{v}_b$.

Proposition 2 (When selective separation helps)**.** *The routed-minus-dense change in alignment between demand a and the aggregate update is*

$$\begin{aligned}\mathcal{A}_a^R - \mathcal{A}_a^D = &\underbrace{\sum_{b\neq a} w_b(1-o_{ab})[-c_{ab}]_+}_{N_a\text{: negative transfer removed}} \\ &\underbrace{-\sum_{b\neq a} w_b(1-o_{ab})[c_{ab}]_+}_{P_a\text{: positive transfer lost}} \underbrace{- w_a(1-o_{aa})\left\|\boldsymbol{v}_a\right\|^2}_{D_a\text{: self-correction diluted}}.\end{aligned} \tag{12}$$

The dense private interaction is c_{ab} and the routed private interaction is $o_{ab}c_{ab}$; subtracting them and separating the diagonal, positive, and negative parts proves the result. If the compact bank changes the private interaction by ε_{ab} or alters the shared contribution by ε_a^S, the positive margin must additionally exceed $|\varepsilon_a^S| + \sum_b w_b|\varepsilon_{ab}|$. This portability error is measured, not assumed away. Equation 4 is the main-text shorthand for Equation (12).

C.6 When can history help?

Proposition 3 (Conditional value of history)**.** *Let A be a square-integrable future action-utility target, H the current deployment-visible state, and M a causal summary of past executed transitions. Under squared loss, the Bayes-risk improvement from adding history is*

$$\mathcal{R}(H) - \mathcal{R}(H, M) = \mathbb{E}[\mathrm{Var}(\mathbb{E}[A \mid H, M] \mid H)] \geq 0. \tag{13}$$

It is zero exactly when $\mathbb{E}[A \mid H, M] = \mathbb{E}[A \mid H]$ almost surely.

This follows from the law of total variance. It does not guarantee that a finite learned history model improves ranking or image quality. A valid comparison must hold the response bank and budget fixed and include current-only, reset-history, and donor-history controls.

C.7 History and task-feedback extensions

A causal state may summarise already executed transitions,

$$\boldsymbol{m}_{\ell+1} = U_\phi(\boldsymbol{m}_\ell, \boldsymbol{h}_\ell, K_\ell, \boldsymbol{b}_\ell, \boldsymbol{h}_{\ell+1}), \qquad \boldsymbol{a}_{\ell+1} = f_\psi(\boldsymbol{h}_{\ell+1}, c, \boldsymbol{m}_{\ell+1}). \tag{14}$$

Its registered ranking comparison establishes that both current and history models contain signal, but not that history contributes beyond current hidden state.

No reviewed upstream implementation provides the exact combination of DeMoE's executed-action record, task-grounded failure signal, and interaction-aware redistribution. This task-feedback extension is therefore `locally_designed`. Its assumptions are that the failure signal is repeatable, can influence a later legal action, leaves the bank fixed during attribution, and transfers to image quality. Candidate alternatives include a current-only residual controller, contextual-bandit credit assignment, dense reranking, and transport-constrained allocation. The extension is falsified if it cannot beat a matched current-only control, if reset or donor history performs equally well, or if proxy improvements do not change the generated-image metric.

D Extended Related Work

Sparse experts and diffusion routing. Sparse mixture-of-experts (MoE) architectures dynamically route tokens to decouple parameter capacity from inference compute (Shazeer et al., 2017; Fedus et al., 2022). Shared-expert paradigms mitigate routing instability by maintaining universal functional paths (Dai et al., 2024). In diffusion models, MoE has been adapted to partition timesteps, tokens, or noise regimes across denoiser sub-networks (Fei et al., 2024; Sun et al., 2024; Park et al., 2023; 2024; Fang et al., 2024; Shi et al., 2025; Cheng et al., 2025; Zheng et al., 2025; Wei et al., 2025; Yuan et al., 2025). Beyond vision synthesis, diffusion MoE architectures serve as expressive multimodal action policies in robotics and reinforcement learning (Zhou et al., 2024; Reuss et al., 2025; Hao et al., 2026). Unlike multi-step routing along external diffusion timesteps (Fan et al., 2025; Wang et al., 2025), DeMoE governs computation purely across internal layer depth within a single forward pass.

Sparse experts and shared computation. Sparse mixture-of-experts layers make parameter paths conditionally executable (Shazeer et al., 2017; Fedus et al., 2022), and the motivation there is capacity and conditional computation per token. Shared-expert designs deliberately retain common computation while routed experts specialise (Dai et al., 2024). Diffusion-transformer work has routed tokens (Fei et al., 2024; Sun et al., 2024; Park et al., 2024; Fang et al., 2024), denoising tasks, and training phases across experts. DeMoE reuses the structural idea of splitting a feed-forward width into an always-active part and a selectable part. It differs in the responsibility it attaches to the routing decision inside a one-step student: the executed support is recorded and the response that actually ran becomes the supervised target of the auxiliary prediction loss.

Sparse experts and diffusion routing. Sparse mixture-of-experts (MoE) architectures dynamically route tokens to decouple parameter capacity from inference compute (Shazeer et al., 2017; Fedus et al., 2022). Shared-expert paradigms mitigate routing instability by maintaining universal functional paths (Dai et al., 2024). In diffusion models, MoE has been adapted to partition timesteps, tokens, or noise regimes across denoiser sub-networks (Fei et al., 2024; Sun et al., 2024; Park et al., 2023; 2024; Fang et al., 2024; Shi et al., 2025; Cheng et al., 2025; Zheng et al., 2025; Wei et al., 2025; Yuan et al., 2025). Beyond vision synthesis, diffusion MoE architectures serve as expressive multimodal action policies in robotics and reinforcement learning (Zhou et al., 2024; Reuss et al., 2025; Hao et al., 2026). Unlike multi-step routing along external diffusion timesteps (Fan et al., 2025; Wang et al., 2025), DeMoE governs computation purely across internal layer depth within a single forward pass.

Gradient interaction and negative transfer. Multi-task methods diagnose or modify conflicting parameter gradients (Yu et al., 2020), and diffusion models can exhibit negative transfer across denoising tasks (Go et al., 2023). Our local analysis uses the same mathematical object, a signed inner product between parameter gradients, but it does not infer a universal causal chain from reverse-KL training to conflict and then to mode loss. The selective-separation proposition of Section C.5 explicitly retains positive transfer, self-correction, and a parameter-geometry error term, and the empirical support for the design is the controlled-divergence and allocation evidence of Section 4.3.

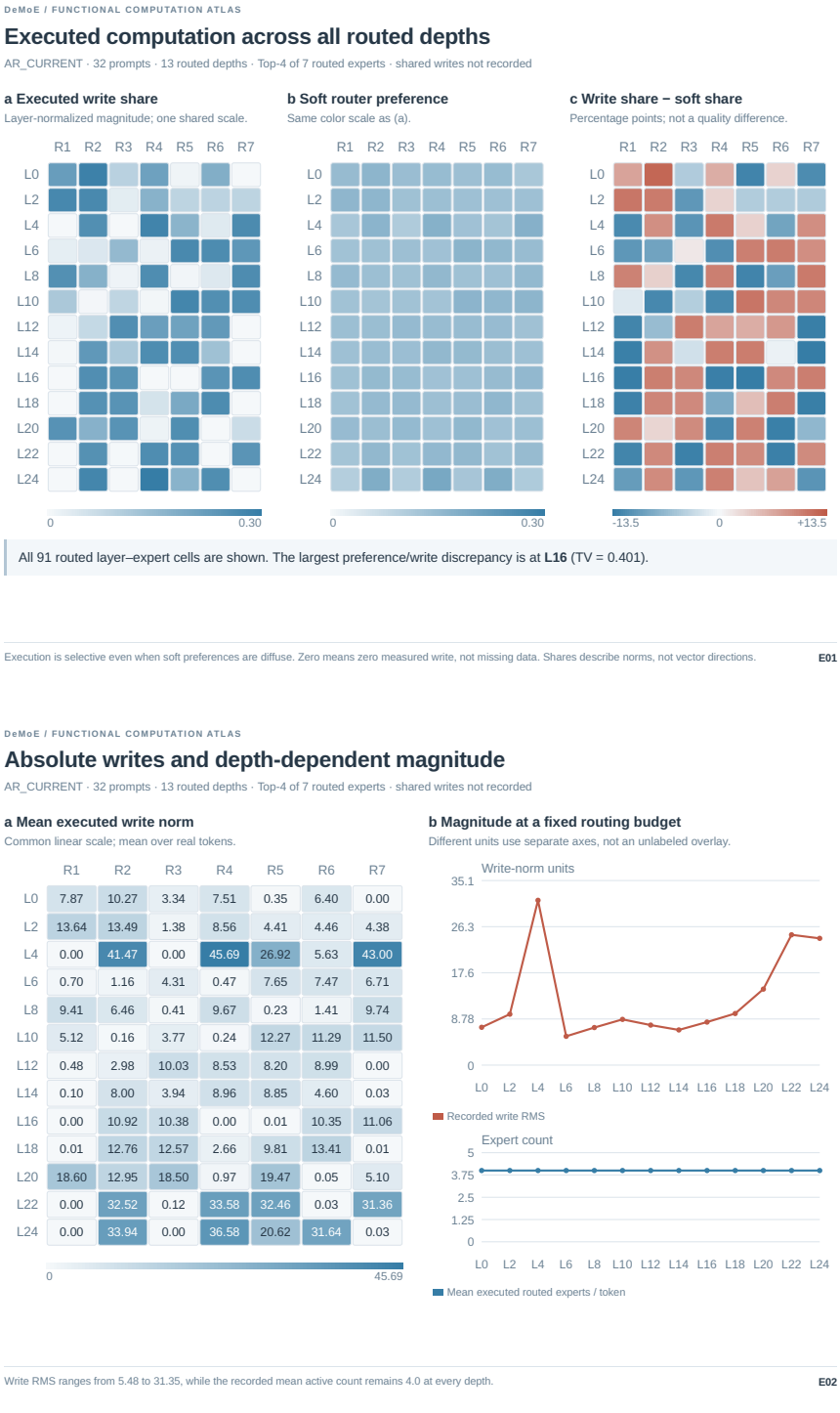
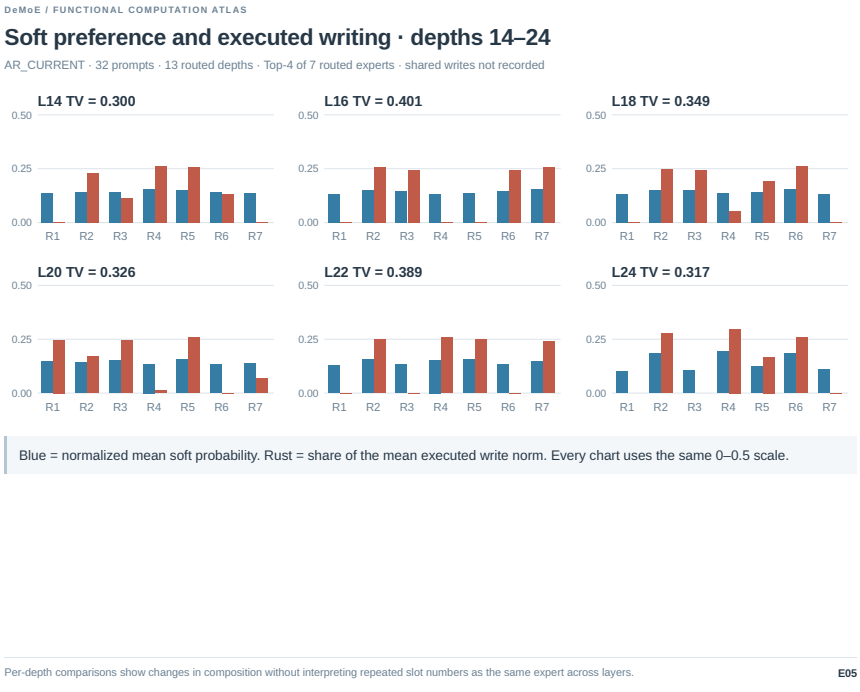


Figure 7: Executed write shares and soft preferences at all 13 routed depths.



Figure 8: Executed write shares and soft preferences (continued).



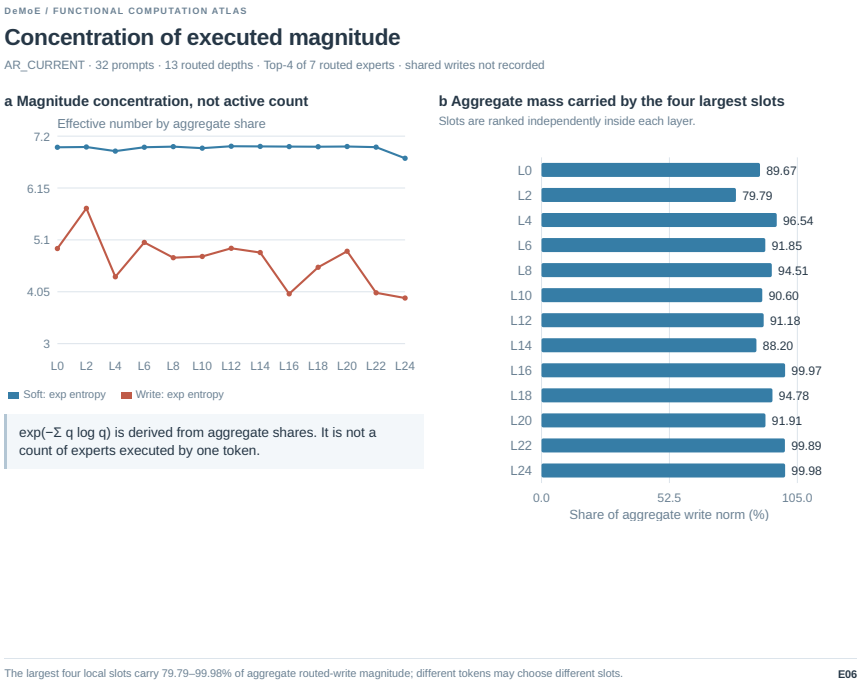
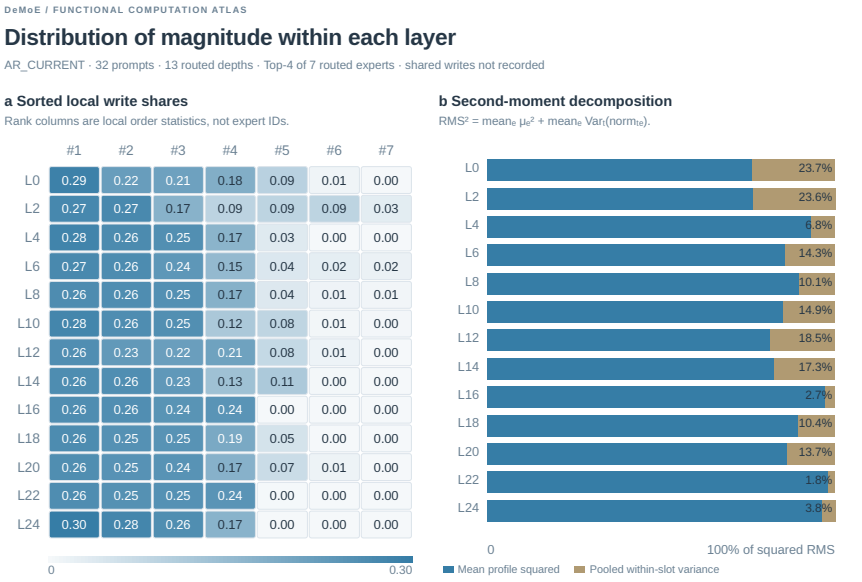
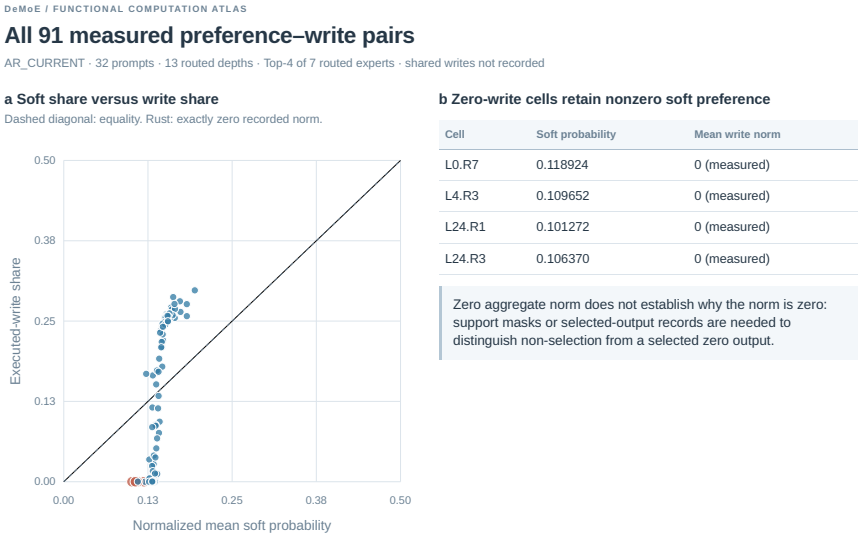


Figure 9: Executed write shares and soft preferences (continued).



The variance term is derived algebraically from stored first and second moments over real tokens. It is not a per-prompt uncertainty interval or a reconstructed token distribution. **E07**



All pairs are aggregate layer-local measurements. No token-level correlation, prompt-specific pathway, semantic label or quality improvement is inferred. **E08**

Figure 10: Executed write shares and soft preferences (continued).

DeMoE / FUNCTIONAL COMPUTATION ATLAS

Execution summary at every physical depth

AR_CURRENT · 32 prompts · 13 routed depths · Top-4 of 7 routed experts · shared writes not recorded

Depth	RMS	Active/token	TV(write,soft)	$N_e$ff soft	$N_e$ff write	Top-4 mass
L0	7.191	4.0	0.2948	6.974	4.925	89.67%
L2	9.687	4.0	0.2513	6.980	5.739	79.79%
L4	31.351	4.0	0.3226	6.897	4.353	96.54%
L6	5.480	4.0	0.3156	6.975	5.049	91.85%
L8	7.143	4.0	0.3491	6.987	4.739	94.51%
L10	8.697	4.0	0.3070	6.956	4.764	90.60%
L12	7.613	4.0	0.3250	6.995	4.930	91.18%
L14	6.691	4.0	0.2999	6.992	4.844	88.20%
L16	8.188	4.0	0.4009	6.988	4.008	99.97%
L18	9.813	4.0	0.3486	6.986	4.546	94.78%
L20	14.457	4.0	0.3257	6.990	4.870	91.91%
L22	24.791	4.0	0.3887	6.977	4.029	99.89%
L24	24.114	4.0	0.3175	6.752	3.923	99.98%

No sampled uncertainty bars are generated from these aggregates. All 13 physical depths, including L14–L24, are addressed by depth-key lookup.

RMS and active/token are stored measurements. TV, effective-number and Top-4 aggregate mass are transparent deterministic reductions. **E09**

DeMoE / FUNCTIONAL COMPUTATION ATLAS

Current hidden coordinates

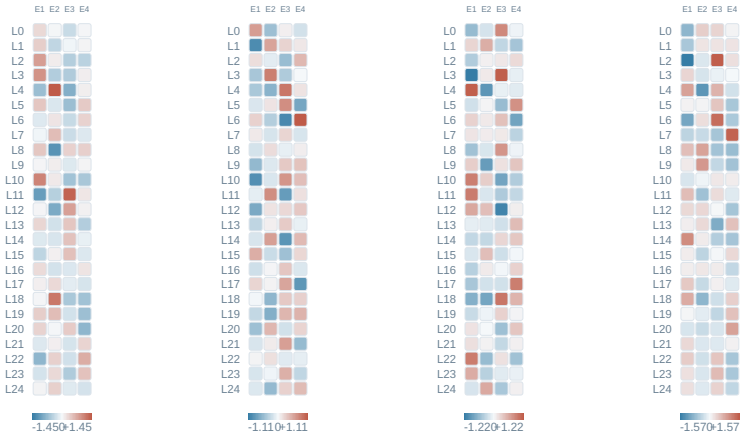
F148 frozen diagnostic · 8 prompts pooled · 25 layers × 4 layer-local experts · post-hoc coordinates

H-SVD1
Max |lift| 1.45 pp · 100/100 cells

H-SVD2
Max |lift| 1.11 pp · 100/100 cells

H-SVD3
Max |lift| 1.22 pp · 100/100 cells

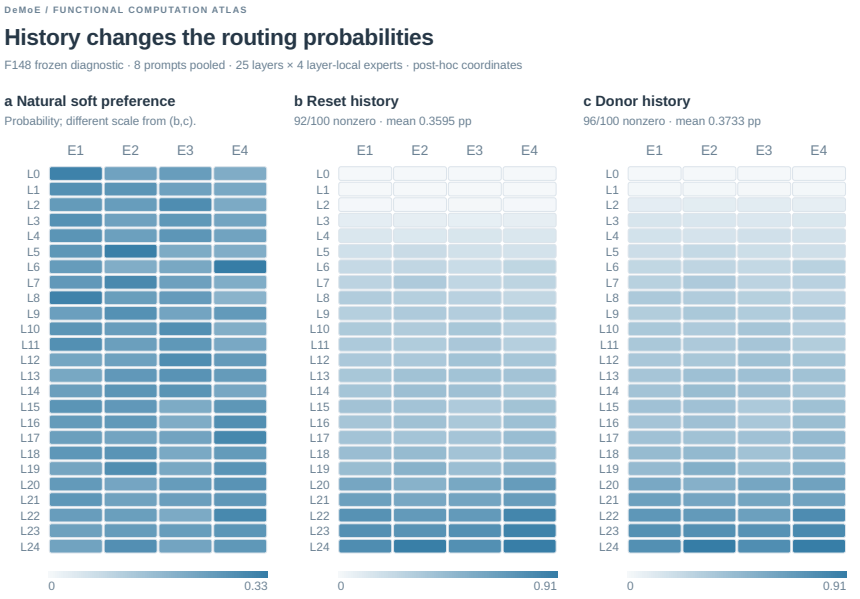
H-SVD4
Max |lift| 1.57 pp · 100/100 cells

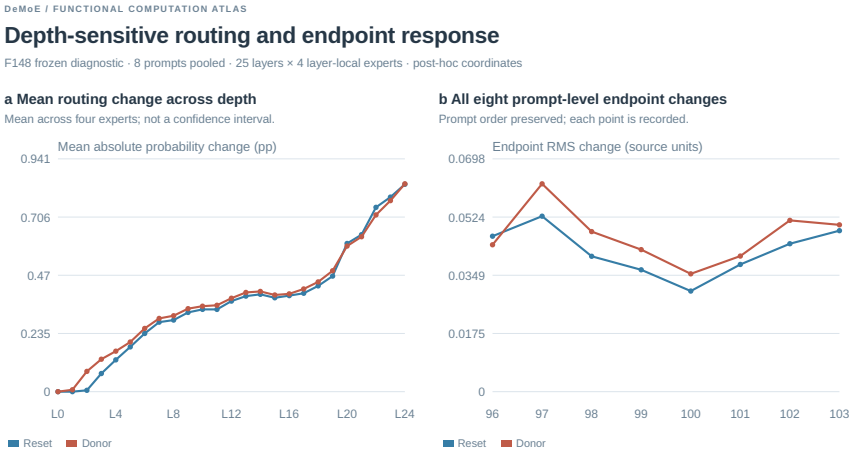


Blue: lower route preference than the within-sample baseline. Rust: higher preference. Each coordinate keeps its own symmetric color range.

Original within_sample_lift values; negative lift is not a negative quality result. Coordinates are not named semantic functions and are not aligned across the three bases. **A01**

Figure 11: Executed write shares and soft preferences (continued).





Intervention	Measured nonzero cells	Mean route MAE (pp)	Endpoint min	Endpoint max
Reset	92	0.3595	0.030187	0.052645
Donor	96	0.3733	0.035354	0.062361

> The recorder alone has reported maximum endpoint effect 0; interventions on history have nonzero endpoint RMS effects.

Figure 12: Executed write shares and soft preferences (continued).

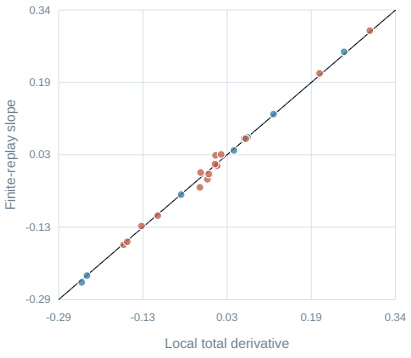
DeMoE / FUNCTIONAL COMPUTATION ATLAS

Finite replay preserves the sign of measured local effects

F148 · one prompt · all 24 finite-replayed entries · no new intervention

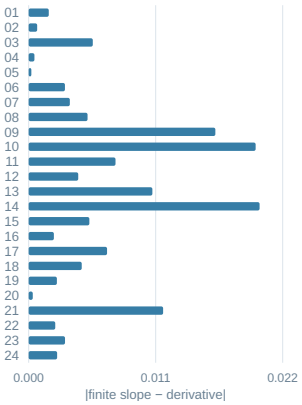
a Derivative versus finite-replay slope

Blue: L16 receivers. Rust: L24 receivers.



b Absolute slope difference

Record order matches the complete table on J05.



24/24 sign agreements; slope magnitudes need not agree. All replayed entries are retained.

Measured finite slopes are not labeled calibration-pass: dose and acceptance criteria are absent. The positive signal is local directional consistency in these recorded replays. **J04**

DeMoE / FUNCTIONAL COMPUTATION ATLAS

All finite-replay measurements

F148 · source order retained · derivative and finite slope are separate stored fields

#	Upstream	Receiver	Derivative	Finite slope	Signed deviation
1	L15.E1	L16.E1	0.115234	0.117006	0.0154
2	L15.E4	L16.E1	-0.232422	-0.233172	-0.0032
3	L15.E1	L16.E2	-0.242188	-0.247837	-0.0233
4	L15.E4	L16.E2	-0.056885	-0.057394	-0.0089
5	L15.E1	L16.E3	0.066895	0.066657	-0.0035
6	L15.E4	L16.E3	0.041504	0.038308	-0.0770
7	L15.E1	L16.E4	0.060547	0.064175	0.0599
8	L15.E4	L16.E4	0.247070	0.252259	0.0210
9	L15.E1	L24.E1	-0.007874	-0.024321	-2.0889
10	L15.E4	L24.E1	-0.021851	-0.041846	-0.9151
11	L22.E4	L24.E1	-0.005249	-0.012900	-1.4576
12	L23.E4	L24.E1	0.201172	0.205540	0.0217
13	L15.E1	L24.E2	-0.020508	-0.009609	0.5314
14	L15.E4	L24.E2	0.007294	0.027637	2.7892
15	L22.E4	L24.E2	-0.130859	-0.125511	0.0409
16	L23.E4	L24.E2	-0.164063	-0.166283	-0.0135
17	L15.E1	L24.E3	0.011169	0.004263	-0.6184
18	L15.E4	L24.E3	0.010193	0.005518	-0.4587
19	L22.E4	L24.E3	-0.157227	-0.159713	-0.0158
20	L23.E4	L24.E3	0.063477	0.063839	0.0057
21	L15.E1	L24.E4	0.017822	0.029668	0.6646
22	L15.E4	L24.E4	0.006348	0.008691	0.3691
23	L22.E4	L24.E4	0.294922	0.298125	0.0109
24	L23.E4	L24.E4	-0.100586	-0.103095	-0.0249

Signed deviation = (finite − derivative)/|derivative| in this export; it may be large for small derivatives. This is not an acceptance rate. **J05**

Figure 13: Executed write shares and soft preferences (continued).

Cost accounting for conditional computation. Conditional execution reduces the parameters touched by a given token, but the realised benefit depends on kernel shapes, dispatch overhead, and memory traffic, which is why sparse layers are a capacity mechanism before they are an efficiency guarantee (Shazeer et al., 2017; Fedus et al., 2022). DeMoE follows the same discipline: the active-width ratio in Table 1 is a structural description of the configuration, and we therefore make no latency, memory, energy, or total-FLOP claim, because no matched measurement exists.

Mode collapse, diversity preservation, and particle guidance. A persistent vulnerability in score distillation and one-step compression is mode collapse and diversity loss (Wang et al., 2024; Cai et al., 2026; Sargent et al., 2025). Remedies include incorporating explicit diversity-preserving regularizers into distillation objectives (Wu et al., 2026). Complementary sampling-stage approaches incorporate inter-particle repulsion and particle guidance to preserve generative variance (Corso et al., 2024; Vinograd et al., 2026; Sadat et al., 2024; Zilberstein et al., 2025; Blessing et al., 2025), alongside dual-expert and swift distillation designs (Li et al., 2026; Nguyen & Tran, 2024).