GRAFT-MoE: Group-guided Routing Adaptation and Functional Transfer for Pretrained MoE Expansion

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) expansion must add capacity without disrupting pretrained routing. We propose GRAFT-MoE, which groups pretrained experts using routed-input statistics and shared-probe responses, allocates new capacity by group traffic, selects representative parents, and applies a layerwise routed-output warm-start that jointly updates new experts and the router. During continued training, confidence-aware intra-cluster exploration allows low-confidence experts to be replaced by same-group alternatives while preserving k distinct selections; it is disabled at inference. On OLMoE-1B-7B-0125 and Qwen1.5-MoE-A2.7B at $1.25\times$, $1.5\times$, and $2\times$ expansion, GRAFT-MoE achieves the best average downstream score in all six settings and the lowest perplexity in five. At $1.5\times$, it reaches 8.86 PPL / 65.08 Avg on OLMoE and 10.97 PPL / 66.63 Avg on Qwen, and recovers the source NLL after 8.39M tokens (12.17 GPU-hours) versus 12.58M (16.29 GPU-hours) for EU-GN. It improves over EU-GN by 0.29/0.36 average-score points and over full-parameter continued pretraining by 0.79/0.77 points on OLMoE/Qwen at $1.5\times$ expansion, while reducing recovery cost by 25.3%.

We provide comprehensive analysis and release our code at `https://anonymous.4open.science/r/Graft_MoE`.

1 Introduction

Mixture-of-Experts (MoE) language models increase total parameter capacity by activating only a subset of experts for each token (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). This separation between total and active parameters enables capacity growth without a proportional increase in expert computation per token. The preferred sparsity configuration, however, depends on model size, training data, and computational constraints (Ludziejewski et al., 2024; Abnar et al., 2025). As these requirements evolve, expanding a useful pretrained checkpoint offers an alternative to training a larger expert pool from scratch.

We study *post-pretraining expert expansion*: increasing the number of routed experts in an existing MoE while retaining expert dimensions and the number activated per token. Prior work establishes checkpoint reuse through dense-to-MoE upcycling (Komatsuzaki et al., 2023), domain-aware expert integration (Gritsch et al., 2025), and modality and task extension (Zhong et al., 2024). For MoE-to-MoE growth, orthogonal model growth combines depth expansion with expert duplication (Wang et al., 2026), while Expert Upcycling studies replication and router extension, including gradient-guided non-uniform replication (Dwivedi et al., 2026). These advances motivate a complementary question: *how can relationships among pretrained experts guide where capacity is added and how it is integrated?*

Answering this question requires connecting expert behavior with routing. Usage frequency indicates how often an expert is selected, but not which transformation it performs. Comparing responses on different routed inputs also confounds functional differences with input-distribution differences. Moreover, parameter copying need not preserve the layer output: additional router entries change probability normalization and top-k competition even when copied transformations are identical. Newly added experts must then receive useful training opportunities without unre-

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.

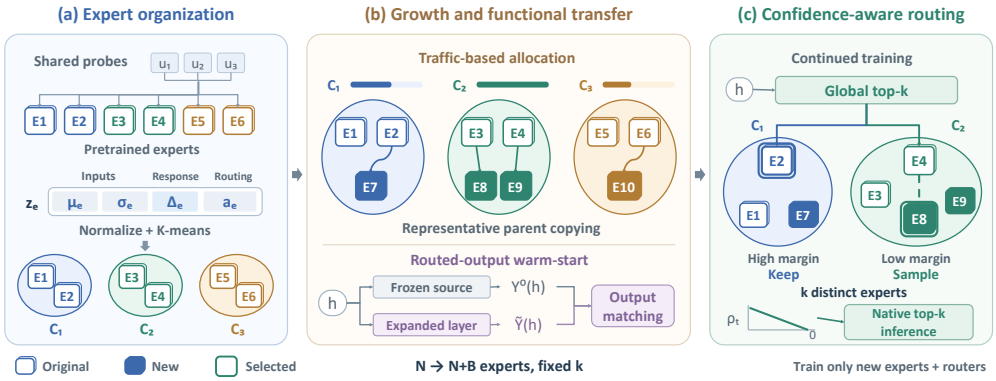


Figure 1: **Overview of GRAFT-MoE.** Shared-probe responses and routed-input statistics organize pretrained experts into groups, which guide capacity allocation and parent selection. Routed-output warm-start transfers source behavior to the expanded layers, followed by continued training with confidence-aware intra-cluster exploration. Inference uses native routing.

stricted changes to existing expert combinations. Expert expansion therefore involves three coupled decisions: where to allocate capacity, what behavior to transfer, and how to adapt routing.

We propose **GRAFT-MoE**, a framework for *Group-guided Routing Adaptation and Functional Transfer*. As illustrated in Figure 1, GRAFT-MoE groups pretrained experts using routed-input statistics, shared-probe responses, and routing preferences. These groups guide traffic-based capacity allocation and representative parent selection. A layerwise warm-start then jointly adjusts the added experts and router to match the source layer's actual routed output. During continued training, confidence-aware intra-cluster exploration allows related experts to exchange selected positions while preserving the number of distinct active experts. Exploration decays during training and is disabled for inference.

Our goal is to integrate additional capacity under a limited continued-training budget while retaining pretrained behavior. Keeping the activation count fixed does not remove storage, optimizer-state, or routing overhead; quality and recovery must therefore be evaluated alongside these costs. Our contributions are threefold:

- **Behavior-guided expert growth.** We combine shared-probe responses and routed-input statistics into comparable expert descriptors, using the resulting groups to guide capacity allocation and parent selection.
- **Routed-output functional transfer.** We introduce a layerwise warm-start that jointly adapts new experts and routers by matching the actual routed mixture, accounting for both expert transformations and routing weights.
- **Confidence-aware routing adaptation.** We develop temporary intra-cluster exploration that preserves k distinct active experts and per-token group selection counts, relative to the expanded router's initial selection, while retaining native routing for inference.

2 Related Work

Mixture-of-Experts Scaling. MoE originated as a framework for distributing inputs among specialized networks (Jacobs et al., 1991). Sparsely gated MoE extended this principle to large neural models (Shazeer et al., 2017), followed by advances in distributed scaling, routing simplicity, and training stability through GShard, Switch Transformer, and ST-MoE (Lepikhin et al., 2021; Fedus et al., 2022; Zoph et al., 2022). Scaling-law studies further characterize the effects of expert granularity and sparsity under different budgets (Ludziejewski et al., 2024; Abnar et al., 2025). Our work concerns how to reach a larger expert configuration from an existing MoE checkpoint while integrating the added capacity.

Upcycling and Expert Expansion. Sparse Upcycling initializes MoEs from dense checkpoints (Komatsuzaki et al., 2023). DeRS improves parameter efficiency through shared base weights and lightweight expert-specific deltas (Huang et al., 2025). Cluster-aware Upcycling clusters dense-model input activations to construct specialized experts during CLIP upcycling (Chu et al., 2026). Nexus uses domain-informed routing to integrate additional experts (Gritsch et al., 2025), and MoExtend adds experts for modality and task extension (Zhong et al., 2024). For pretrained MoE growth, orthogonal growth combines layer copying and expert duplication (Wang et al., 2026), while Expert Upcycling uses gradient-based utility to guide replication (Dwivedi et al., 2026). GRAFT-MoE groups experts already present in a pretrained MoE using their input statistics and common-probe responses, then connects these groups to allocation, output transfer, and routing adaptation.

Expert Organization and Routing. Recent studies explicitly examine the interaction between expert capabilities and router decisions. Expert–Router Coupling introduces an auxiliary alignment loss (Lv et al., 2026), while cross-layer analysis traces how preceding components influence subsequent routing (Li et al., 2026). DirMoE separates expert selection from mixture weighting through probabilistic routing (Vahidi et al., 2026), and LD-MoLE learns token-dependent activation budgets for LoRA experts (Zhuang et al., 2026). These studies motivate treating expert behavior and routing jointly. GRAFT-MoE addresses their interaction specifically during expert-count expansion: training-time refinement operates within discovered groups, retains the native activation count, and is disabled during inference.

Function Transfer during Model Growth. Net2Net enlarges networks through function-preserving transformations (Chen et al., 2016), and bert2BERT transfers pretrained parameters to larger Transformers (Chen et al., 2022). Knowledge distillation transfers model behavior through teacher supervision (Hinton et al., 2015). In sparse MoEs, expert replication additionally changes routing competition and mixture weights. GRAFT-MoE therefore matches the complete routed layer output on shared inputs, jointly adapting new experts and the router. This provides an empirical warm-start for further training, rather than a guarantee of exact function preservation.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Consider a pretrained MoE layer with N routed experts. For an input hidden state $h \in \mathbb{R}^d$, its output is

$$Y^0(h) = Y^0_{\mathrm{sh}}(h) + \sum_{e \in \mathcal{S}^0_k(h)} g^0_e(h) E^0_e(h), \tag{1}$$

where E^0_e is expert e, $\mathcal{S}^0_k(h)$ is the source top-k set, and $g^0_e(h)$ follows the model's native mixture-weight convention, including any renormalization over selected experts. Y^0_{sh} denotes the shared-expert branch, including its gate when present, and is zero otherwise. Layer indices are omitted unless needed.

Given a per-layer budget $B > 0$, we expand the routed expert count to $N + B$ while keeping expert dimensions and k unchanged. Shared experts are excluded from the budget. Original experts, shared branches, and the remaining backbone stay frozen; only added experts and routed-expert routers are optimized during warm-start and continued training.

3.2 COMPARABLE EXPERT DESCRIPTORS AND GROUPING

We characterize both the inputs assigned to each expert and its responses to common inputs. Calibration examples are drawn from the training corpus and separated into fitting and checking subsets; only the fitting subset is used to construct groups and initialize the expanded model.

Routed-input statistics. Using the frozen source model, let $\mathcal{H}_e$ contain the fitting hidden states assigned to expert e, with $M_e = |\mathcal{H}_e| > 0$. We compute

$$\mu_e = \frac{1}{M_e} \sum_{h \in \mathcal{H}_e} h, \quad \sigma_e = \left[\frac{1}{M_e} \sum_{h \in \mathcal{H}_e} (h - \mu_e) \odot (h - \mu_e) \right]^{1/2}, \quad a_e = \frac{1}{M_e} \sum_{h \in \mathcal{H}_e} p^0(e \mid h). \tag{2}$$

Here, σ_e is the coordinate-wise standard deviation, and a_e is the mean global routing probability conditional on selection, rather than an activation frequency.

Common-probe responses. To compare transformations on a common basis, all experts within a layer process the same fitting probes $\{u_q\}_{q=1}^{Q}$. For column-vector outputs and a fixed shared projection $P \in \mathbb{R}^{d\times r}$, we define

$$\Delta_e = Q^{-1/2}\,\mathrm{vec}\left([P^\top E_e^0(u_1), \ldots, P^\top E_e^0(u_Q)]\right). \tag{3}$$

This Qr-dimensional descriptor retains responses to individual probes and represents projected expert outputs, rather than complete Transformer residual updates.

Expert grouping. We normalize each feature block separately and concatenate them:

$$z_e = [\mathcal{N}(\mu_e); \mathcal{N}(\sigma_e); \mathcal{N}(\Delta_e); \mathcal{N}(a_e)] \in \mathbb{R}^{2d+Qr+1}. \tag{4}$$

$\mathcal{N}$ centers each block across experts and scales it by its root mean squared centered norm; negligible-energy blocks are set to zero. Standard Euclidean K-means partitions the descriptors into K groups $\{\mathcal{C}_c\}_{c=1}^{K}$. These groups capture empirical similarity under the calibration distribution, rather than assumed semantic categories. Normalization and calibration details appear in Appendix A.1.

3.3 Group-Guided Growth and Functional Transfer

Capacity allocation and initialization. Let n_e count fitting-token assignments to expert e. Each group's traffic share is

$$A_c = \frac{\sum_{e\in\mathcal{C}_c} n_e}{\sum_{e=1}^{N} n_e}. \tag{5}$$

Largest-remainder rounding of BA_c gives integer budgets B_c with $\sum_c B_c = B$. Traffic counts expert assignments rather than unique tokens and serves as a capacity-allocation heuristic. Within each group, we select the expert nearest its centroid, followed by farthest-first selection to cover additional descriptor locations. Each new expert inherits its parent's parameters, router row, and group label. Appendix A.2 specifies the rounding and parent-selection rules, together with optional initialization perturbations.

Routed-output warm-start. Parameter copying need not preserve the layer output because expansion changes routing normalization and top-k competition. We therefore match the actual routed mixture, as illustrated in Figure 2(a), rather than fitting individual experts to group prototypes. At layer ℓ, we minimize

$$\mathcal{L}_{\mathrm{warm}}^{(\ell)} = \frac{\mathbb{E}_{h\sim\mathcal{D}_{\mathrm{fit}}^{(\ell)}} \|\widetilde{Y}^{(\ell)}(h) - Y^{0,(\ell)}(h)\|_2^2}{\max\{\mathbb{E}_{h\sim\mathcal{D}_{\mathrm{fit}}^{(\ell)}} \|Y^{0,(\ell)}(h)\|_2^2, \epsilon\}}. \tag{6}$$

Here, $\mathcal{D}_{\mathrm{fit}}^{(\ell)}$ is the empirical distribution of source hidden states and $\epsilon > 0$ stabilizes the denominator. The frozen source and expanded layer receive identical inputs; both outputs include the unchanged shared branch when present. Warm-start proceeds layer by layer, jointly updating new experts and the corresponding router under native top-k routing. Checking inputs are reserved for measuring output recovery.

Lemma 1 separates output perturbation into changes in expert transformations and in routing weights aggregated by source parent, motivating this joint matching objective. The analysis concerns matched layer inputs; end-to-end recovery must still be assessed on the expanded language model.

3.4 Confidence-Aware Intra-Cluster Routing Adaptation

After warm-start, we allow temporary exploration when a selected expert lacks a sufficiently large within-group routing margin (Figure 2(b)). Let $p_e(h)$ be the expanded router's global softmax probability and $\mathcal{C}^+(e)$ its group, including added members. Define $q_j(h \mid \mathcal{C}) = p_j(h)/\sum_{u\in\mathcal{C}} p_u(h)$. For a selected expert e in a non-singleton group, the margin is

$$m_e(h) = q_e(h \mid \mathcal{C}^+(e)) - \max_{j\in\mathcal{C}^+(e)\setminus\{e\}} q_j(h \mid \mathcal{C}^+(e)). \tag{7}$$

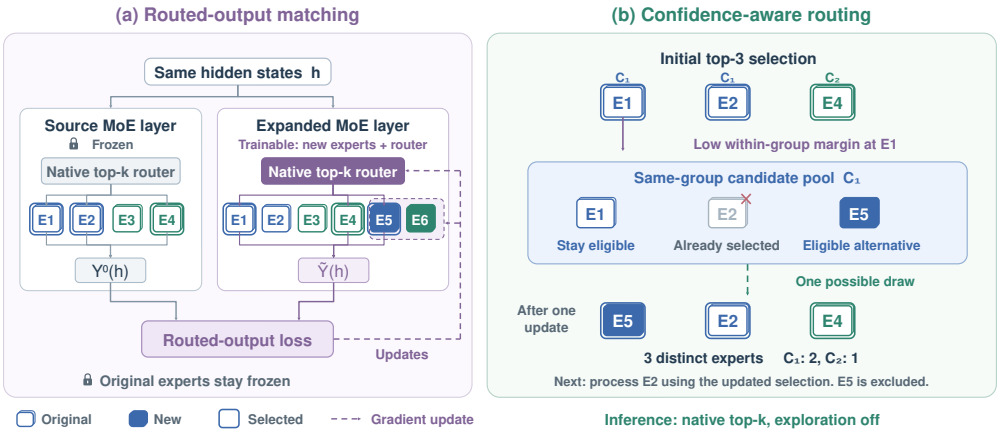


Figure 2: **Functional transfer and routing adaptation.** The frozen source and expanded MoE layer receive the same hidden states. Routed-output matching updates new experts and the expanded router while retaining the original experts. During continued training, a selected expert with a small within-group margin may be replaced by an eligible expert from the same group. Sequential replacement excludes other selected experts, preserving k distinct selections and their group counts. Exploration is disabled for inference.

The margin compares e with its strongest group competitor. It can be negative and is used as an exploration criterion, rather than a calibrated probability of correctness.

Sequential local replacement. Starting from the expanded router's global top-k set, we process selected positions sequentially in a fixed order. Let e_i occupy the current position and $\mathcal{S}_i$ denote the current selected set. The eligible set is $\mathcal{A}_i = \mathcal{C}^+(e_i) \setminus (\mathcal{S}_i \setminus \{e_i\})$, excluding other selected experts while retaining e_i. We keep positions with $m_{e_i}(h) \geq \tau$ and singleton groups. Otherwise, we sample

$$\Pr(\widetilde{e}_i = j \mid h, \mathcal{S}_i) = (1 - \rho_t)\mathbb{I}[j = e_i] + \rho_t \frac{\mathbb{I}[j \in \mathcal{A}_i] p_j(h)}{\sum_{u \in \mathcal{A}_i} p_u(h)}. \tag{8}$$

The selected set is updated after each replacement. This rule preserves k distinct experts and each group's selection count relative to the initial expanded top-k set (Appendix A.4). It permits, but does not guarantee, additional traffic to new experts.

Training and inference. Conditional probabilities govern sampling only. The final mixture uses the selected experts' global probabilities with the native renormalization convention. Continued training optimizes $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{LB}}\mathcal{L}_{\text{LB}}$ over new experts and routed-expert routers. Load balancing uses post-refinement assignments and is averaged across layers; the warm-start objective is not retained. We linearly decay ρ_t to zero and use $\tau = 0.2$ by default. Validation, testing, and inference disable replacement and use deterministic native top-k routing.

4 Experiments

4.1 Experimental Setup

Models. We consider OLMoE-1B-7B-0125 (Muennighoff et al., 2025) and Qwen1.5-MoE-A2.7B (Qwen Team, 2024) at $1.25\times$, $1.5\times$, and $2\times$ routed-expert expansion. OLMoE expands from 64 to $\{80, 96, 128\}$ routed experts per layer, and Qwen from 60 to $\{75, 90, 120\}$. We retain expert dimensions and native activation budgets (top-8 for OLMoE and top-4 for Qwen). Qwen additionally has 4 shared experts per layer, which remain unchanged and are excluded from expansion.

Data and evaluation. Continued pretraining uses 33.55M input tokens from DCLM (Li et al., 2024), with 4.19M tokens each for validation and testing. A 1.05M-token calibration subset from

the training data is partitioned into fitting and checking subsets. We report test perplexity and scores on HellaSwag (Zellers et al., 2019), ARC-Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), MMLU (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021). Evaluation is zero-shot except for 5-shot MMLU and 8-shot GSM8K. Avg. is the unweighted mean of the six downstream scores; perplexity is compared within each backbone.

Baselines. The unexpanded controls are Original and full-parameter continued pretraining (Continue Original). Expansion baselines include Random Copy, which samples parents uniformly without replacement; Traffic-Prop., which allocates copies in proportion to calibration traffic; and EU-GN, our gradient-norm-based adaptation of Expert Upcycling (Dwivedi et al., 2026). At $2\times$ expansion, Random Copy becomes Uniform Copy, duplicating every source expert once. Expanded methods share the target expert count and LM training budget.

Training and reporting. Training uses NVIDIA A800 80GB GPUs, sequences of 2,048 tokens, and 4,096 optimizer steps with an effective batch of 8,192 tokens and a peak learning rate of 10^{-4}. Expanded models update only new experts and routers; Continue Original updates all parameters. GRAFT-MoE uses eight clusters and 64 warm-start steps per layer. Preparation and warm-start are additional to the shared LM budget. We specify three training seeds (42, 43, and 44) per configuration and aggregate their scores using arithmetic means and sample standard deviations. Further evaluation details and hyperparameters are deferred to Appendix B.1 and Appendix B.2.

4.2 Overall Performance

Table 1 compares GRAFT-MoE with copy-based expansion baselines, EU-GN, and full-parameter continued pretraining across two backbones and three expansion ratios. Expansion methods are compared at matched target expert counts and LM training budgets, with preparation and warm-start costs considered separately.

Comparison with expansion baselines. In the experimental comparison, GRAFT-MoE has the highest average downstream score in all six backbone–scale configurations and the lowest perplexity in five. At $1.5\times$ expansion, its perplexity is 8.86 on OLMoE and 10.97 on Qwen, compared with 8.99 and 11.12 for EU-GN. The corresponding average downstream scores are 65.08 and 66.63, exceeding EU-GN by 0.29 and 0.36 percentage points. These aggregate differences do not imply superiority on every task: EU-GN retains a higher PIQA score on OLMoE at this scale.

Comparison with continued pretraining. At $1.5\times$ expansion, GRAFT-MoE's average downstream scores exceed those of Continue Original by 0.79 percentage points on OLMoE and 0.77 on Qwen, with perplexity reductions of 0.19 and 0.30, respectively. These comparisons use a shared LM token budget, but differ in trainable parameter sets and preparation costs. Computational efficiency is evaluated separately in Section 4.3.

Effect of expansion scale. Increasing GRAFT-MoE from $1.5\times$ to $2\times$ expansion changes OLMoE perplexity from 8.86 to 8.92, while its average score changes only from 65.08 to 65.10. For Qwen, perplexity increases from 10.97 to 11.02 and the average score decreases from 66.63 to 66.57. At $2\times$ expansion on OLMoE, EU-GN has slightly lower perplexity than GRAFT-MoE (8.90 versus 8.92). In this illustrative comparison, enlarging the expert pool therefore does not uniformly improve quality under the fixed LM training budget.

4.3 Recovery Dynamics and Efficiency

We examine recovery following $1.5\times$ expansion of OLMoE, from 64 to 96 routed experts per layer. Figure 3 shows validation NLL against continued-pretraining tokens and cumulative GPU-hours for seed 42. All checkpoints are evaluated on the same validation subset. We define recovery as the first evaluated checkpoint whose NLL reaches or falls below the source-model reference of 2.26881. The computational cost includes method-specific preparation and continued pretraining, excluding evaluation and checkpoint I/O.

Although GRAFT-MoE starts with a larger NLL increase than the other expansion methods, it recovers more rapidly. It first reaches the source-model reference after 8.39M tokens, compared with

Experts	Method	PPL↓	Hella.	ARC-C	PIQA	Wino.	MMLU	GSM8K	Avg.↑
OLMoE-1B-7B-0125 (top-8)									
64	Continue Original (full)	9.05	78.65	49.66	78.82	69.14	55.48	53.98	64.29
80	Random Copy	9.35	77.96	49.40	78.23	68.71	54.96	52.92	63.70
	Traffic-Prop.	9.16	78.47	49.83	78.64	69.22	55.37	53.61	64.19
	EU-GN†	9.04	78.73	50.12	78.82	69.38	55.64	**54.13**	64.47
	GRAFT-MoE	**8.98**	**78.91**	**50.26**	**79.01**	**69.61**	**55.72**	54.08	**64.60**
96	Random Copy	9.41	77.61	49.32	78.45	68.34	54.77	52.65	63.52
	Traffic-Prop.	9.12	78.56	50.03	78.91	69.30	55.49	54.10	64.40
	EU-GN†	8.99	78.95	50.41	**79.14**	69.69	55.91	54.62	64.79
	GRAFT-MoE	**8.86**	**79.24**	**50.87**	79.06	**70.01**	**56.18**	**55.11**	**65.08**
128	Uniform Copy	9.54	77.44	49.15	78.30	68.17	54.63	52.42	63.35
	Traffic-Prop.	9.15	78.70	50.12	79.09	69.54	55.66	54.42	64.59
	EU-GN†	**8.90**	79.18	50.61	**79.26**	69.93	56.05	**55.07**	65.02
	GRAFT-MoE	8.92	**79.32**	**50.75**	79.20	**70.10**	**56.30**	54.91	**65.10**
Qwen1.5-MoE-A2.7B (top-4)									
60	Continue Original (full)	11.27	77.51	44.63	80.02	69.38	61.66	61.95	65.86
75	Random Copy	11.72	76.92	43.98	79.40	68.79	60.87	60.83	65.13
	Traffic-Prop.	11.43	77.23	44.42	79.71	69.14	61.19	61.52	65.54
	EU-GN†	11.30	77.48	44.80	79.98	69.53	61.48	**62.06**	65.89
	GRAFT-MoE	**11.19**	**77.74**	**45.16**	**80.13**	**69.65**	**61.87**	61.98	**66.09**
90	Random Copy	11.85	76.64	43.72	79.15	68.60	60.68	60.45	64.87
	Traffic-Prop.	11.34	77.45	44.78	79.93	69.38	61.43	61.87	65.81
	EU-GN†	11.12	77.88	45.23	80.16	69.91	61.99	62.44	66.27
	GRAFT-MoE	**10.97**	**78.20**	**45.71**	**80.34**	**70.13**	**62.37**	**63.02**	**66.63**
120	Uniform Copy	12.02	76.31	43.50	78.94	68.35	60.42	60.11	64.61
	Traffic-Prop.	11.39	77.36	44.67	79.98	69.54	61.64	61.73	65.82
	EU-GN†	11.04	78.04	45.53	**80.42**	70.01	62.21	**62.89**	66.52
	GRAFT-MoE	**11.02**	**78.12**	**45.64**	80.30	**70.24**	**62.35**	62.76	**66.57**

Table 1: **Comparison across backbones and expansion scales.** Experts denotes routed experts per layer; Qwen's shared experts remain unchanged. Avg. averages the six downstream scores (%). Continue Original updates all parameters, whereas expansion methods update new experts and routers. Traffic-Prop. allocates copies in proportion to calibration traffic; at 2×, Uniform Copy duplicates each source expert once. †EU-GN denotes an adapted gradient-norm baseline.

12.58M tokens for EU-GN, requiring 33.3% fewer tokens at the evaluated checkpoints. Random Copy and Traffic Copy remain above the reference at every evaluated checkpoint within the 33.55M-token budget. At the end of training, GRAFT-MoE achieves a validation NLL of 2.18543, compared with 2.23654 for EU-GN, 2.27584 for Random Copy, and 2.27865 for Traffic Copy.

The advantage in recovery persists when preparation costs are included. GRAFT-MoE first reaches the reference at 12.17 GPU-hours, compared with 16.29 GPU-hours for EU-GN, reducing the observed recovery cost by 25.3%. For this run, faster recovery therefore compensates for the additional preparation cost relative to EU-GN. These recovery budgets correspond to evaluated checkpoints, rather than interpolated crossings of the reference. More experimental results are provided in Appendix B.

4.4 Routing Behavior and New-Expert Utilization

We investigate whether training-time exploration leads to sustained use of newly added experts under native routing. For OLMoE expanded from 64 to 96 experts per layer, new-expert utilization denotes the fraction of token–expert assignments received by the 32 added experts. All checkpoints

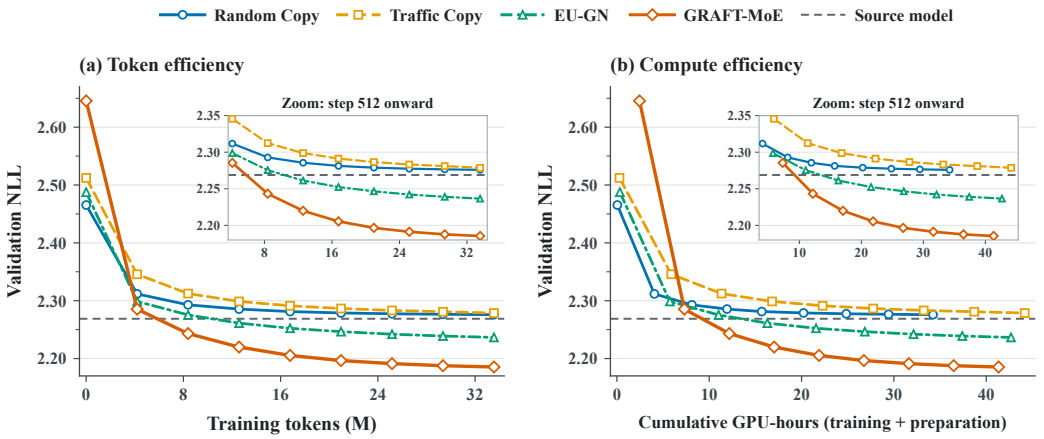


Figure 3: Recovery dynamics for OLMoE under $1.5\times$ routed-expert expansion (64 to 96 experts per layer), using seed 42. Validation NLL is plotted against (a) continued-pretraining tokens and (b) cumulative GPU-hours, including preparation and continued training. The dashed horizontal line indicates the source-model NLL. Markers denote evaluated checkpoints; insets show enlarged views from step 512 onward.

are evaluated on the same validation subset using native top-8 routing with exploration disabled. All methods share the same load-balancing objective and coefficient.

Figure 4(a) tracks utilization averaged over the 16 layers. Although all methods begin near 30%, their utilization trajectories diverge during continued training. After 33.55M input tokens, GRAFT-MoE retains 22.80% of assignments for the added experts, compared with 10.20% for EU-GN, 5.50% for Traffic Copy, and 2.16% for Random Copy. The no-exploration variant retains the same grouping and warm-start but finishes at only 2.10%, a gap of 20.70 percentage points. This comparison supports the role of confidence-aware exploration in sustaining new-expert traffic under the tested load-balancing configuration. Because exploration is disabled at evaluation, this gap reflects differences in learned native-routing behavior.

Figure 4(b) resolves this difference across network depth. At the final checkpoint, GRAFT-MoE achieves 19.46–26.79% utilization across all 16 layers, exceeding the corresponding baselines in every layer. The no-exploration variant remains within 1.23–2.92%, while Random Copy falls below 1% in seven layers. The higher aggregate utilization of GRAFT-MoE thus reflects greater participation of the added experts throughout the network. We further examine sensitivity to the confidence threshold τ through an ablation study in Appendix B.

4.5 Component Ablations

We ablate functional grouping, routed-output warm-start, and confidence-aware intra-cluster exploration at $1.5\times$ expansion on OLMoE and Qwen. Within each backbone, all variants share the expanded expert count, native activation budget, 33.55M continued-training input tokens, trainable parameter scope, and load-balancing configuration. Preparation, including warm-start, is additional to the continued-training budget. All evaluations use native routing with exploration disabled.

Table 2 shows that all three ablations increase mean perplexity and reduce mean downstream performance on both backbones. The largest degradation occurs when functional groups are replaced with size-matched random partitions: PPL increases by 0.24 on OLMoE and 0.30 on Qwen, while downstream averages decrease by 0.52 and 0.60 percentage points, respectively. The remaining construction and training procedures are retained, but group assignments determine capacity allocation, parent selection, and the candidate sets for exploration. This ablation therefore evaluates functional organization across the grouping-guided pipeline as a whole.

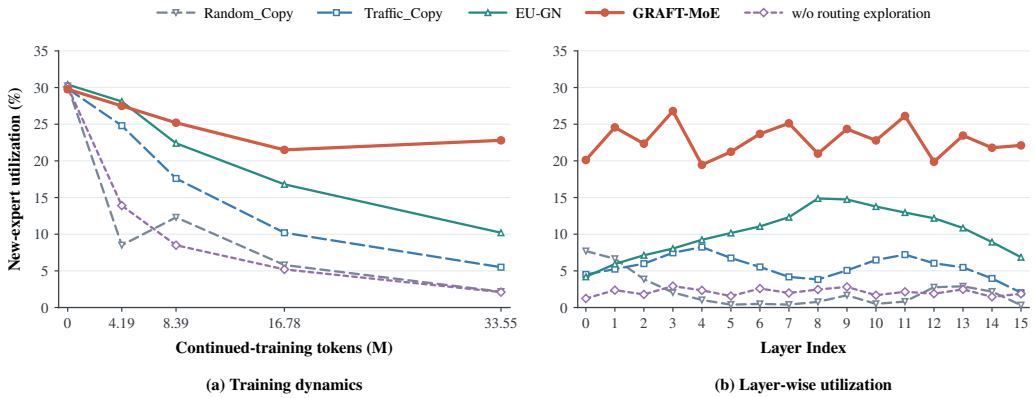


Figure 4: **New-expert utilization under native routing.** OLMoE is expanded from 64 to 96 experts per layer while retaining top-8 activation. Utilization measures the percentage of token–expert assignments received by the 32 added experts. (a) Utilization averaged over all 16 layers during continued training. (b) Layerwise utilization after 4096 training steps (33.55M input tokens). All measurements use the same validation subset with exploration disabled.

	OLMoE: 64 → 96		Qwen: 60 → 90	
Variant	PPL↓	Avg.↑	PPL↓	Avg.↑
GRAFT-MoE	$\mathbf{8.86 \pm 0.04}$	$\mathbf{65.08 \pm 0.21}$	$\mathbf{10.97 \pm 0.05}$	$\mathbf{66.63 \pm 0.24}$
Random groups	9.10 ± 0.07	64.56 ± 0.32	11.27 ± 0.08	66.03 ± 0.35
w/o warm-start	9.04 ± 0.06	64.69 ± 0.28	11.19 ± 0.07	66.19 ± 0.31
w/o routing exploration	9.01 ± 0.05	64.74 ± 0.25	11.15 ± 0.06	66.25 ± 0.28

Table 2: **Component ablations at $1.5\times$ expansion.** Results are means $\pm$ sample standard deviations over three training seeds. PPL is measured on the held-out test set; Avg. is the unweighted mean of the six downstream task scores, expressed as percentages. All variants retain top-8 activation for OLMoE and top-4 for Qwen. Bold denotes the best mean in each column.

Removing routed-output warm-start while retaining the expansion plan, copied initialization, and exploration raises mean PPL from 8.86 to 9.04 on OLMoE and from 10.97 to 11.19 on Qwen. Downstream averages also decline on both backbones. These results show that the benefit of the warm-start stage remains evident in the final model after the same amount of continued training.

Disabling exploration while retaining functional grouping and warm-start likewise worsens both metrics. The downstream average decreases by 0.34 percentage points on OLMoE and 0.38 on Qwen, accompanied by PPL increases of 0.15 and 0.18. These results complement the OLMoE utilization analysis in Section 4.4 by showing that training-time exploration also improves the reported model-quality metrics under the evaluated configuration.

5 Conclusion

We introduced GRAFT-MoE for group-guided MoE expansion. It combines behavior-based expert grouping, traffic-guided capacity allocation and parent selection, routed-output warm-start, and confidence-aware intra-cluster exploration. Experiments on OLMoE and Qwen show consistent gains over copy-based and gradient-norm-based expansion at matched expert counts and training budgets. GRAFT-MoE also recovers source performance faster than EU-GN. Ablations confirm the value of grouping, warm-start, and exploration. Limitations remain: fixed activation does not remove storage or routing overhead, and larger expert pools do not uniformly improve quality under a fixed budget.

AI USE STATEMENT

The authors used generative AI tools, including *[ChatGPT / Claude / GitHub Copilot*, during the preparation of this manuscript. Their use was limited to *[language polishing and grammar correction / code completion and debugging / LaTeX formatting / literature search assistance / figure preparation]*. All AI-generated outputs were reviewed, verified, and edited by the authors. The AI tools did not generate experimental results, fabricate data, or create references. The authors take full responsibility for the content, correctness, and integrity of this work.

REFERENCES

Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin El-Nouby, Joshua M. Susskind, and Vimal Thilak. Parameters vs FLOPs: Scaling laws for optimal sparsity for mixture-of-experts language models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/abnar25a.html.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2BERT: Towards reusable pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022. URL https://aclanthology.org/2022.acl-long.151/.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2Net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05641.

Sanghyeok Chu, Pyunghwan Ahn, Gwangmo Song, Seung Hwan Kim, Honglak Lee, and Bohyung Han. Enhancing mixture-of-experts specialization via cluster-aware upcycling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chu_Enhancing_Mixture-of-Experts_Specialization_via_Cluster-Aware_Upcycling_CVPR_2026_paper.html.

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. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Chaitanya Dwivedi, Binxuan Huang, Himanshu Gupta, Pratik Jayarao, Neeraj Varshney, and Bing Yin. Expert upcycling: Shifting the compute-efficient frontier of mixture-of-experts. *arXiv preprint arXiv:2604.19835*, 2026. URL https://arxiv.org/abs/2604.19835.

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. URL https://jmlr.org/papers/v23/21-0998.html.

Nikolas Gritsch, Qizhen Zhang, Acyr Locatelli, Sara Hooker, and Ahmet Üstün. Nexus: Adaptive upcycling to efficiently pretrain mixture of experts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24364–24381, 2025. doi: 10.18653/v1/2025.findings-emnlp.1323. URL https://aclanthology.org/2025.findings-emnlp.1323/.

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. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Yongqi Huang, Peng Ye, Chenyu Huang, Jianjian Cao, Lin Zhang, Baopu Li, Gang Yu, and Tao Chen. DeRS: Towards extremely efficient upcycled mixture-of-experts models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_DeRS_Towards_Extremely_Efficient_Upcycled_Mixture-of-Experts_Models_CVPR_2025_paper.html.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=T5nUQDrM4u.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.16668.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets_and_Benchmarks_Track.html.

Wengang Li, Lingqi Zhang, Toshio Endo, and Mohamed Wahib. Understanding cross-layer contributions to mixture-of-experts routing in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3efdd3cd3d70da09ec6ed3b3d17d2fd5-Abstract-Conference.html.

Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling laws for fine-grained mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 33270–33288, 2024. URL https://proceedings.mlr.press/v235/ludziejewski24a.html.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. Coupling experts and routers in mixture-of-experts via an auxiliary loss. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/79ff18412c1a5816f071a796375abc3d-Abstract-Conference.html.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah Smith, Pang Wei Koh, Amanpreet Singh, and Hanna Hajishirzi. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9b224ace8963c9385ad5e2b5c9039b97-Abstract-Conference.html.

Qwen Team. Qwen1.5-MoE: Matching 7B model performance with 1/3 activated parameters. Qwen official blog, 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

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. URL https://arxiv.org/abs/1701.06538.

Amirhossein Vahidi, Hesam Asadollahzadeh, Navid Akhavan Attar, Marie Moullet, Kevin Ly, Xingyi Yang, and Mohammad Lotfollahi. DirMoE: Dirichlet-routed mixture of experts. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fc0e3f908a2116ba529ad0a1530a3675-Abstract-Conference.html.

Ruizhe Wang, Yucheng Ding, Xiao Liu, Yaoxiang Wang, Peng Cheng, Baining Guo, Zhengjun Zha, and Yeyun Gong. Beyond sunk costs: Boosting LLM pre-training efficiency via orthogonal growth of mixture-of-experts. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.08008.

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. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Shanshan Zhong, Shanghua Gao, Zhongzhan Huang, Wushao Wen, Marinka Zitnik, and Pan Zhou. MoExtend: Tuning new experts for modality and task extension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 494–505, 2024. URL https://aclanthology.org/2024.acl-srw.53/.

Yuan Zhuang, Yi Shen, Yuexin Bian, Qing Su, Shihao Ji, Yuanyuan Shi, and Fei Miao. LD-MoLE: Learnable dynamic routing for mixture of LoRA experts. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6a39cf3b666f8bdb2223f253981f3869-Abstract-Conference.html.

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

A Additional Method Details and Proofs

A.1 Descriptor Construction and Normalization

All statistics are constructed independently within each layer using the frozen source model. Calibration examples come from the training corpus and are separated into fitting and checking subsets before hidden-state collection. Only fitting examples contribute to routed-input statistics, common

probes, group construction, traffic allocation, parent selection, and warm-start optimization. Checking examples are reserved for diagnostics. Within a layer, all experts use identical probe inputs and the same fixed projection, so response coordinates are comparable. The input-statistic definitions require nonzero fitting coverage for every expert; an uncovered expert requires additional fitting data rather than an undefined mean or variance.

For any feature block b_e, define

$$\bar{b} = \frac{1}{N}\sum_{e=1}^{N} b_e, \qquad s_b^2 = \frac{1}{N}\sum_{e=1}^{N} \|b_e - \bar{b}\|_2^2, \qquad \mathcal{N}(b_e) = \begin{cases} (b_e - \bar{b})/s_b, & s_b > \delta, \\ 0, & s_b \le \delta, \end{cases} \tag{9}$$

where $\delta > 0$ is a numerical tolerance. Each nondegenerate normalized block has unit average squared norm across experts. This balances aggregate block scale without whitening coordinates or guaranteeing equal clustering influence.

Writing m_c for a descriptor centroid, the grouping objective is

$$\min_{\{\mathcal{C}_c, m_c\}_{c=1}^{K}} \sum_{c=1}^{K} \sum_{e \in \mathcal{C}_c} \|z_e - m_c\|_2^2. \tag{10}$$

Groups partition the source experts; $1 \le K \le N$ and the allocation stage requires nonempty groups. The descriptor provides a finite, projected behavioral comparison, not an identification of the full expert function. Its reliability depends on calibration coverage, probe size, and projection dimension.

A.2 Integer Allocation and Parent Initialization

Using the traffic shares in Equation 5, let $v_c = BA_c$ and $R = B - \sum_c \lfloor v_c \rfloor$. Let $\mathcal{R}$ contain the R groups with the largest fractional remainders, resolving ties by group index. The allocation is

$$B_c = \lfloor v_c \rfloor + \mathbb{I}[c \in \mathcal{R}]. \tag{11}$$

Since traffic shares are nonnegative and sum to one, these budgets are nonnegative integers summing exactly to B.

For each group receiving capacity, first select the expert nearest its centroid. If $\mathcal{P}_c$ contains the selected parents, each subsequent distinct parent is chosen by

$$e^\star = \arg \max_{e \in \mathcal{C}_c \setminus \mathcal{P}_c} \min_{p \in \mathcal{P}_c} \|z_e - z_p\|_2^2. \tag{12}$$

Ties are resolved by expert index. If the budget exceeds the group size, cycle through the completed parent ordering. Coverage here refers to locations in descriptor space, not a guarantee of semantic diversity or downstream utility.

For completeness, a copied expert weight matrix or router row W_{par} admits the optional perturbation

$$W_{\text{new}} = W_{\text{par}} + \eta \frac{\|W_{\text{par}}\|_F}{\sqrt{\text{numel}(W_{\text{par}})}} Z, \qquad Z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1). \tag{13}$$

Exact-copy initialization corresponds to $\eta = 0$. For $\eta > 0$, perturbations are sampled independently for added weight blocks; retained parameters are not perturbed. The chosen scale is part of the experimental configuration, distinguishing exact-copy and perturbed-initialization variants. Small parameter perturbations alone do not bound function changes without further assumptions on the experts and their inputs.

A.3 Expansion-Induced Output Perturbation

Consider a fixed layer input h, suppressing this argument below. Let g_e^0 and $\widetilde{g}_j$ be nonnegative source and expanded mixture weights, including selection masks and the model's renormalization convention. Unselected experts have zero weight; we do not require the weights to sum to one. An unchanged shared branch, including its gate, cancels at matched inputs.

Let $\pi(j)$ map every expanded expert to its source parent, with $\pi(e) = e$ for retained experts. Define the parent-aggregated weight

$$\bar{g}_e = \sum_{j:\pi(j)=e} \widetilde{g}_j. \tag{14}$$

Let $g^0 = (g_1^0, \ldots, g_N^0)$ and $\bar{g} = (\bar{g}_1, \ldots, \bar{g}_N)$.

Lemma 1 (Expansion-induced output perturbation)**.** *With $M = \max_{1 \le e \le N} \|E_e^0(h)\|_2$, source and expanded layer outputs satisfy*

$$\|\widetilde{Y} - Y^0\|_2 \le \sum_{j=1}^{N+B} \widetilde{g}_j \|\widetilde{E}_j - E^0_{\pi(j)}\|_2 + M\|\bar{g} - g^0\|_1. \tag{15}$$

Proof. Add and subtract the mixture of source-parent outputs weighted by the expanded router. Grouping its terms by parent gives

$$\widetilde{Y} - Y^0 = \sum_{j=1}^{N+B} \widetilde{g}_j(\widetilde{E}_j - E^0_{\pi(j)}) + \sum_{e=1}^{N} (\bar{g}_e - g_e^0) E_e^0. \tag{16}$$

Apply the triangle inequality, use $\widetilde{g}_j \ge 0$, and bound every source expert norm by M to obtain Equation 15. □

The first term accounts for deviations from parent transformations; the second accounts for changes in parent-aggregated routing weights. Under exact copying, the first term vanishes at initialization, but softmax normalization and selection competition can change the second. Exact copying together with $\bar{g} = g^0$ is sufficient for preserving the layer output, but is not necessary because expert-output errors can cancel. This decomposition motivates Equation 6; it does not guarantee that its optimization finds zero error or that local recovery implies recovery of end-to-end language modeling.

A.4 Properties of Intra-Cluster Routing

Fix a token and let $\mathcal{S}_{\text{init}}$ be the expanded router's initial set of k distinct experts. Expanded groups $\{\mathcal{C}_c^+\}$ form a partition, with each new expert inheriting its parent's group. Process positions once in a fixed order and update the selected set after every replacement. The eligible set defined in Section 3.4 contains the current expert and excludes all other selected experts. Its sampling probabilities are therefore well-defined under finite softmax logits. They may equivalently be computed by a softmax restricted to eligible logits for numerical stability.

Every sampled replacement belongs to the same group as the expert it replaces and cannot duplicate another selected expert. Induction over the positions yields

$$|\mathcal{S}_{\text{final}}| = k, \qquad |\mathcal{S}_{\text{final}} \cap \mathcal{C}_c^+| = |\mathcal{S}_{\text{init}} \cap \mathcal{C}_c^+|, \quad \forall c. \tag{17}$$

Retention steps preserve these properties trivially. This is a structural property of the replacement rule, independent of whether the groups are functionally informative. It concerns counts relative to the expanded router's initial selection, not the source model's selection or mixture weights.

The confidence margin uses the complete group; eligibility exclusions apply only when sampling replacements. A low-confidence position can therefore remain unchanged, including when its eligible set contains only the current expert. Over T continued-training steps, the exploration schedule is $\rho_t = \max(0, 1 - t/T)$; evaluation sets $\rho_t = 0$. Sampling is a discrete training-time operation: gradients pass through selected expert outputs and their mixture weights, without requiring differentiation through sampled indices. Exploration provides possible training opportunities rather than a guarantee of increased new-expert use or specialization. Both effects require empirical evaluation.

Split	Sequences	Input tokens
Training	16,384	33,554,432
Validation	2,048	4,194,304
Test	2,048	4,194,304
Calibration	512	1,048,576

Table B.1: Data budgets per backbone. Calibration is drawn from training data and is not an additional independent corpus.

Task	Shots	Metric
HellaSwag	0	Length-normalized accuracy
ARC-Challenge	0	Length-normalized accuracy
PIQA	0	Accuracy
WinoGrande	0	Accuracy
MMLU	5	Harness aggregate accuracy
GSM8K	8	Exact match, flexible extraction

Table B.2: Downstream protocol. Scores are percentages; Avg. is their unweighted six-task mean. Local MMLU files preserve the task's subject splits and schema.

B Experimental Details and Additional Results

B.1 Data and Evaluation Protocol

Data preparation. We use eight fixed shards from DCLM-baseline-1.0. The pinned revision is `a3b142c183aebe5af344955ae20836eb34dcf69b`. Document pools are constructed before tokenization. Each backbone uses its native tokenizer and 2,048-token packed sequences.

Calibration examples come from training data and are split into fitting and checking subsets. Only fitting examples support descriptor construction, grouping, allocation, parent selection, and warm-start. Checking examples support diagnostics. Table B.1 summarizes the budgets.

Equal token budgets need not cover identical text across tokenizers. DCLM measures performance on the continuation data, rather than either backbone's complete pretraining distribution.

Language modeling. A one-token shift within each sequence gives 4,192,256 prediction targets in the full test split. PPL is the exponential of mean token NLL, not the average of sequence-level PPL.

Intermediate evaluations use the same 256 validation sequences selected with seed 1701, including the source reference. Development uses validation data; final comparisons use the full test split. PPL is compared within the same backbone and tokenizer.

Downstream evaluation. Table B.2 gives the protocol for `lm-eval` 0.4.9.1. Evaluation uses batch size one, no chat template, and a maximum context length of 4,096 tokens or the model limit if smaller. Few-shot selection uses seed 1234. Dataset snapshots, task configurations, and per-example predictions are retained. All evaluations disable exploration and use native top-k routing.

B.2 Baseline Construction and Hyperparameters

Baselines. Original evaluates the unchanged checkpoint. Continue Original updates all parameters without expansion. Random Copy selects parents without replacement until the source pool is exhausted. Traffic Copy selects parents by descending fitting-set assignment counts. Both copy expert parameters and their corresponding router rows.

EU-GN adapts Expert Upcycling (Dwivedi et al., 2026). It scores experts by the squared norm of their mean LM gradient over ten fitting calibration sequences, allocating up to three added copies per parent in descending score order. New router biases receive uniform perturbations in $[-10^{-3}, 10^{-3}]$.

Setting	Value
Hardware	One NVIDIA A800 80GB per run
Precision / attention	BF16 / SDPA
Training seeds	42, 43, 44
Sequence / microbatch	2,048 tokens / one sequence
Accumulation / effective batch	Four microbatches / 8,192 tokens
Continuation steps	4,096
Continuation optimizer	Adafactor; CPU FP32 master parameters and states
Peak learning rate / weight decay	10^{-4} / 0.01
Learning-rate schedule	128-step linear warmup; cosine decay to 10^{-5}
Adafactor settings	No first moment; decay rate -0.8; update clipping 1
Load-balancing coefficient	0.01
Warm-start optimizer	AdamW; learning rate 10^{-4}; weight decay 0
Warm-start budget	64 updates per layer; up to 128 fitting states per update
Warm-start gradient clipping	Global norm 1
Groups / confidence threshold	8 / $\tau = 0.2$
Exploration schedule	Linear decay to zero over continuation
Evaluation / checkpoint interval	Every 512 optimizer steps

Table B.3: Primary hyperparameters. Warm-start samples cached fitting states with replacement. Exact settings are retained in frozen run configurations.

Its continuation protocol matches the other expansion methods, rather than reproducing the complete original training recipe.

GRAFT-MoE. The primary setting uses eight groups per layer, traffic-based integer allocation, and representative-parent selection. Euclidean k-means uses ten initializations and at most 300 iterations. Experts and router rows are copied exactly, without expert-weight noise in the current configuration.

The ablations use random groups, omit warm-start, or disable routing exploration. Random groups preserve group sizes but recompute allocations and parent choices.

Expansion methods update only added experts and all routed-expert router rows, including during warm-start. Inherited experts, attention, embeddings, and Qwen's shared branches remain frozen. Continue Original updates the full model. Equal continuation tokens do not equalize trainable parameters, preparation costs, or elapsed time.