BEYOND STATIC IMPORTANCE: NOISE-AWARE EXPERT PRUNING FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Expert pruning has been widely studied for autoregressive Mixture-of-Experts (MoE) language models as an effective way to reduce memory overhead by removing less important experts. However, pruning methods developed for autoregressive models do not transfer well to diffusion language models (dLLMs), which perform multi-step denoising and may rely on different experts across noise levels, unlike autoregressive models that generate text along a single left-to-right trajectory. Our analysis shows that routing and activation statistics can mis-rank experts under the denoising objective, while expert importance varies substantially across noise levels, with different rankings at high and low noise levels. Consequently, evaluating experts under a single input distribution or fixed noise level may retain redundant experts while pruning those needed at other denoising stages. Motivated by these observations, we propose **NAPE**, a one-shot pruning criterion that scores each expert using the absolute gradient of the masked-prediction loss with respect to its gate and aggregates the score over mask ratios spanning high- to low-noise stages. By aggregating expert importance across noise levels, NAPE avoids biased estimates from calibration at fixed noise. Experiments on two MoE dLLM backbones across ten downstream benchmarks show that NAPE consistently outperforms existing pruning methods under different pruning ratios, improving the strongest baseline by up to 12.3% relative at 50% sparsity. We release our code at `https://anonymous.4open.science/r/NAPE-35C2` to facilitate future research on efficient MoE dLLMs.

1 INTRODUCTION

Diffusion language models (dLLMs) have recently emerged as a promising alternative to autoregressive LLMs (arLLMs) by replacing left-to-right generation with iterative denoising. Starting from a heavily masked sequence, a dLLM predicts many tokens in parallel at each step, which enables flexible infilling and a favorable trade-off between generation quality and latency (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). Recent models such as LLaDA (Nie et al., 2025), Dream (Ye et al., 2025), BD3-LM (Arriola et al., 2025), and SDAR (Cheng et al., 2026) demonstrate the scalability and potential of dLLMs.

As dLLMs scale up, they have turned to sparse Mixture-of-Experts (MoE) layers, the architecture behind most frontier arLLMs (Shazeer et al., 2017; Fedus et al., 2022; Dai et al., 2024). By routing each token to only a few experts, MoE lets a model hold far more parameters than it pays for at each forward pass. However, sparse activation does not translate into a small deployment overhead. Although only a few experts fire per token, *all* experts must remain resident in memory during inference, and serving such a model therefore requires holding several times its active parameters. Expert pruning, which permanently removes a subset of experts, has proven effective at reducing this memory cost on MoE arLLMs, retaining most of the original performance even at high pruning ratios (Lu et al., 2024; Xie et al., 2024; Muzio et al., 2024; Lasby et al., 2026), yet it has received little attention and remains largely unexplored for dLLMs.

A natural question is whether arLLM pruning criteria can be directly applied to dLLMs. Existing methods typically rank experts using forward-pass statistics collected from a single input distribution (Muzio et al., 2024; Lasby et al., 2026; Jaiswal et al., 2025; Lu et al., 2024), implicitly treating expert importance as static across inputs and independent of masked-token prediction quality. This

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.

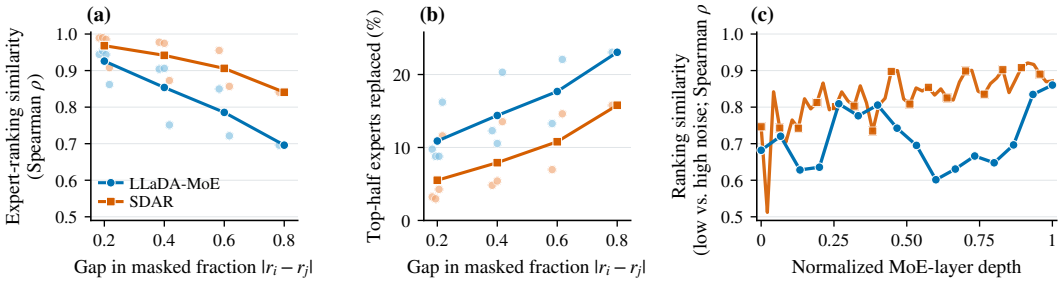


Figure 1: **Expert importance shifts across noise levels.** We independently compute gate-gradient saliency across noise levels and compare the resulting expert rankings within each layer. (a) Ranking consistency decreases as the noise gap increases. (b) The selected expert sets differ increasingly across noise levels. (c) Ranking drift is observed throughout the network.

assumption may not hold for dLLMs, which are queried across an entire denoising trajectory. Prior work shows that different noise levels require specialized parameters or varying model capacity in image diffusion and dLLMs (Balaji et al., 2022; Hang et al., 2023; Park et al., 2024a; Sedykh et al., 2026), but whether MoE experts exhibit similar specialization remains unknown. To answer this question, we measure expert saliency independently at different noise levels and compare the resulting per-layer rankings and top-half selections. As Figure 1 shows, the rankings become less consistent and the selected expert sets change as the noise gap increases. Between the lowest and highest noise levels, 16–23% of the retained experts differ per layer across the two backbones, and this variation appears throughout the network. These results suggest that expert importance is not fixed along the denoising trajectory. A criterion calibrated at a single noise level may therefore retain experts useful at that stage while pruning others needed elsewhere in denoising.

Motivated by this observation, we propose **NAPE** (Noise-Aware Pruning of Experts), a noise-aware expert pruning method for MoE dLLMs. NAPE measures how much each expert matters to the diffusion objective itself. It corrupts the calibration data at sampled noise levels spanning high- to low-noise stages, and takes the first-order gradient of the masked-prediction loss with respect to a differentiable gate attached to each expert, so that a large gradient magnitude indicates strong local sensitivity of the denoising loss to the expert (LeCun et al., 1989; Molchanov et al., 2019). Gradient magnitudes are then accumulated across all sampled states, letting every denoising stage contribute to the final score while oppositely signed effects are kept from cancelling, and the lowest-scoring experts are removed. In this way, NAPE captures expert importance across the denoising trajectory and avoids the biased estimates caused by calibration at a fixed noise level.

We evaluate NAPE on two MoE dLLMs across ten downstream benchmarks and compare against recent competitive expert pruning methods. NAPE attains the best average accuracy across models and pruning ratios, and outperforms the strongest baseline by 7.15 points on SDAR-30B-A3B when 50% of the experts are pruned. The results further indicate that the criterion hierarchy established on arLLM models does not transfer to dLLMs: gradient saliency, less commonly used in arLLM expert pruning, consistently surpasses the activation-based scores that dominate it.

In summary, our contributions are threefold. (1) We conduct the first study of expert pruning for MoE dLLMs, showing that expert importance varies substantially with the noise level and that criteria built on routing and activation statistics mis-rank experts under the diffusion objective. (2) We then propose NAPE, which scores each expert by the absolute gradient of the diffusion loss, aggregated over calibration states drawn at different noise levels. (3) Finally, extensive experiments across different pruning ratios and two MoE dLLMs demonstrate that NAPE consistently outperforms recent expert pruning methods, highlighting the importance of noise-aware expert saliency for efficient compression of MoE diffusion language models.

2 Related Work

Diffusion language models. Discrete diffusion models generate text by iteratively denoising a corrupted sequence (Austin et al., 2021; Lou et al., 2024), and masked diffusion has become the standard formulation in recent work (Sahoo et al., 2024; Shi et al., 2024). LLaDA (Nie et al., 2025)

and Dream (Ye et al., 2025) scale this paradigm to match arLLM models of similar size, and blockwise hybrids such as BD3-LM (Arriola et al., 2025) and SDAR (Cheng et al., 2026) decode blocks autoregressively while denoising tokens within a block in parallel. MoE variants have followed, trained from scratch as in LLaDA-MoE (Zhu et al., 2025) or converted from an arLLM MoE backbone as in SDAR. Efficiency efforts on dLLMs have so far targeted inference, accelerating decoding through caching and parallelism (Liu et al., 2025; Wu et al., 2025), while compressing the parameters of MoE dLLMs has not been systematically studied.

Expert pruning for MoE LLMs. MoE architectures scale LLMs by activating only a few experts per token (Shazeer et al., 2017; Jiang et al., 2024; Dai et al., 2024), but all experts must stay in memory, which has motivated a line of work on expert-level compression. Most pruning methods score experts by routing statistics or reconstruction error (Lu et al., 2024; Xie et al., 2024; Muzio et al., 2024). REAP (Lasby et al., 2026) combines router gate values with activation norms and shows that pruning beats merging on generative tasks. What these methods share is the reliance on routing and activation statistics collected from one input distribution, a reasonable abstraction for arLLM models, where every position carries a loss term and inference sees clean context alone. Neither premise holds for dLLMs, whose loss reads only masked positions and whose inference spans a range of noise levels. Gradient-based Taylor importance has a long history in structured pruning (LeCun et al., 1989; Molchanov et al., 2019; Michel et al., 2019), yet has received limited attention in arLLM expert pruning; we show that on dLLMs it becomes the decisive signal, and estimate it across the noise levels the model serves.

Noise-level specialization in diffusion models. Diffusion models denoise across a wide range of noise levels, and a growing body of work finds that different levels demand different computation. In image diffusion, eDiff-I (Balaji et al., 2022) trains an ensemble of expert denoisers over noise intervals; timesteps have been shown to conflict as competing training tasks (Hang et al., 2023; Go et al., 2023), and architectures such as DTR (Park et al., 2024b) and Switch-DiT (Park et al., 2024a) route computation by timestep to exploit this structure. For dLLMs, step-level heterogeneity has recently been observed as well (Sedykh et al., 2026). All of these works target training or architecture design. We show instead that analogous noise-level dependence emerges among the experts of pretrained MoE dLLMs, and exploit it to build a noise-aware pruning criterion.

3 METHOD

3.1 PRELIMINARIES AND PROBLEM SETUP

Masked diffusion language models. Given a clean sequence x_0, the forward process samples a noise level $t \in (0, 1]$ and independently replaces each token of x_0 with a mask token $\mathbf{M}$ with probability t, producing x_t. Let $\mathcal{M}_t = \{\, j : x_t^j = \mathbf{M} \,\}$ denote the masked positions. The model p_θ is trained to recover the masked tokens with

$$\mathcal{L}_{\text{diff}}(\theta) = -\,\mathbb{E}\Big[\tfrac{1}{t}\textstyle\sum_{j\in\mathcal{M}_t} \log p_\theta\big(x_0^j \mid x_t\big)\Big], \tag{1}$$

where the expectation is over x_0, t, and x_t (Sahoo et al., 2024; Nie et al., 2025). Inference proceeds along a denoising trajectory from high to low noise; for blockwise dLLMs, the same process is applied within each block conditioned on preceding blocks.

MoE layers. In a MoE model, an MoE layer holds N experts and a router that activates only a few of them per token during sparse inference, so that the model can hold far more parameters than it uses at each forward pass; we describe one such layer and omit the layer index throughout for notational simplicity. Each MoE layer contains N experts $\{E_i\}_{i=1}^{N}$, where E_i denotes the i-th expert, and a router that produces gate values $g_i(h)$ for a hidden state h; its output is $\sum_{i\in\mathcal{R}(h)} g_i(h)\, E_i(h)$, where $\mathcal{R}(h)$ is the set of top-k routed experts. Throughout, k is the number of experts activated per token and K the number of experts retained after pruning.

Target. Given a target ratio, our goal is to select for every MoE layer a subset $\mathcal{S} \subseteq [N]$ with $|\mathcal{S}| = K$, such that removing the remaining $N - K$ experts degrades the model as little as possible. Layers are pruned independently with a uniform ratio. After pruning, the logits of the removed experts are excluded, and the router softmax is recomputed over the remaining experts.

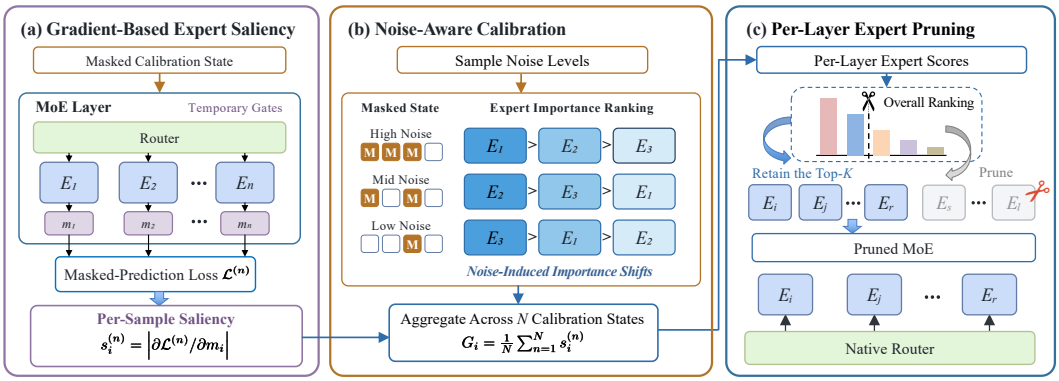


Figure 2: **Overview of NAPE.** (a) Gradients of the masked-prediction loss with respect to temporary per-expert gates yield per-sample saliency. (b) Repeating this computation at sampled noise levels captures noise-dependent expert rank shifts and accumulates a score for each expert. (c) The top-K experts in each layer are retained and the rest are removed.

3.2 Gradient-Based Expert Saliency

This subsection introduces a gradient-based criterion that measures expert importance through its effect on the masked-prediction loss. Gradient-based criteria have a long history in structured pruning (LeCun et al., 1989; Molchanov et al., 2019), but are rarely used for MoE expert pruning in ar-LLMs, where inexpensive forward-pass statistics have already proven effective (Jaiswal et al., 2025; Lasby et al., 2026). The same choice may be less suitable for masked dLLMs, whose loss is defined specifically over masked targets. Routing and activation statistics summarize the forward pass without measuring each expert's effect on this objective, and may therefore assign high importance to experts whose outputs have little impact on the masked-prediction loss.

Gate-gradient saliency. To measure this loss sensitivity, we attach a differentiable gate m_i to each expert E_i: $\tilde{o}_i = m_i o_i,\ o_i = g_i(h)E_i(h)$, where $m_i = 1$ leaves the expert unchanged and $m_i = 0$ removes its contribution. For a calibration sample $x^{(n)}$ at noise level $t^{(n)}$, let $\mathcal{L}^{(n)}$ denote the cross entropy averaged over its masked positions. Unlike equation 1, this per-state loss is normalized by the number of masked targets. We define:

$$\delta_i^{(n)} = \frac{\partial \mathcal{L}^{(n)}}{\partial m_i}, \qquad s_i^{(n)} = \left|\delta_i^{(n)}\right|, \tag{2}$$

where $\delta_i^{(n)}$ is the signed gate derivative and $s_i^{(n)}$ is the per-sample saliency of expert i. Only scalar-gate gradients are stored, not model-weight gradients. This follows the Taylor criterion of Molchanov et al. (2019), applied to expert gates under the masked-prediction loss.

What gate gradients capture. The key advantage of gate gradients is that they measure not only expert usage and output magnitude, but also alignment with the loss gradient. For a fixed calibration sample, write $y_j = \sum_i m_i o_{i,j}$ for the MoE output at position j, where $o_{i,j} = \mathbf{1}[i \in \mathcal{R}(h_j)]\, g_i(h_j)E_i(h_j)$ is the routed contribution of expert E_i at position j. Since y_j is linear in m_i, the chain rule gives

$$\delta_i^{(n)} = \sum_j \langle u_j,\, o_{i,j} \rangle, \qquad u_j = \frac{\partial \mathcal{L}^{(n)}}{\partial y_j}. \tag{3}$$

Each term depends on $\|u_j\|\,\|o_{i,j}\| \cos \angle(u_j, o_{i,j})$, combining whether the expert is routed, its output magnitude, and its alignment with the loss gradient. Routing frequency captures only whether an expert is selected, while activation-based scores additionally capture output magnitude. Neither accounts for the alignment term, which depends on u_j rather than on the expert output alone. Note that u_j need not vanish at unmasked positions, since gradients from masked targets can propagate through them via attention.

Why activation magnitude is insufficient. A large expert output does not imply high loss sensitivity if it is poorly aligned with the loss gradient. As a simple counterexample, consider one position with $u_j = (1, 0)$, $o_{1,j} = (0, C)$, and $o_{2,j} = (1, 0)$, embedded in the model dimension by zero padding. Activation magnitude ranks expert 1 above expert 2 by a factor of C, which can be

Algorithm 1 NAPE expert pruning

Require: calibration set $\mathcal{D}$, experts $[N]$, budget K, noise distribution q
1: attach gates $m_i \leftarrow 1$; $G_i \leftarrow 0$ for all $i \in [N]$ ▷ Initialize scores
2: **for** each clean calibration sample $x \in \mathcal{D}$ **do**
3: sample $t \sim q$ and construct a corrupted state x_t ▷ Sample noise level
4: compute $\mathcal{L}(x_t)$ and backpropagate to the gates ▷ Loss-aware sensitivity
5: $G_i \leftarrow G_i + |\mathcal{D}|^{-1}\left|\partial\mathcal{L}(x_t)/\partial m_i\right|$ **for all** i ▷ Accumulate saliency
6: **end for**
7: retain the K experts with the largest G_i and remove the rest ▷ Prune experts

arbitrarily large, whereas their gradient saliencies are 0 and 1, respectively. The same distinction also appears near the empirical pruning boundary: at 50% sparsity, activation- and gradient-based selection differ by six to seven experts per layer.

Aggregation without sign cancellation. Because signed gate derivatives may cancel across calibration samples, we aggregate their magnitudes:

$$G_i = \mathbb{E}_n\big[s_i^{(n)}\big] = \mathbb{E}_n\big[|\delta_i^{(n)}|\big], \tag{4}$$

where $\mathbb{E}_n$ denotes the average over calibration samples. Under bounded curvature, the gate derivative gives the local Taylor approximation to the loss change induced by removing expert E_i, so $s_i^{(n)}$ measures the magnitude of the corresponding loss change on sample n. Taking the magnitude before averaging prevents effects with opposite signs from cancelling across samples.

Empirical evidence of cancellation. Sign cancellation is observed on both backbones. The median ratio $G_i/|\mathbb{E}_n[\delta_i^{(n)}]|$ across active layer–expert pairs is 9.0 on LLaDA-MoE and 6.0 on SDAR-30B-A3B. The ratio exceeds 10 for 46% and 31% of these pairs, respectively, indicating substantial cancellation under signed averaging.

3.3 Noise-Aware Calibration

The gradient-based score in equation 2 is conditioned on the corrupted state used for calibration. In a dLLM, changing the noise level changes both the masked positions and the context available for prediction, which in turn changes the loss gradient u_j in equation 3. This suggests that expert saliency may vary along the denoising trajectory, making calibration at a single mask ratio potentially unrepresentative.

To examine this effect, we compute expert saliency at $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ on 128 C4 Allen Institute for AI (2024) samples (4,096 tokens each) and compare the resulting rankings and retained sets. Figure 1 shows clear noise-dependent variation. On LLaDA-MoE, the median layerwise Spearman correlation decreases from 0.93 for neighbouring levels to 0.70 at the two extremes. At 50% sparsity, the corresponding retained sets differ by an average of 7.4 among the 32 experts kept per layer. SDAR-30B-A3B shows the same pattern, with a correlation of 0.84 between the extreme levels and an average difference of 10.1 among the 64 retained experts. The variation therefore extends to experts near the pruning boundary, where ranking differences directly affect selection.

These observations motivate calibrating over a distribution of noise levels rather than a single one. Let q denote the noise-level distribution used for calibration, and let $\mu_i(t)$ be the conditional saliency of expert i at noise level t, defined as the average of $s_i^{(n)}$ over calibration states at that level. Since the same pruned model is used throughout denoising, we define the noise-marginal saliency as

$$\bar{G}_i = \mathbb{E}_{t\sim q}\big[\mu_i(t)\big]. \tag{5}$$

NAPE estimates $\bar{G}_i$ by sampling calibration noise levels from q. The resulting average per-state saliency is an unbiased estimate of $\bar{G}_i$ and converges to it as the calibration set grows. By contrast, calibration at a fixed level t_0 converges to $\mu_i(t_0)$, leaving the persistent bias $\mu_i(t_0) - \bar{G}_i$.

For each calibration sample, NAPE samples a noise level from q, constructs the corresponding corrupted state, and accumulates the absolute gate gradient for every expert. Each layer then retains the K highest-scoring experts and removes the rest together with their router logits. The router

softmax is recomputed over the retained experts. The procedure requires one backward pass per calibration state and no fine-tuning after pruning. Algorithm 1 summarizes the full procedure.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and baselines. We evaluate on two representative open-source MoE dLLMs with different decoding schemes. LLaDA-MoE-7B-A1B-Instruct (Zhu et al., 2025) is pretrained from scratch with 16 MoE layers, 64 experts per layer, and top-8 routing (7B total, 1.4B active), and denoises the full sequence under bidirectional attention. SDAR-30B-A3B-Chat (Cheng et al., 2026) is adapted from the arLLM backbone Qwen3-30B-A3B (Yang et al., 2025) with 48 MoE layers, 128 experts per layer, and top-8 routing (30B total, 3B active), and decodes 32-token blocks under blockwise-causal attention. We compare NAPE against the strongest expert pruning methods currently available for MoE LLMs, Frequency (routing counts), Expert Activation Norm (EAN) (Jaiswal et al., 2025), and REAP (gate-weighted activation norm) (Lasby et al., 2026), together with a Random baseline. All criteria are applied in an identical pipeline so that differences reflect only expert selection.

Calibration and evaluation. We use C4 (Allen Institute for AI, 2024) as the calibration dataset, which is commonly used for model compression (Frantar & Alistarh, 2023). For each model, saliency is computed from 128 calibration samples of 4,096 tokens, about 0.5M tokens in total. Following previous MoE pruning research (Zhang et al., 2026; Lasby et al., 2026), we adopt lm-evaluation-harness (Gao et al., 2024) to measure zero-shot accuracy on ten downstream tasks: ARC-C/ARC-E (Clark et al., 2018), BoolQ (Clark et al., 2019), COPA (Roemmele et al., 2011), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), RTE (Bentivogli et al., 2009), and WinoGrande (Sakaguchi et al., 2021), reporting length-normalized accuracy where standard. Since dLLMs define no autoregressive likelihood, each option is scored by a Monte Carlo estimate of the conditional ELBO following Nie et al. (2025), masking only the answer tokens with stratified mask-ratio sampling over 16 passes; further details are provided in Appendix D. The Monte Carlo mask samples are fixed across methods, ensuring paired and directly comparable results across pruning criteria.

Implementation details. We implement NAPE and all baseline criteria in PyTorch (Ansel et al., 2024). The router logits and expert activations required by the baseline criteria are collected in the same calibration pass as our gradient statistics, so all methods share identical calibration inputs. All pruning is performed in a one-shot manner, and no fine-tuning or weight update is applied after pruning. Pruned experts are physically deleted from the checkpoint, and the router softmax is recomputed over the remaining experts. Following common practice in MoE expert pruning (Lasby et al., 2026; Zhang et al., 2026), we test two pruning ratios, 25% and 50%, applied uniformly to every layer on both models. Noise levels are sampled from $q = \mathcal{U}(0.1, 0.9)$ to cover the denoising trajectory while avoiding endpoints with too few masked targets or too little visible context. All experiments are conducted on NVIDIA RTX 6000 Ada Generation GPUs.

4.2 MAIN RESULTS

Tables 1 and 2 report the results of LLaDA-MoE-7B-A1B and SDAR-30B-A3B, respectively. Across ten downstream benchmarks and two pruning ratios, NAPE attains the best average accuracy in every model and pruning ratio. We summarize three key observations below.

(1) NAPE outperforms the activation- and routing-based criteria that dominate arLLM expert pruning. It attains the best average accuracy in all four model and ratio settings, including 69.69 on SDAR-30B-A3B at 25% sparsity against 69.40 for Frequency and 68.75 for EAN, and 60.25 on LLaDA-MoE-7B-A1B against 59.48 for EAN, the strongest baseline there. All methods use the same calibration data, pruning procedure, and evaluation protocol, isolating the effect of the pruning criterion. NAPE differs by measuring expert sensitivity to the diffusion objective across noise levels.

(2) The gains are largest at higher sparsity when sufficient expert redundancy remains. On SDAR-30B-A3B at 50% sparsity, NAPE leads the strongest baseline by 7.15 points (65.23 vs. 58.08 for EAN) and retains 93.0% of the dense accuracy with half the experts removed, ranking among the

Table 1: Performance of LLaDA-MoE-7B-A1B-Instruct under different expert pruning ratios. **Bold** numbers indicate that NAPE outperforms the strongest baseline.

Pruning Ratio	Model/Method	ARC-C	ARC-E	BoolQ	COPA	HellaSwag	MMLU	OBQA	PIQA	RTE	WinoG	Avg.
0%	LLaDA	45.99	68.64	76.64	77.00	60.84	60.94	30.20	67.46	81.59	59.19	62.85
25%	Random	39.51	59.51	66.64	69.00	54.22	51.55	28.60	63.22	76.90	55.33	56.45
	Frequency	32.94	45.37	76.54	66.00	54.95	50.95	29.40	63.33	62.82	56.35	53.87
	EAN	42.32	59.26	71.74	78.00	60.80	55.68	30.40	66.10	72.20	58.33	59.48
	REAP	40.53	55.39	78.32	77.00	59.62	56.30	30.20	64.58	70.04	57.38	58.94
	NAPE (Ours)	41.47	**61.83**	74.74	**79.00**	**61.07**	52.98	**31.40**	**67.41**	74.01	**58.56**	**60.25**
50%	Random	29.78	42.42	45.66	57.00	39.74	36.95	26.80	57.67	64.26	50.59	45.09
	Frequency	27.56	37.50	62.17	65.00	44.25	33.29	26.80	59.68	59.57	53.59	46.94
	EAN	31.31	40.95	66.45	67.00	50.82	29.58	26.40	59.96	59.57	56.04	48.81
	REAP	31.40	41.33	52.23	67.00	50.16	28.36	27.40	60.12	59.93	55.96	47.39
	NAPE (Ours)	**31.74**	**43.94**	53.79	**70.00**	**54.49**	28.41	**29.60**	**63.33**	58.84	55.88	**49.00**

Table 2: Performance of SDAR-30B-A3B-Chat under different expert pruning ratios. **Bold** numbers indicate that NAPE outperforms the strongest baseline.

Pruning Ratio	Model/Method	ARC-C	ARC-E	BoolQ	COPA	HellaSwag	MMLU	OBQA	PIQA	RTE	WinoG	Avg.
0%	SDAR	49.57	74.07	87.65	84.00	68.33	78.40	41.60	69.15	81.59	67.25	70.16
25%	Random	48.46	71.55	80.95	78.00	59.71	68.52	37.00	65.78	78.70	62.27	65.09
	Frequency	50.00	75.04	87.92	83.00	68.05	73.44	40.20	69.75	79.06	67.56	69.40
	EAN	49.40	74.28	87.43	78.00	67.93	73.65	41.20	69.75	79.06	66.77	68.75
	REAP	48.63	71.25	86.76	81.00	64.35	71.31	38.60	69.80	79.42	66.46	67.76
	NAPE (Ours)	**50.09**	73.48	87.71	**85.00**	67.47	**74.53**	39.60	69.53	**80.87**	**68.59**	**69.69**
50%	Random	39.68	65.28	56.70	71.00	45.42	42.15	34.40	61.59	72.20	56.43	54.49
	Frequency	30.89	45.12	63.24	76.00	35.73	23.92	31.00	58.71	63.90	52.88	48.14
	EAN	37.71	54.34	72.75	68.00	65.93	57.77	30.00	70.46	67.87	55.96	58.08
	REAP	35.75	56.94	79.63	72.00	54.34	49.39	32.20	64.69	74.37	59.51	57.88
	NAPE (Ours)	**43.86**	**68.56**	**87.03**	**84.00**	64.67	55.68	**38.40**	69.21	72.56	**68.35**	**65.23**

top two pruned methods on all ten tasks. The larger gain at 50% sparsity is consistent with the ranking differences observed in our analysis. As fewer experts are retained, differences between pruning criteria are more likely to affect which experts remain near the selection boundary. LLaDA-MoE-7B-A1B has a smaller expert pool, and at 50% sparsity no method retains more than 78% of the dense average. The gap between pruning criteria is therefore smaller, although NAPE still achieves the best average and leads on six of the ten tasks.

(3) NAPE remains consistent across models and pruning ratios. The two models represent different dLLM designs, with blockwise-causal decoding in SDAR and fully bidirectional masked diffusion in LLaDA-MoE, while the ten tasks cover a broad range of reasoning and language understanding benchmarks. The strongest baseline changes across models, pruning ratios, and tasks. Frequency, for example, is the strongest baseline on SDAR-30B-A3B at 25% sparsity with 69.40, but drops to 48.14 at 50%, below Random selection at 54.49. Activation- and routing-based criteria summarize expert usage without accounting for its variation across noise levels, which can make their rankings sensitive to the model and evaluation setting. A criterion that transfers across these settings is especially useful for dLLMs, where expert importance changes with the denoising process itself. NAPE ranks first on average in all four settings and on 23 of the 40 task-level comparisons, more than any baseline. Its consistent performance across both backbones and a diverse set of tasks suggests that the gains are not specific to a single model or benchmark.

Memory Efficiency. Table 3 shows that pruning 50% of the experts reduces loading memory by 45.7% on SDAR-30B-A3B (58.25 GB to 31.65 GB), fitting the 30B model on a single 48 GB GPU, and by 41.5% on LLaDA-MoE; at 25% sparsity the savings are 22.9% and 20.8%, respectively. Since all criteria remove the same number of experts, these savings are shared by every method and confirm expert pruning as an effective way to reduce the memory cost of MoE dLLMs during deployment.

Table 3: **Peak GPU memory for loading dense and physically pruned checkpoints.** Models are loaded in BF16. Expert pruning substantially reduces model loading memory, achieving up to 45.7% savings at 50% sparsity.

Model	Dense	25% Sparsity		50% Sparsity	
	Memory	Memory	Saving	Memory	Saving
LLaDA	15.55 GB	12.32 GB	20.8%	9.10 GB	41.5%
SDAR	58.25 GB	44.92 GB	22.9%	31.65 GB	45.7%

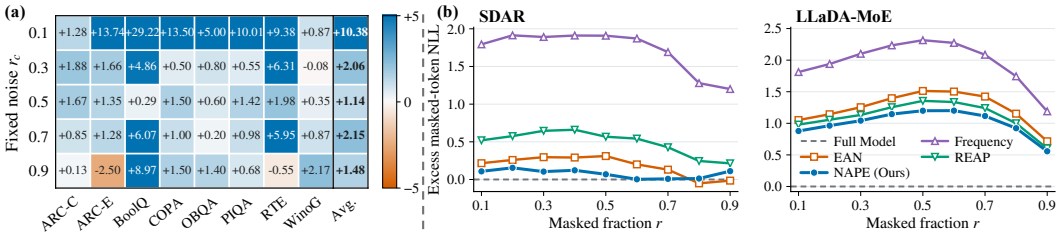


Figure 3: **(a) Fixed-noise calibration is sensitive to the selected noise level.** Each cell reports NAPE minus fixed-noise accuracy at 50% expert pruning, averaged over LLaDA-MoE and SDAR; blue favors NAPE, orange the fixed ratio. NAPE has the higher average at every tested ratio. **(b) Masked-token NLL after pruning on held-out WikiText-2 data.** Excess NLL relative to the full model across nine mask ratios; lower values indicate better performance.

4.3 Ablation Study

In this section, we separately examine the contributions of gradient-based saliency and multi-noise calibration. For these comparisons, we report the average over eight tasks and omit MMLU and HellaSwag because their large evaluation sets make likelihood evaluation computationally expensive.

Effect of gradient-based saliency. We first test whether the gain comes from the scoring signal itself rather than from noise coverage. We use the same calibration samples and masked target positions across methods, varying only the expert-importance criterion. As shown in Table 4 and Table 9, NAPE outperforms the strongest baseline by 4.52 points on LLaDA and 5.05 points on SDAR at 50% pruning. The gains show that gate gradients capture information beyond activation magnitude on masked targets.

Table 4: **Masked-target score ablation at 50% pruning.** We report the average performance over eight tasks; **bold** marks the best result.

Method	Signal	LLaDA	SDAR
Masked-REAP	Route-weighted norm	46.37	57.44
Masked-EAN	Routed norm	46.19	61.45
NAPE (Ours)	Gate gradient	**50.89**	**66.50**

Effect of multi-noise calibration. We next isolate the effect of noise coverage to test whether a single noise level is sufficient for pruning. Figure 3(a) compares multi-noise calibration with five fixed noise values at 50% pruning, averaging results over LLaDA-MoE and SDAR (full results are available in Table 10). These results clearly show that multi-noise calibration achieves a higher overall mean than every fixed value. The best fixed value varies across tasks and models, consistent with expert importance changing across denoising. Aggregating saliency across noise levels therefore gives better overall pruning performance in practice.

4.4 Analysis and Discussion

Denoising behavior. Since expert importance varies across noise levels, evaluating pruning performance throughout denoising can provide a direct view of this effect beyond downstream benchmark accuracy and reveal where pruning errors emerge. Using 64 held-out 512-token segments from WikiText-2 (Merity et al., 2017), we construct corrupted inputs at $r \in \{0.1, \ldots, 0.9\}$ and evaluate all methods on the same inputs. Figure 3(b) reports the excess masked-token NLL relative to the full model. At 50% sparsity, NAPE achieves the lowest normalized noise-AUC among all pruned methods on both models, with AUC lower than the strongest baseline by 0.12 on SDAR and 0.11 on LLaDA-MoE. This indicates that accounting for expert importance across noise levels better preserves the masked-prediction objective throughout denoising and reduces degradation across different corruption levels.

Free-form generation. Apart from answer-selection benchmarks, we also evaluate NAPE on XSum (Narayan et al., 2018), an abstractive summarization task requiring concise summaries of news articles, to assess whether its advantage extends to open-ended generation. At 50% expert sparsity, NAPE achieves the highest ROUGE scores among the pruned methods on both backbones (Table 5). It improves over the strongest pruned baseline by up to 1.11 ROUGE points on LLaDA and 0.78 points on SDAR, while remaining close to the full model on SDAR.

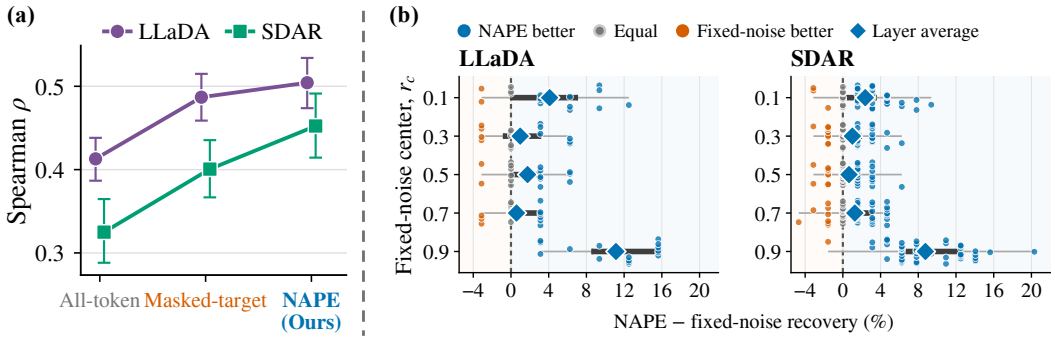


Figure 4: **Masked-target gradients and multi-noise calibration.** **(a)** Spearman correlation between each score and the held-out loss change caused by removing one expert from the final MoE layer. Error bars show 95% bootstrap intervals over C4 sources. **(b)** Difference in retained-half recovery between NAPE and fixed-noise calibration, using a disjoint noise-marginal ranking as reference. Points denote layers and diamonds the mean across layers; positive values favor NAPE.

What activation scores miss. An effective way to assess whether a pruning score captures loss sensitivity is to compare it with the loss change caused by removing each expert. Figure 4(a) reports this comparison for three scoring criteria using held-out single-expert removals. Using held-out C4 samples corrupted at mask ratios drawn from $r \in [0.1, 0.9]$, we remove each expert in turn from the final MoE layer of LLaDA and SDAR, recompute top-k routing, and record the resulting loss change. We then compute the Spearman correlation between each score and these removal effects over experts used by at least one masked target. Both models show the same trend. Restricting activation to masked targets improves agreement with the removal effects, while NAPE yields a further gain by accounting for the alignment between expert outputs and the loss gradient. Relative to all-token activation, NAPE increases ρ by 0.091 on LLaDA and 0.127 on SDAR.

Table 5: **Free-form summarization on XSum under 50% expert sparsity. Bold** marks the best pruned result; scores are on a 0–100 scale.

	ROUGE-1 (↑)		ROUGE-L (↑)	
Method	LLaDA	SDAR	LLaDA	SDAR
Full Model	23.02	24.43	16.26	18.37
Frequency	17.36	5.94	12.38	5.37
EAN	18.67	23.96	13.17	17.65
REAP	20.22	20.23	14.22	14.46
NAPE (Ours)	**21.33**	**24.15**	**15.05**	**18.43**

Why noise coverage matters. Agreement with removal effects does not show whether calibration at one noise level can represent expert importance across denoising. We therefore compare multi-noise calibration with five fixed noise levels, $r_c \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, using the same gradient score and calibration budget. Each ranking is evaluated by how well its top-half expert set recovers the set selected by a separate, higher-budget noise-marginal reference. As Figure 4(b) shows, NAPE achieves higher mean layerwise recovery than every fixed noise level on both models. Averaged across the five fixed levels, the gain is 3.71 percentage points on LLaDA and 2.81 points on SDAR. The largest gap occurs at $r_c = 0.9$, where the ranking differs most from the noise-marginal reference.

5 Conclusion

In this work, we presented what is, to our knowledge, the first study of expert pruning for MoE diffusion language models. We showed that expert importance varies substantially with the noise level, and that the activation- and routing-based criteria dominating arLLM expert pruning mis-rank experts under the diffusion objective. Building on this observation, we proposed NAPE, which scores each expert by absolute gradients of the diffusion loss aggregated over calibration states drawn at different noise levels. Across two MoE dLLMs, NAPE consistently outperforms existing pruning criteria, improving the strongest baseline by up to 12.3% relative at 50% sparsity, while roughly halving deployment memory. We see next steps in adaptive per-layer budgets and in combining pruning with expert merging, quantization, and runtime acceleration.

Reproducibility Statement

We provide the model and baseline details, evaluation protocol, calibration configuration, and additional sensitivity analyses in the appendix. The main text specifies the pruning procedure, calibration data, noise distribution, pruning ratios, and evaluation benchmarks used in our experiments. We additionally provide anonymized source code and evaluation scripts in the supplementary material.

AI Use Statement

All experiments, analyses, visualizations, and scientific conclusions were produced by the authors. Large language models (e.g., ChatGPT) were used only to refine the writing, including grammar, clarity, and presentation. They were not used to design the methodology, generate experimental results, or conduct data analysis. All LLM-assisted edits were reviewed by the authors, who take full responsibility for the final manuscript.

References

Allen Institute for AI. allenai/c4. `https://huggingface.co/datasets/allenai/c4`, 2024. Hugging Face dataset.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM international conference on architectural support for programming languages and operating systems, volume 2*, pp. 929–947, 2024.

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference (TAC)*, 2009.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22058–22075, 2026.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 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.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Hyojun Go, Kim Kim, Yunsung Lee, Seunghyun Lee, Shinhyeok Oh, Hyeongdon Moon, and Seungtaek Choi. Addressing negative transfer in diffusion models. *Advances in Neural Information Processing Systems*, 36:27199–27222, 2023.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7441–7451, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Ajay Jaiswal, Jianyu Wang, Yixiao Li, Pingzhi Li, Tianlong Chen, Zhangyang Wang, Chong Wang, Ruoming Pang, and Xianzhi Du. Finding fantastic experts in moes: A unified study for expert dropping strategies and observations. *arXiv preprint arXiv:2504.05586*, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot moe compression. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, Yichen Zhu, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32819–32848, 2024.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6159–6172, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11264–11272, 2019.

Alexandre Muzio, Alex Sun, and Churan He. Seer-moe: Sparse expert efficiency through regularization for mixture-of-experts. *arXiv preprint arXiv:2404.05089*, 2024.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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*, pp. 461–477. Springer, 2024a.

Byeongjun Park, Sangmin Woo, Hyojun Go, Jin-Young Kim, and Changick Kim. Denoising task routing for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 10334–10358, 2024b.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pp. 90–95, 2011.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Ivan Sedykh, Nikita Sorokin, and Valentin Malykh. Not all denoising steps are equal: Model scheduling for faster masked diffusion language models. *arXiv preprint arXiv:2604.02340*, 2026.

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. In *International Conference on Learning Representations*, 2017.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Geng Zhang, Han Yuxuan, Yuxuan Lou, Yiqi Zhang, Wangbo Zhao, and Yang You. MoNE: Replacing redundant experts with lightweight novices for structured pruning of moe. In *The Fourteenth International Conference on Learning Representations*, 2026.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, et al. Llada-moe: A sparse moe diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

We organize the appendix as follows. Section A discusses several limitations in our study; Section B provides model details; Section C describes the pruning baselines; Section D presents the evaluation protocol and Monte Carlo sensitivity analysis; Section E studies robustness to calibration-set size; and Section F provides the complete ablation results.

A LIMITATIONS

Our study has several limitations. First, we focus on one-shot expert pruning without any post-pruning fine-tuning or recovery. Incorporating recovery procedures may further improve the performance of the pruned models. Second, for controlled comparison, we prune the same fraction of experts in each MoE layer. Jointly optimizing layer-wise sparsity allocation may further improve the accuracy–efficiency trade-off. Finally, NAPE requires a backward pass during calibration and is therefore primarily intended for offline expert pruning rather than online pruning settings.

B MODEL DETAILS

Table 6 summarizes the two MoE dLLMs used in our study. Routing sparsity denotes the fraction of experts inactive per token, $1 - k/N$.

Table 6: Comparison of the MoE dLLMs included in our study. †LLaDA-MoE reports non-embedding parameter counts; the full checkpoint contains 7.36B parameters.

	LLaDA-MoE	SDAR
MoE layers	16	48
Routed experts	64	128
Shared experts	0	0
Top-k	8	8
Routing sparsity	87.50%	93.75%
Reported params	7.03B†	30.5B
Reported active params	1.40B†	3.3B
Attention	global bidirectional	blockwise causal
Native denoising	global	32-token blocks

LLaDA-MoE-7B-A1B-Instruct (Zhu et al., 2025) is an instruction-tuned masked dLLM whose backbone is pretrained from scratch. Its training objective samples a noise level and masks the corresponding fraction of positions, and its native inference denoises a globally masked response under fully bidirectional attention, with optional semi-autoregressive blockwise sampling.

SDAR-30B-A3B-Chat (Cheng et al., 2026) is adapted from the arLLM backbone Qwen3-30B-A3B-Base (Yang et al., 2025) through continued pre-training and supervised fine-tuning. It decodes 32-token blocks under blockwise-causal attention, causal across blocks and bidirectional within the current block; we use the b32 checkpoint.

In our experiments, both models use their native top-8 routers, and expert parallelism is not used.

C BASELINE DETAILS

We describe the baseline pruning criteria used in our experiments. For a hidden state h_j at position j, let $\mathcal{R}(h_j)$ be the set of experts selected by the router, $g_i(h_j)$ the gate value of expert i, and $E_i(h_j)$ its output. Let $\mathcal{J}_i = \{(n, j) : i \in \mathcal{R}(h_j^{(n)})\}$ collect the positions at which expert i is routed and $F_i = |\mathcal{J}_i|$.

Random. A uniformly random subset of K experts is retained in every layer, providing a criterion-free reference.

Frequency. Experts are ranked by how often they are routed,

$$S_i^{\text{Freq}} = F_i. \tag{6}$$

Expert Activation Norm (EAN). EAN (Jaiswal et al., 2025) ranks experts by the accumulated norm of their outputs over routed positions,

$$S_i^{\text{EAN}} = \sum_{(n,j)\in\mathcal{J}_i} \left\| E_i(h_j^{(n)}) \right\|_2. \tag{7}$$

REAP. REAP (Lasby et al., 2026) weights activation norms by the router gate and averages over routed positions only,

$$S_i^{\text{REAP}} = \frac{1}{F_i} \sum_{(n,j)\in\mathcal{J}_i} g_i(h_j^{(n)}) \left\| E_i(h_j^{(n)}) \right\|_2, \tag{8}$$

we set $S_i^{\text{REAP}} = 0$ when $F_i = 0$.

The three criteria use different forward-pass statistics. Frequency measures how often an expert is routed, EAN additionally incorporates output magnitude, and REAP weights output magnitude by the router gate while dividing out routing frequency. None of these criteria directly depends on the masked-prediction loss. For every method, we retain the K highest-scoring experts in each layer, delete the remaining experts together with their router logits, and renormalize the router over the surviving experts.

D Evaluation Protocol

Likelihood scoring. Let $y = (y_1, \ldots, y_T)$ be one tokenized answer continuation and let x be its context. Following Nie et al. (2025), we score a dLLM with the Monte Carlo estimate

$$\widehat{\ell}_M(y \mid x) = \frac{1}{M} \sum_{m=1}^{M} \frac{T}{K_m} \sum_{j\in\mathcal{M}_m} \log p_\theta\big(y_j \mid x, y_{\setminus\mathcal{M}_m}\big), \qquad K_m = |\mathcal{M}_m|. \tag{9}$$

The per-pass weight is $T/K_m = 1/r_m$, where $r_m = K_m/T$. We mask continuation tokens only. LLaDA uses full bidirectional attention, while SDAR retains its 32-token block-causal attention.

Mask schedule and choice of M. The code samples mask counts directly. For each continuation it draws $K_0 \sim \text{Unif}\{1, \ldots, T\}$ and uses

$$K_m = 1 + \text{mod}\left(K_0 - 1 + \text{round}_{\text{even}}\left(\frac{(m-1)T}{M}\right), T\right), \qquad m = 1, \ldots, M, \tag{10}$$

where $\text{round}_{\text{even}}$ is nearest-integer rounding with ties rounded to the even integer, as in Python.

Given K_m, the code samples $\mathcal{M}_m$ uniformly without replacement. With $M = 16$, this schedule visits every nonzero mask count at least once when $T \le 16$; when $T > 16$, it uses 16 distinct counts spread over $\{1, \ldots, T\}$. A direct audit of the harness requests shows that at least 98.8% of the choices have $T \le 16$ on eight of the ten tasks under both the LLaDA and SDAR tokenizers. The corresponding fractions are about 54% for PIQA and 24% for HellaSwag. Thus, for nearly all choices on the other eight tasks, $M = 16$ covers the full mask-count range.

Prompts and metrics. We use lm-evaluation-harness (Gao et al., 2024), zero-shot task templates, and a maximum sequence length of 4096; overlength contexts are left-truncated. Chat templates are disabled. For plain accuracy, the harness selects $\arg\max_y \widehat{\ell}_M(y \mid x)$. For length-normalized accuracy, it selects $\arg\max_y \widehat{\ell}_M(y \mid x)/c(y)$, where $c(y)$ is the character length of the raw harness choice string. We use length-normalized accuracy for ARC-C, ARC-E, HellaSwag, OpenBookQA, and PIQA, and plain accuracy for BoolQ, COPA, MMLU, RTE, and WinoGrande.

Table 7: SDAR NAPE sensitivity to the number of Monte Carlo passes. Scores are percentages under each task's standard metric. Δ is $M = 32$ minus $M = 16$; Mean is the unweighted average across tasks.

	25% pruning			50% pruning		
Task	$M = 16$	$M = 32$	Δ	$M = 16$	$M = 32$	Δ
ARC-C	50.09	51.02	+0.93	43.86	45.82	+1.96
COPA	85.00	82.00	-3.00	84.00	79.00	-5.00
OpenBookQA	39.60	40.60	+1.00	38.40	39.20	+0.80
PIQA	69.53	72.25	+2.72	69.21	71.22	+2.01
RTE	80.87	80.14	-0.73	72.56	67.15	-5.41
WinoGrande	68.59	71.98	+3.39	68.35	72.06	+3.71
Mean	65.61	66.33	+0.72	62.73	62.41	-0.32

Table 8: **Sensitivity to calibration-set size.** Task accuracy (%) at 50% expert pruning using nested C4 calibration sets. N denotes the number of calibration samples, each containing 4096 tokens; averages use unrounded task scores, and bold marks the best average at each N within a model.

	Frequency			Masked-EAN			Masked-REAP			**NAPE (Ours)**		
Task	$N = 32$	$N = 64$	$N = 128$	$N = 32$	$N = 64$	$N = 128$	$N = 32$	$N = 64$	$N = 128$	$N = 32$	$N = 64$	$N = 128$
LLaDA-MoE-7B-A1B												
ARC-C	25.85	26.79	27.56	29.61	28.50	27.82	31.40	30.46	31.31	31.40	30.46	31.74
ARC-E	37.04	37.54	37.50	36.83	34.85	34.89	40.15	39.39	38.59	44.61	42.00	43.94
BoolQ	60.52	57.58	62.17	41.07	41.71	46.88	45.87	45.72	44.13	60.49	48.23	53.79
COPA	57.00	57.00	65.00	58.00	62.00	65.00	63.00	56.00	62.00	69.00	62.00	70.00
OBQA	27.80	28.20	26.80	27.80	28.00	27.80	28.40	27.60	28.40	30.20	29.40	29.60
PIQA	59.14	60.01	59.68	59.25	58.54	58.38	59.63	58.32	58.81	63.38	61.15	63.33
RTE	56.68	60.29	59.57	56.32	59.57	58.48	57.04	56.32	54.87	61.73	53.07	58.84
WinoG	51.46	49.25	53.59	49.57	50.67	50.28	53.35	52.88	52.80	52.96	51.22	55.88
Average	46.94	47.08	48.98	44.80	45.48	46.19	47.36	45.84	46.37	**51.72**	**47.19**	**50.89**
SDAR-30B-A3B												
ARC-C	31.57	32.25	30.89	38.91	39.76	42.15	41.64	43.17	41.13	44.03	43.94	43.86
ARC-E	46.93	48.19	45.12	60.86	60.61	63.17	65.66	67.30	64.44	67.97	67.42	68.56
BoolQ	62.78	62.05	63.24	76.33	74.80	77.83	51.07	45.72	50.12	86.45	86.57	87.03
COPA	76.00	77.00	76.00	80.00	81.00	83.00	74.00	75.00	74.00	84.00	83.00	84.00
OBQA	31.20	32.40	31.00	36.60	37.00	37.00	34.00	32.80	32.60	39.20	39.40	38.40
PIQA	58.98	59.74	58.71	64.42	64.58	64.96	64.74	63.49	63.71	69.48	69.21	69.21
RTE	61.73	59.57	63.90	57.76	57.76	59.21	79.78	78.70	74.73	67.51	67.51	72.56
WinoG	53.67	53.35	52.88	63.93	64.01	64.25	60.69	58.25	58.80	66.77	68.75	68.35
Average	52.86	53.07	52.72	59.85	59.94	61.45	58.95	58.05	57.44	**65.68**	**65.73**	**66.50**

Sensitivity to M. We additionally evaluate the SDAR NAPE checkpoints with $M = 32$ on six tasks to assess sensitivity to the Monte Carlo evaluation budget. Table 7 shows that the 25%-pruned checkpoint achieves the higher six-task mean at both $M = 16$ (65.61 vs. 62.73) and $M = 32$ (66.33 vs. 62.41). It leads on all six tasks at $M = 16$ and five of six tasks at $M = 32$. Increasing the evaluation budget changes the six-task mean by only $+0.72$ points at 25% pruning and -0.32 points at 50% pruning. We therefore use $M = 16$ throughout, with the same Monte Carlo masks across pruning criteria for paired comparison.

E Calibration Robustness

Table 8 compares all scoring criteria using nested calibration sets of 32, 64, and 128 samples. NAPE achieves the highest eight-task average at every calibration-set size on both backbones. On LLaDA-MoE, its margins over the strongest competing criterion are 4.36, 0.11, and 1.91 points for $N = 32$, 64, and 128, respectively. On SDAR, the corresponding margins are 5.83, 5.79, and 5.05 points. NAPE therefore preserves its leading performance even when the calibration set is reduced from 128 to 32 samples.

Table 9: **Full masked-target score ablation at 50% expert pruning.** All methods restrict the calibration signal to masked prediction targets. Results are percentages. Small colored numbers denote the difference to NAPE ($\Delta = \text{NAPE} - \text{baseline}$); green indicates that NAPE performs better and red indicates the opposite.

Method	ARC-C	ARC-E	BoolQ	COPA	OBQA	PIQA	RTE	WinoG	Avg. Rank ↓
				LLaDA-MoE-7B-A1B					
Masked-REAP	31.31$_{+0.43}$	38.59$_{+5.35}$	44.13$_{+9.66}$	62.00$_{+8.00}$	28.40$_{+1.20}$	58.81$_{+4.52}$	54.87$_{+3.97}$	52.80$_{+3.08}$	2.375
Masked-EAN	27.82$_{+3.92}$	34.89$_{+9.05}$	46.88$_{+6.91}$	65.00$_{+5.00}$	27.80$_{+1.80}$	58.38$_{+4.95}$	58.48$_{+0.36}$	50.28$_{+5.60}$	2.625
NAPE (Ours)	**31.74**	**43.94**	**53.79**	**70.00**	**29.60**	**63.33**	**58.84**	**55.88**	**1.000**
				SDAR-30B-A3B					
Masked-REAP	41.13$_{+2.73}$	64.44$_{+4.12}$	50.12$_{+36.91}$	74.00$_{+10.00}$	32.60$_{+5.80}$	63.71$_{+5.50}$	**74.73**$_{-2.17}$	58.80$_{+9.55}$	2.625
Masked-EAN	42.15$_{+1.71}$	63.17$_{+5.39}$	77.83$_{+9.20}$	83.00$_{+1.00}$	37.00$_{+1.40}$	64.96$_{+4.25}$	59.21$_{+13.35}$	64.25$_{+4.10}$	2.250
NAPE (Ours)	**43.86**	**68.56**	**87.03**	**84.00**	**38.40**	**69.21**	72.56	**68.35**	**1.125**

Table 10: **Full fixed- versus multi-noise calibration ablation.** Task columns report accuracy (%) at 50% expert pruning. Small colored numbers denote the difference to NAPE ($\Delta = \text{NAPE} - \text{Fixed}$): green indicates an advantage for NAPE, red indicates an advantage for the corresponding fixed-noise calibration, and gray indicates a tie. Avg. Rank is computed across the six calibration settings within each task and then averaged over the eight tasks; lower is better.

Calibration	ARC-C	ARC-E	BoolQ	COPA	OBQA	PIQA	RTE	WinoG	Avg. Rank ↓
				LLaDA-MoE-7B-A1B					
Fixed $r = 0.1$	**32.17**$_{-0.43}$	44.61$_{-0.67}$	44.56$_{+9.23}$	70.00$_{0.00}$	28.20$_{+1.40}$	63.00$_{+0.33}$	59.93$_{-1.09}$	**57.06**$_{-1.18}$	2.63
Fixed $r = 0.3$	30.72$_{+1.02}$	44.11$_{-0.17}$	45.11$_{+8.68}$	**72.00**$_{-2.00}$	**29.80**$_{-0.20}$	**63.49**$_{-0.16}$	49.46$_{+9.38}$	55.80$_{+0.08}$	2.94
Fixed $r = 0.5$	29.69$_{+2.05}$	41.46$_{+2.48}$	**54.25**$_{-0.46}$	67.00$_{+3.00}$	29.60$_{0.00}$	61.21$_{+2.12}$	52.71$_{+6.13}$	54.85$_{+1.03}$	4.44
Fixed $r = 0.7$	31.23$_{+0.51}$	41.37$_{+2.57}$	44.13$_{+9.66}$	68.00$_{+2.00}$	29.20$_{+0.40}$	61.97$_{+1.36}$	51.99$_{+6.85}$	55.80$_{+0.08}$	4.56
Fixed $r = 0.9$	31.14$_{+0.60}$	**46.30**$_{-2.36}$	38.96$_{+14.83}$	70.00$_{0.00}$	27.80$_{+1.80}$	62.68$_{+0.65}$	**62.82**$_{-3.98}$	53.35$_{+2.53}$	3.88
NAPE (Ours)	31.74	43.94	53.79	70.00	29.60	63.33	58.84	55.88	**2.56**
				SDAR-30B-A3B					
Fixed $r = 0.1$	40.87$_{+2.99}$	40.40$_{+28.16}$	37.83$_{+49.20}$	57.00$_{+27.00}$	29.80$_{+8.60}$	49.51$_{+19.70}$	52.71$_{+19.85}$	65.43$_{+2.92}$	6.00
Fixed $r = 0.3$	41.13$_{+2.73}$	65.07$_{+3.49}$	85.99$_{+1.04}$	81.00$_{+3.00}$	36.60$_{+1.80}$	67.95$_{+1.26}$	69.31$_{+3.25}$	68.59$_{-0.24}$	4.13
Fixed $r = 0.5$	42.58$_{+1.28}$	68.35$_{+0.21}$	85.99$_{+1.04}$	**84.00**$_{0.00}$	37.20$_{+1.20}$	68.50$_{+0.71}$	**74.73**$_{-2.17}$	**68.67**$_{-0.32}$	2.75
Fixed $r = 0.7$	42.66$_{+1.20}$	68.56$_{0.00}$	84.56$_{+2.47}$	**84.00**$_{0.00}$	**38.40**$_{0.00}$	68.61$_{+0.60}$	67.51$_{+5.05}$	66.69$_{+1.66}$	3.00
Fixed $r = 0.9$	**44.20**$_{-0.34}$	**71.21**$_{-2.65}$	83.91$_{+3.12}$	81.00$_{+3.00}$	37.40$_{+1.00}$	68.50$_{+0.71}$	69.68$_{+2.88}$	66.54$_{+1.81}$	3.25
NAPE (Ours)	43.86	68.56	**87.03**	**84.00**	**38.40**	**69.21**	72.56	68.35	**1.88**

F Full Ablation Results

Masked-target scoring. Table 9 reports the complete per-task results for the masked-target score ablation. NAPE ranks first on all eight tasks on LLaDA-MoE and on seven of eight tasks on SDAR, yielding average ranks of 1.000 and 1.125, respectively. The consistent gains over Masked-EAN and Masked-REAP show that restricting the calibration signal to masked prediction targets alone does not account for NAPE's improvement; the gate-gradient criterion provides a stronger expert-importance signal.

Fixed- versus multi-noise calibration. Table 10 provides the complete results for the noise-calibration ablation. NAPE obtains the best average rank on both LLaDA-MoE (2.56) and SDAR (1.88), while also achieving a higher eight-task mean than every tested fixed-noise setting on both backbones. The strongest fixed ratio varies across tasks and models, whereas multi-noise calibration provides the strongest aggregate performance without selecting a task-specific noise level.