MoE-DisCo: Low Economic Cost Training Mixture-of-Experts Models

Anonymous authors
Paper under double-blind review

Abstract

Training large-scale Mixture-of-Experts (MoE) models faces prohibitive memory capacity constraints. While only a subset of parameters is activated during inference, the entire parameter space must be updated during training. This forces the use of multi-GPU parallelism such as Tensor Parallelism, introducing massive inter-GPU communication overhead that significantly bottlenecks training efficiency. To address this, we propose MoE-DisCo (Mixture-of-Experts with Disentangled Clustering and Coordination), a staged, communication-efficient training framework. MoE-DisCo decomposes the massive MoE model into multiple dense submodels, each consisting of a shared backbone and a single expert, ensuring each submodel fits within the memory of either low-cost GPUs or a single GPU. Concurrently, it partitions the training data into subsets using unsupervised clustering. Each submodel is trained independently and in parallel on its assigned data subset on isolated GPUs, completely eliminating inter-GPU communication overhead. Subsequently, all experts are integrated into a complete MoE model, which is fine-tuned globally for a short period. Experiments demonstrate that MoE-DisCo matches or exceeds the performance of full-parameter training across downstream tasks and in terms of perplexity (PPL), while drastically reducing training costs and communication overhead on architectures such as Qwen-MoE and Llama-MoE.

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.

1 Introduction

Current MoE training paradigms face prohibitive memory constraints. For instance, Qwen1.5-MoE-2.7B activates 2.7 billion parameters per forward pass but has over 14 billion in total, exceeding single-GPU capacity even on an A100 and forcing reliance on multi-GPU strategies such as Tensor Parallelism (TP) or Expert Parallelism (EP). These strategies inevitably introduce substantial inter-device communication overhead, including all-reduce and all-to-all operations. As shown in Figure 1, Model FLOPs Utilization (MFU) per GPU consistently decreases as cluster size increases, indicating that communication bottlenecks severely limit large-scale distributed training.

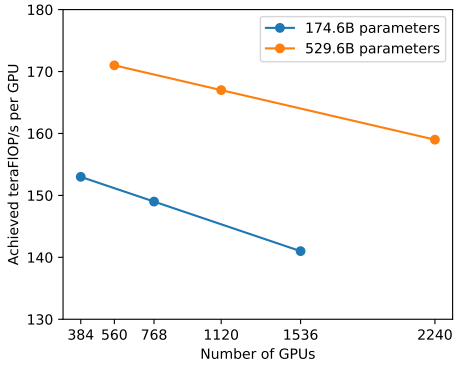


Figure 1: Model FLOPs Utilization (MFU) per GPU under 3D parallelism for language models of different scales as the number of GPUs increases. As the cluster scales from tens to thousands of GPUs, MFU per GPU consistently declines across all model sizes (Narayanan et al., 2021).

These observations highlight a key principle for cost efficiency: training should avoid exclusive reliance on expensive multi-GPU setups, restricting their use to stages that require global coordination. MoE models offer a unique opportunity to realize this principle because their expert modules are naturally separable and can be trained in a more disentangled manner.

Consequently, maximizing the utilization of low-cost single-GPU instances while minimizing expensive multi-GPU coordination is critical for reducing large-scale MoE training costs.

Motivated by this, we propose MoE-DisCo (Mixture-of-Experts with Disentangled Clustering and Coordination), a staged framework that exploits MoE modularity. Unlike methods such as Branch-Train-Mix (BTX) (Sukhbaatar et al., 2024), which rely on human-defined domain labels and overlook latent semantic correlations, MoE-DisCo employs unsupervised K-means clustering to automatically discover feature similarities, guiding expert specialization without human priors.

Given a full MoE model with E experts per MoE layer, MoE-DisCo decomposes it into E independent dense submodels, each consisting of the shared backbone and one expert, while the corpus is partitioned into E clusters. Each submodel trains exclusively on its assigned subset, reducing the parameter footprint enough to fit on a low-cost single GPU. Crucially, this phase is fully parallelizable across independent devices and eliminates inter-device communication overhead.

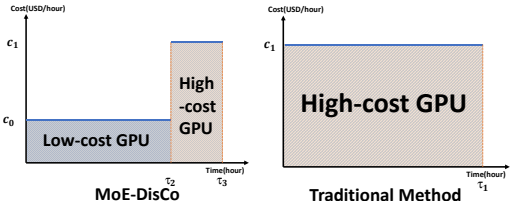


Figure 2: Training cost profiles for MoE-DisCo (left) and full-parameter training (right). Hourly rates are c_0 for low-cost single-GPU devices and c_1 for high-cost multi-GPU devices. Elapsed-time endpoints are τ_1 (baseline completion), τ_2 (submodel training), and τ_3 (global fine-tuning). Shaded areas represent total cost.

MoE-DisCo then reintegrates the trained experts into a unified MoE architecture for a short global fine-tuning stage on the full dataset. Because the prior independent training yields a highly effective parameter initialization, this stage converges rapidly. Although it requires high-cost multi-GPU systems, its brief duration drastically reduces overall expenses compared with conventional training, as illustrated in Figure 2.

Our main contributions are summarized as follows. (1) We propose MoE-DisCo, a staged framework that decouples training to shift the computational bulk from expensive multi-GPU clusters to fully parallelized, low-cost single-GPU instances. (2) We evaluate MoE-DisCo across multiple datasets and model architectures. Experimental results show that MoE-DisCo matches or exceeds conventional full-parameter training while reducing total training cost by 45.1% to 55.1% on Qwen1.5-MoE-2.7B and LLaMA-MoE-3.5B. (3) Comparative analysis demonstrates that our unsupervised partitioning outperforms BTX without requiring human-annotated labels.

2 Related Work

MoE increases model capacity through conditional expert activation (Jacobs et al., 1991; Shazeer et al., 2017). Transformer MoEs such as GShard and Switch Transformer established large-scale sparse training (Lepikhin et al., 2021; Fedus et al., 2022; Roller et al., 2021; Dai et al., 2022), while later systems improved memory management, expert parallelism, and communication (Gray et al., 2017; Narang et al., 2017; Artetxe et al., 2022; Du et al., 2022; Rajbhandari et al., 2020; Kim et al., 2021; He et al., 2021; 2022; Liu et al., 2025; Yu et al., 2024; Jin et al., 2025a;b; Yuan et al., 2025). These advances mainly optimize how a complete MoE is distributed and communicated; reducing its economic cost by changing the training algorithm remains less explored. MoE-DisCo addresses this gap by making most optimization independent of full-model residency and cross-device communication.

3 Method

Our approach draws on the optimization principles of Block Coordinate Descent (BCD), WP-SGD, and SimuParallel SGD (Cheng et al., 2020; Zinkevich et al., 2010). Following BCD, we treat the shared backbone and one expert as a parameter block: only these parameters are updated, while all other experts remain frozen. Each block forms a dense single-expert submodel that fits on one low-cost GPU. Following parallel SGD, these submodels train independently without communication or synchronization, after which their shared parameters are aggregated.

Parameter decoupling alone is insufficient because it does not determine which data each expert should learn from. We therefore cluster sequence representations and assign one semantic cluster to

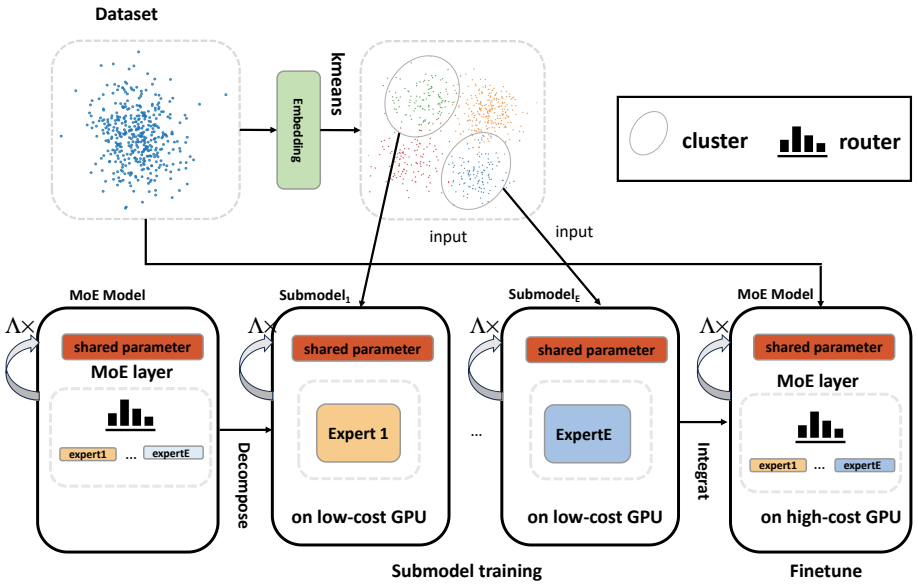


Figure 3: MoE-DisCo clusters the corpus, trains dense single-expert submodels independently, and merges them for short global MoE fine-tuning. Here E is the expert count per layer and Λ is the layer count.

each expert. Each submodel receives a semantically coherent subset, while different experts cover different parts of the corpus. After submodel training, the experts are integrated directly and the backbone copies are averaged with sample-count weights. Because the submodels omit the router, the assembled MoE is globally fine-tuned to coordinate the router, backbone, and experts.

The complete pipeline has four stages (Figure 3): model decomposition constructs E dense submodels, data partitioning assigns one semantic cluster to each submodel, independent training optimizes the submodels without communication, and model reintegration combines the trained components before global fine-tuning on the full corpus.

3.1 Construction of Single-Expert Submodels

Let an MoE contain E experts per layer, with parameters $\Theta = (s, e_1, \ldots, e_E, \phi)$, where s contains the shared backbone parameters, e_k the parameters of expert k across MoE layers, and ϕ the router parameters. Submodel k has parameters $u_k = (s_k, e_k)$: its own backbone copy and one expert, with the router and other experts removed. Each submodel starts from the backbone s_0 and corresponding expert $e_{k,0}$ in the checkpoint Θ^{init}. It trains on $\mathcal{D}_k$ for T_k steps, storing no parameters, gradients, or optimizer states for other experts. The submodels train independently on separate single GPUs or sequentially. After training, s_k and e_k denote the returned parameter blocks.

3.2 Dataset Partitioning

MoE-DisCo assigns semantically similar examples to each submodel without training a separate encoder. For a sequence $x = (x_1, \ldots, x_{N_x})$ of N_x tokens, we average token embeddings from the pretrained checkpoint:

$$h(x) = \frac{1}{N_x} \sum_{i=1}^{N_x} \text{Embedding}(x_i) \tag{3.1}$$

K-means partitions the representations into E subsets $\mathcal{D}_k$, one per expert. This sequence-level hard assignment groups nearby examples without domain labels or a predetermined taxonomy. We record $n_k = |\mathcal{D}_k|$ to determine the backbone-aggregation weights below. Cluster labels only allocate training data; they are neither router targets nor inference inputs. After integration, the top-K_{act} router is optimized on the full corpus, where $1 \le K_{\text{act}} \le E$ is the number of experts activated per token. Appendix G provides the clustering objective, visualization, and cluster statistics.

Algorithm 1 MoE-DisCo training

Require: Corpus $\mathcal{D}$; initialization Θ^{init}; local step budgets $\{T_k\}_{k=1}^{E}$; global step budget S.
Ensure: Trained MoE parameters $\Theta_{D,S}$.

```
h(x) ← MeanPool(Embed(x)) for all x ∈ D
{D_1, ..., D_E} ← K-means({h(x)}; clusters = E)
p_k ← |D_k|/|D| for k = 1, ..., E
for k ← 1 to E do
    u_k ← (s_0, e_{k,0})
    Train u_k on D_k for T_k steps
end for
s̄ ← Σ_{k=1}^{E} p_k s_k
Θ_{D,0} ← (s̄, e_1, ..., e_E, φ_0)
Fine-tune Θ_{D,0} on D for S steps to obtain Θ_{D,S}
return Θ_{D,S}
```

3.3 Submodel Integration and Global Fine-Tuning

After local training, each expert e_k is restored to its original position, while the backbone copies s_k are merged using sample-count weights:

$$\bar{s} = \sum_{k=1}^{E} p_k s_k, \qquad p_k = \frac{n_k}{\sum_{r=1}^{E} n_r}. \tag{3.2}$$

The weights reduce to $1/E$ for balanced clusters. Weighted parallel aggregation is studied in WP-SGD (Cheng et al., 2020); Appendix D considers other fixed merge weights. Restoring the router initialization ϕ_0 gives the merged model $\Theta_{D,0} = (\bar{s}, e_1, \ldots, e_E, \phi_0)$. Global fine-tuning jointly updates all components on the full corpus for S steps, yielding $\Theta_{D,S}$. Only this stage requires the complete MoE and distributed communication.

3.4 MoE-DisCo Algorithm

Algorithm 1 summarizes the complete procedure.

3.5 Clustering Quality and Two-Stage Optimization

MoE-DisCo trains one dense submodel per data cluster, merges their backbones and experts, and then performs global fine-tuning. The analysis asks two questions: how the partition affects the quality of the merged initialization, and whether this initialization reduces the number of synchronized global steps. We model the first stage as finite-step local optimization followed by weighted backbone averaging, while retaining the error caused by heterogeneous cluster objectives (Zinkevich et al., 2010; Li et al., 2020).

Setup. Use the same parameters and weights for the partition $\mathcal{P} = \{\mathcal{D}_k\}_{k=1}^{E}$. The analysis takes $T_k = T$ for all submodels and writes $u_{k,t} = (s_{k,t}, e_{k,t})$ for u_k after t local updates. The returned blocks are $s_{k,T}$ and $e_{k,T}$, and $\bar{s}_T$ denotes the merged backbone. Let $f_k(s, e)$ be the single-expert objective, H their weighted auxiliary objective, and F the routed-MoE objective, with $F_\star = \min_\Theta F(\Theta)$. Appendix A specifies the objectives, updates, and assumptions.

The analysis assumes that every f_k is L-smooth and μ-strongly convex, while F is L_F-smooth in a weighted norm and μ_F-strongly convex. Stochastic gradients are unbiased with bounded conditional variance. Three partition-dependent quantities determine the merge quality: $\mathcal{W}$ measures gradient noise within each cluster, $\mathcal{C}_s$ measures the residual shared-backbone gradient after each expert adapts to its cluster, and $\mathcal{R}$ measures the mismatch between the complete routed objective and the fixed cluster assignment. These definitions include all model blocks and auxiliary losses used by the corresponding objectives.

Merged initialization. Let $q = 1 - \eta\mu$, $\kappa = L/\mu$, and $D_0 = \mathcal{R}/(2\mu_F)$, and let $\mathcal{I}_0$ denote the initial local-gradient error defined in Appendix A. Local SGD first gives the contraction term R_T. Smoothness and parameter merging convert it to an auxiliary-objective error A_T, after which weighted smoothness and router mismatch give the complete-model error B_T:

$$\begin{aligned} R_T &= q^T\mathcal{I}_0 + \frac{\eta\mathcal{W}}{\mu}(1-q^T), \\ A_T &= \frac{L}{2}R_T + \frac{\kappa-1}{2\mu}\mathcal{C}_s, \qquad B_T = \left(\sqrt{D_0} + \sqrt{\frac{L_F}{\mu}A_T}\right)^2. \end{aligned} \tag{3.3}$$

The factor $q^T\mathcal{I}_0$ vanishes geometrically with the number of local steps T, while within-cluster noise, residual backbone conflict, and routing discrepancy remain as partition-dependent terms.

Theorem 3.1 (Merged initialization)**.** *Under the conditions in Appendix A, the merged initialization $\Theta_{D,0}$ satisfies*

$$\mathbb{E}[F(\Theta_{D,0}) - F_\star] \le B_T. \tag{3.4}$$

Let $\mathcal{E}_0 = F(\Theta^{\text{init}}) - F_\star$ and $\underline{\mathcal{E}}_0 = \|\nabla F(\Theta^{\text{init}})\|_{p,*}^2/(2L_F)$. If $B_T < \underline{\mathcal{E}}_0$, the merged model has a strictly smaller expected objective than the common initialization.

Algorithm convergence. Write $\Theta_{a,j}$ for the full model after j global updates, with $a = D$ for MoE-DisCo and $a = B$ for the baseline, which starts at $\Theta_{B,0} = \Theta^{\text{init}}$. Let $\rho = 1 - \alpha\mu_F$ and $\nu = L_F\alpha\sigma_F^2/(2\mu_F)$, where σ_F^2 bounds the gradient noise during global fine-tuning. The second stage of Algorithm 1 contracts at rate ρ toward the stochastic noise floor ν, with the first-stage initialization entering through B_T.

Theorem 3.2 (Convergence of MoE-DisCo)**.** *Under the conditions of Theorem 3.1, after S global SGD steps, the MoE-DisCo iterates and the full-parameter baseline satisfy*

$$\begin{aligned} \mathbb{E}[F(\Theta_{D,S}) - F_\star] &\le \nu + \rho^S(B_T - \nu), \\ \mathbb{E}[F(\Theta_{B,S}) - F_\star] &\le \nu + \rho^S(\mathcal{E}_0 - \nu). \end{aligned} \tag{3.5}$$

Hence, when $B_T < \underline{\mathcal{E}}_0 \le \mathcal{E}_0$, the sufficient iteration count for any target above the noise floor cannot exceed that of the baseline. Appendix B gives the one-step recursion, closed-form counts, and their gap. Appendix F gives a vanishing-error result for decaying step sizes.

Under the stated smoothness, strong convexity, and fixed-step SGD assumptions, B_T reduces the sufficient synchronized iteration count when the initialization condition holds, without changing ρ or ν. The experiments use AdamW on nonconvex MoE models, so these bounds provide a conditional mechanism explanation, not a guarantee of the observed loss curves or cost reductions.

4 Experiment

In this work, we conduct a systematic comparison between MoE-DisCo and a full-parameter trained MoE baseline, evaluating both model performance and training cost. For performance evaluation, we assess language modeling quality via training loss (on the training set), PPL (on the validation set), and downstream tasks, demonstrating that MoE-DisCo matches or enhances algorithmic effectiveness relative to standard MoE training. Our economic cost analysis provides a comprehensive accounting of resource consumption across all training stages. By offloading the bulk of the computational burden to single-GPU setups, the sub-expert training phase yields a highly effective parameter initialization. This strong starting point drives rapid convergence, substantially shortening the required fine-tuning duration and drastically cutting multi-GPU expenses. Moreover, we specifically compare our unsupervised clustering against the BTX paradigm to validate the superiority of our data partitioning strategy, and in ablation study we evaluate the framework's scalability under varying expert counts.

4.1 Experimental Setup

Models

Qwen1.5-MoE-2.7B. The Qwen1.5-MoE-2.7B model activates approximately 2.7 billion parameters during inference. In our experiments, we set the number of experts to 4 and increase the number

of layers, so that the model remains large enough to require multi-GPU training. This brings the total parameter count to 10 billion. It is important to emphasize that this entire parameter space must be updated during training; therefore, the smaller size of the activated parameters is irrelevant to the actual training workload and memory overhead. This model is hereafter abbreviated as Qwen.

LLaMA-MoE-3.5B. The LLaMA-MoE-3.5B model activates approximately 3.5 billion parameters during inference. In our experiments, we likewise set the number of experts to 4 and increase the number of layers, bringing the total parameter count to 10 billion. Consequently, it presents the same massive memory overhead and computational workload during the training phase, as all parameters must be updated simultaneously. This model is hereafter abbreviated as Llama.

Table 1: Performance comparison between MoE-DisCo and full-parameter training across two MoE architectures. The table shows that the number of steps required for MoE-DisCo to achieve full-parameter convergence result.

Model	**Data**	**Full-Parameter**			**MoE-DisCo**		
		Step	Loss	PPL	Step	Loss	PPL
Qwen	C4	21,150	4.954	230.32	4,100	4.925	165.86
	WikiText	12,500	2.303	71.89	3,150	2.296	57.55
	OpenWebText	55,200	4.545	93.71	17,400	4.506	90.13
Llama	C4	11,750	5.274	356.28	4,150	5.276	296.35
	WikiText	15,000	3.805	160.91	5,300	3.794	160.27
	OpenWebText	21,000	4.536	114.93	7,800	4.541	106.82

Table 2: Comprehensive downstream task performance of different MoE architectures pre-trained on various corpora. (ARC-e and MMLU are evaluated under 5-shot settings; HellaSwag and PIQA under zero-shot).

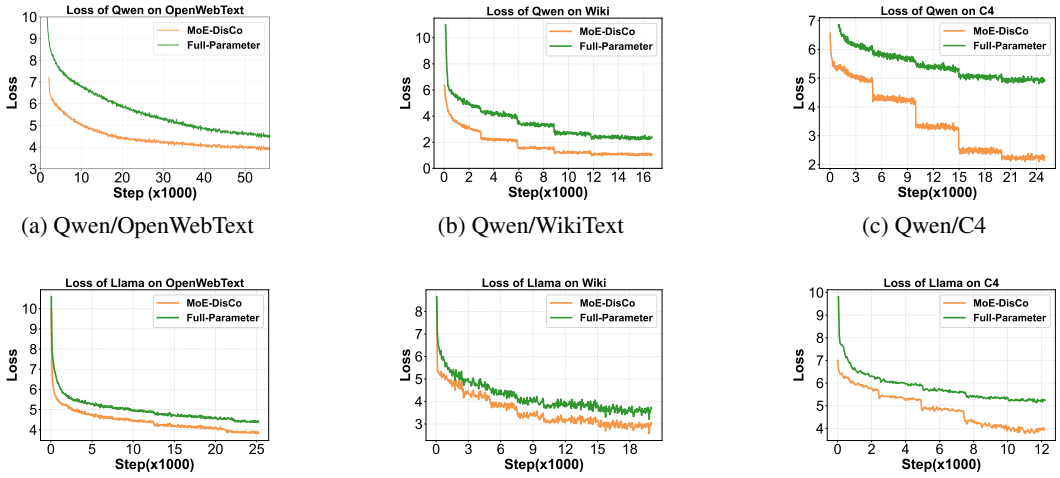
Model	**Corpus**	**Method**	**ARC-e (5)**	**MMLU (5)**	**HellaSwag (%)**	**PIQA (%)**
Qwen	C4	Full-Parameter	27.9	23.0	27.4	52.2
		MoE-DisCo	29.1	25.3	29.4	57.0
	OpenWebText	Full-Parameter	31.5	29.2	23.7	55.3
		MoE-DisCo	34.8	30.6	26.1	57.8
Llama	C4	Full-Parameter	28.1	28.1	22.5	53.7
		MoE-DisCo	29.4	28.5	23.4	53.9
	OpenWebText	Full-Parameter	28.4	28.2	22.9	53.6
		MoE-DisCo	30.0	28.8	23.7	54.1

DATASETS

To evaluate our method, we utilize three standard public datasets. OpenWebText, an open web text collection constructed following the GPT-2 training data, serves as our primary training corpus. To further verify the generalization and robustness of the MoE-DisCo framework across different data distributions, we conduct supplementary validation on two additional datasets: C4, a large-scale English corpus with strict filtering and cleaning, and WikiText, a high-quality English dataset compiled from Wikipedia.

EVALUATION METRICS

We evaluate MoE-DisCo against the full-parameter trained MoE baseline using two key criteria: (1) language modeling capability, measured by training loss, perplexity (PPL), and downstream task performance; and (2) overall economic cost, evaluated by the total financial expenditure (in USD) encompassing both the sub-expert training and global fine-tuning phases. To ensure a rigorous and fair comparison, the reported training loss represents the pure token-level cross-entropy, explicitly excluding any auxiliary MoE routing losses. Correspondingly, PPL is evaluated on the validation set using independent, non-overlapping sequences of 1024 tokens. By leveraging the pre-trained sub-experts as a high-quality starting point, MoE-DisCo either achieves a lower loss at an equivalent step count or reaches a target loss in significantly fewer steps than the baseline, directly translating to substantial cost savings on high-cost multi-GPU devices.



(a) Qwen/OpenWebText (b) Qwen/WikiText (c) Qwen/C4

(d) Llama/OpenWebText (e) Llama/WikiText (f) Llama/C4

Figure 4: Loss trends between MoE-Disco on fine-tune stage and Full-Parameter across different datasets.

4.2 Performance Comparison

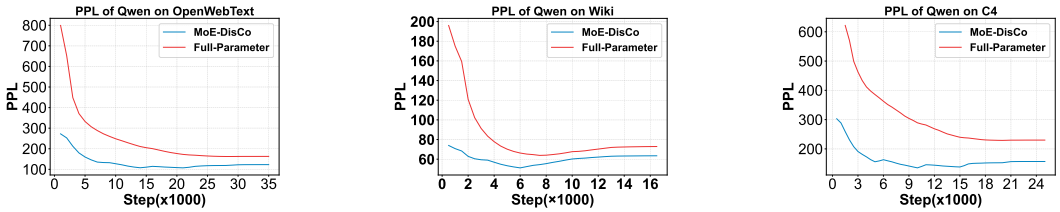
Table 1 presents a comparison of training efficiency and convergence performance between MoE-DisCo and full-parameter training across two MoE architectures (Qwen and Llama) and three datasets (C4, WikiText, and OpenWebText). The results demonstrate that MoE-DisCo substantially reduces the number of training steps required to reach the final convergence loss of the full-parameter baseline. For instance, on the Qwen architecture, the steps needed to achieve this target loss decrease from 21,150, 12,500, and 55,200 to just 4,100, 3,150, and 17,400 on the C4, WikiText, and OpenWebText datasets, demonstrating that MoE-DisCo effectively reduces the iterations needed to reach the target performance. A similar trend is observed for the Llama architecture. Moreover, upon reaching the same training loss, the PPL of models trained by the MoE-DisCo method is lower than that of the full-parameter baseline, indicating that our method significantly accelerates training without compromising language modeling performance.

Figures 4 and 5 show the training loss and PPL curves of the Full-Parameter and MoE-DisCo fine-tune stage methods on three datasets, respectively. We use fine-tune stage of MoE-DisCo because this part is the most expensive stage (cost information is shown in Economic Cost Analysis section). It can be observed that, given the same training duration, models trained with the MoE-DisCo method achieve significantly better performance than those trained with full-parameter training. If the target is to reach the same model performance, the MoE-DisCo method substantially reduces the required training time. This indicates that the proposed method can markedly lower training costs without sacrificing model performance. The whole training process including submodel training is shown in Cost Analysis part and Appendix.

4.3 Downstream Task Performance

Table 2 reports the downstream task performance of MoE-DisCo and full-parameter training. The models are pre-trained on C4 or OpenWebText and evaluated on multiple downstream tasks. ARC-e and MMLU are evaluated in the 5-shot setting, while HellaSwag and PIQA are evaluated in the zero-shot setting.

As shown in the table, MoE-DisCo consistently achieves better performance than full-parameter training on both few-shot (ARC-e, MMLU) and zero-shot (HellaSwag, PIQA) downstream tasks. Overall, the results show that MoE-DisCo achieves better downstream task performance than full-parameter training across the evaluated benchmarks.



(a) Qwen/OpenWebText (b) Qwen/WikiText (c) Qwen/C4

(d) Llama/OpenWebText (e) Llama/WikiText (f) Llama/C4

Figure 5: PPL trends between MoE-DisCo on fine-tune stage and Full-Parameter across different datasets

Table 3: Comparison of Full-Parameter and MoE-DisCo training costs, times, and platforms for Qwen and Llama across three datasets. All costs are reported in USD and all times in hours. Prefixes denote the training stage: S- for submodel training, F- for the fine-tune stage, and T- for the total across both stages. S-Cost aggregates the cost of training all submodels.

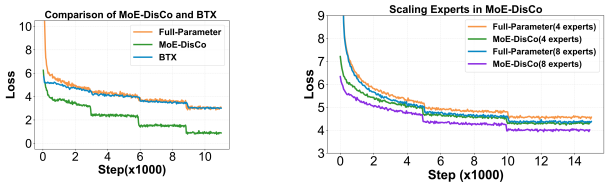
Model	Dataset	Full-Parameter			MoE-DisCo							
		Cost	Time	Platform	S-Cost	S-Time	S-Platform	F-Cost	F-Time	F-Platform	Total Cost	Total Time
Qwen	C4	83.27	9.13	A100*4	12.68	2.42	A800*4	24.72	2.71	A100*4	37.40	5.13
	Wiki	29.46	3.23	A100*4	5.08	0.97	A800*4	8.30	0.91	A100*4	13.38	1.88
	OpenWebText	380.40	41.71	A100*4	66.50	12.69	A800*4	117.92	12.93	A100*4	184.42	25.62
Llama	C4	62.56	6.86	A100*4	8.96	1.71	A800*4	21.80	2.39	A100*4	30.76	4.10
	Wiki	44.41	4.87	A100*4	8.70	1.66	A800*4	15.60	1.71	A100*4	24.29	3.37
	OpenWebText	203.10	22.27	A100*4	36.26	6.92	A800*4	75.15	8.24	A100*4	111.41	15.16

4.4 Economic Cost Analysis

Training large-scale models, especially those based on MoE architectures, incurs substantial computational and memory costs. This section analyzes the economic advantages of the proposed MoE-DisCo training strategy and compares it with conventional full-parameter training methods.

Following the market pricing of major cloud service providers (Infinigence AI) and considering the hardware environment used in our experiments, we adopt the following GPU rental prices as the cost estimation baseline: A100 (80GB): \$2.28 / GPU·hour, A800 (80GB): \$1.31 / GPU·hour.

Based on the results in Table 3, MoE-DisCo achieves model performance comparable to full-parameter training, while substantially reducing cost and training time. The approach decouples MoE training into two phases: Submodel Training (S-phase) and Lightweight Fine-tuning (F-phase). During the S-phase, submodels are trained independently in parallel on single-GPU instances (predominantly incorporating more cost-effective A800 GPUs), eliminating the need for 4-GPU synchronized clusters. In the S-phase, 4 submodels are trained simultaneously on 4 independent single GPUs. Thus, the wall-clock time is determined by the slowest submodel ($\tau_{\max}$). For a conservative economic estimate, we calculate the total S-phase cost as $4 \times \tau_{\max}$ GPU-hours, billing all parallel instances at the maximum duration. Due to the reduced computational footprint, this phase incurs minimal costs ranging from \$5.08 to \$66.50. The subsequent F-phase requires only a brief synchronized fine-tuning step (costing \$8.30 to \$117.92) to fully recover model convergence. Overall, MoE-DisCo demonstrates significant efficiency gains. For Qwen—which requires updating a total parameter space exceeding 10B—costs on the C4, WikiText, and the primary OpenWebText datasets are drastically reduced by 55.1%, 54.6%, and 51.5% respectively, cutting total training time by 43.8%, 41.8%, and 38.6%. Similarly, on the Llama architecture, MoE-DisCo reduces total costs by 45.1% to 50.8% across all datasets, alongside robust training time reductions of 30.8% to 40.2%.



(a) Data partitioning. (b) Expert-count scaling.

Figure 6: MoE-DisCo ablations on Qwen/C4. Left: replacing the clustering operation with BTX-style data assignment causes fine-tuning performance to degrade toward full-parameter training. Right: MoE-DisCo performance with 4 and 8 experts.

By shifting the bulk of the computational workload from monolithic high-cost multi-GPU clusters to low-cost single-GPU instances, MoE-DisCo offers a highly cost-efficient framework for training large-scale MoE models without compromising final performance.

4.5 Comparison with BTX

In this experiment, we compare MoE-DisCo against the BTX paradigm to evaluate the effectiveness of our data partitioning strategy. Specifically, during the training of the Qwen model on C4, we removed the original k-means clustering step and replaced it with a human assignment of data to different experts for submodel training. We assign different datasets to different experts based on our knowledge of different domains, i.e., BTX style. The results, Figure 6a, show that substituting k-means clustering with BTX style assignment significantly degrades training efficiency, and the final performance essentially regresses to that of full-parameter training. The downstream task results, in Table 4, also show that our methods outperform the output from BTX method. These results indicate that our proposed method does not require explicitly partitioning the dataset by domain; instead, it can still effectively capture latent semantic similarities automatically.

Table 4: Comprehensive downstream task performance of MoE-DisCo and BTX. (ARC-e and MMLU are evaluated under 5-shot settings; HellaSwag and PIQA under zero-shot).

Method	**ARC-e**	**MMLU**	**HellaSwag**	**PIQA**
BTX	25.5	23.1	25.8	53.4
MoE-DisCo	26.1	23.2	26.0	53.7

4.6 Ablation Study: Multi-experts in MoE-DisCo

In this ablation study, we evaluate MoE-DisCo's performance with varying expert counts to assess how architectural capacity affects training efficiency and model quality. Specifically, we compare the convergence of the Qwen on C4 using configurations of 4 and 8 experts. As shown in Figure 6b, MoE-DisCo maintains fast convergence and strong final performance in both settings. Notably, the 8-expert model yields lower loss than the 4-expert version. These results demonstrate that our proposed framework scales robustly, remaining highly effective and efficient even in more complex architectural configurations with a larger number of experts.

5 Conclusion

We introduced MoE-DisCo, a staged framework that decomposes an MoE into independently trained single-expert submodels and reserves synchronized multi-GPU training for a short global fine-tuning stage. Across Qwen- and Llama-based MoEs on C4, WikiText-2, and OpenWebText, it reaches comparable or better language-modeling and downstream performance than full-parameter training while reducing total monetary cost by 45.1%–55.1%. The BTX comparison links part of this benefit to learned semantic partitioning, while the eight-expert ablation shows that the staged procedure remains effective. These results show that most MoE optimization can move to communication-free, lower-cost hardware without sacrificing performance.

AI USE STATEMENT

AI tools were used to assist with language polishing, consistency checking, selected aspects of data analysis, and code implementation checks. They were also used to help review experimental outputs, identify potential inconsistencies, and improve the clarity and presentation of the manuscript.

ETHICS STATEMENT

This work studies a training method using publicly available text corpora and established evaluation benchmarks. Such corpora may contain biased, harmful, copyrighted, or personally identifying material, and models trained on them may reproduce these patterns. The reported evaluations measure language-modeling and downstream-task performance; they do not assess privacy, bias, or misuse risks. Lower training costs may broaden access to MoE research while also reducing the resources needed for harmful applications. These risks remain relevant when models trained with this method are developed or deployed.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe data partitioning, submodel training, integration, and global fine-tuning. Section 3.5 and Appendices A–F give the optimization assumptions and proofs. Section 4.1 reports the models, datasets, and evaluation metrics; Appendix G details the clustering procedure and C4 partition statistics; Appendices H–J report hyperparameters, submodel training steps and times, and the gating architecture. Table 3 states the hardware, stage times, and costs used in the economic comparison. We will provide the code in the supplementary materials.

REFERENCES

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11699–11732, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.804. URL https://arxiv.org/abs/2112.10684.

Daning Cheng, Shigang Li, and Yunquan Zhang. WP-SGD: Weighted parallel SGD for distributed unbalanced-workload training system. *Journal of Parallel and Distributed Computing*, 145:202–216, 2020. doi: 10.1016/j.jpdc.2020.06.011. URL https://shigangli.github.io/files/jpdc2020.pdf.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7085–7095, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.489. URL https://aclanthology.org/2022.acl-long.489/.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5547–5569. PMLR, 2022. URL https://proceedings.mlr.press/v162/du22c.html.

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.

Scott Gray, Alec Radford, and Diederik P. Kingma. GPU kernels for block-sparse weights. Technical report, OpenAI, 2017. URL https://cdn.openai.com/blocksparse/blocksparsepaper.pdf.

Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang. FastMoE: A fast mixture-of-expert training system. arXiv preprint arXiv:2103.13262, 2021. URL https://arxiv.org/abs/2103.13262.

Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. FasterMoE: Modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, pp. 120–134, New York, NY, USA, 2022. Association for Computing Machinery. doi: 10.1145/3503221.3508418. URL https://pacman.cs.tsinghua.edu.cn/~zjd/publication/generated/dblp-confppopp-he-zawlsl-22/.

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. URL https://www.cs.toronto.edu/~hinton/absps/jjnh91.pdf.

Chao Jin, Ziheng Jiang, Zhihao Bai, Zheng Zhong, Juncai Liu, Xiang Li, Ningxin Zheng, Xi Wang, Cong Xie, Qi Huang, Wen Heng, Yiyuan Ma, Wenlei Bao, Size Zheng, Yanghua Peng, Haibin Lin, Xuanzhe Liu, Xin Jin, and Xin Liu. MegaScale-MoE: Large-scale communication-efficient training of mixture-of-experts models in production. arXiv preprint arXiv:2505.11432, 2025a. URL https://arxiv.org/abs/2505.11432.

Zewen Jin, Shengnan Wang, Jiaan Zhu, Hongrui Zhan, Youhui Bai, Lin Zhang, Zhenyu Ming, and Cheng Li. BigMac: A communication-efficient mixture-of-experts model structure for fast training and inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (17):17689–17698, 2025b. doi: 10.1609/aaai.v39i17.33945. URL https://ojs.aaai.org/index.php/AAAI/article/view/33945.

Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient MoE training for multitask multilingual models. arXiv preprint arXiv:2109.10465, 2021. URL https://arxiv.org/abs/2109.10465.

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.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations*, 2020. URL https://wangshusen.github.io/papers/ICLR2020.pdf.

Xinyi Liu, Yujie Wang, Fangcheng Fu, Xupeng Miao, Shenhan Zhu, Xiaonan Nie, and Bin Cui. NetMoE: Accelerating MoE training through dynamic sample placement. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e0c256700465c158de71081b4cf5e8c3-Paper-Conference.pdf.

Sharan Narang, Eric Undersander, and Gregory Diamos. Block-sparse recurrent neural networks. arXiv preprint arXiv:1711.02782, 2017. URL https://arxiv.org/abs/1711.02782.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on GPU clusters using Megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Association for Computing Machinery, 2021. doi: 10.1145/3458817.3476209. URL https://arxiv.org/abs/2104.04473.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020. doi: 10.1109/SC41405.2020.00024. URL https://arxiv.org/abs/1910.02054.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021. URL https://arxiv.org/abs/2106.04426.

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.

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, and Xian Li. Branch-Train-MiX: Mixing expert LLMs into a mixture-of-experts LLM. In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2403.07816.

Dianhai Yu, Liang Shen, Hongxiang Hao, Weibao Gong, Huachao Wu, Jiang Bian, Lirong Dai, and Haoyi Xiong. MoESys: A distributed and efficient mixture-of-experts training and inference system for internet services. *IEEE Transactions on Services Computing*, 17(5):2626–2639, 2024. doi: 10.1109/TSC.2024.3399654. URL https://arxiv.org/abs/2205.10034.

Yueming Yuan, Ahan Gupta, Jianping Li, Sajal Dash, Feiyi Wang, and Minjia Zhang. X-MoE: Enabling scalable training for emerging mixture-of-experts architectures on HPC platforms. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1315–1331. Association for Computing Machinery, 2025. doi: 10.1145/3712285.3759886. URL https://arxiv.org/abs/2508.13337.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 23, pp. 2595–2603. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/hash/abea47ba24142ed16b7d8fbf2c740e0d-Abstract.html.

A Preliminary Lemmas: Local Contraction and Parameter Merging

A.1 Formal Setup and Assumptions

Let $\mathcal{P} = \{\mathcal{D}_k\}_{k=1}^{E}$ partition the fixed corpus $\mathcal{D}$, and let ξ be a sequence sampled uniformly from this corpus. Its cluster label is $C(\xi)$, so $p_k = \Pr(C = k) = n_k / \sum_r n_r > 0$ and $\sum_k p_k = 1$. Use the shared backbone s, expert blocks e_k, and router ϕ defined in Section 3.1. Write $z = (s, e_1, \ldots, e_E)$ and $\Theta = (z, \phi)$. The cluster objectives, their fixed-assignment auxiliary objective, and the complete routed objective are

$$\begin{aligned} f_k(s, e) &= \mathbb{E}[\ell(s, e; \xi) \mid C = k], \\ H(z) &= \sum_k p_k f_k(s, e_k), \\ F(\Theta) &= \mathbb{E}[\ell_F(\Theta; \xi)]. \end{aligned} \tag{A.1}$$

Here e denotes the expert block of a generic single-expert submodel, and $F_\star = \min_\Theta F(\Theta)$. All objectives include their corresponding regularizers and auxiliary losses and use the same normalization as their sample gradients.

Both procedures start from the fixed model $\Theta^{\text{init}} = (s_0, e_{1,0}, \ldots, e_{E,0}, \phi_0)$. The analysis specializes Algorithm 1 to $T_k = T$ for every k and to local and global SGD updates. Write $u_{k,t} = (s_{k,t}, e_{k,t})$ for the submodel parameters u_k after t updates. Starting at $u_{k,0} = (s_0, e_{k,0})$, local SGD follows

$$u_{k,t+1} = u_{k,t} - \eta g_{k,t}, \qquad 0 \le t < T. \tag{A.2}$$

At the end of local training, the blocks s_k, e_k in Algorithm 1 are $s_{k,T}, e_{k,T}$. Denote the merged backbone $\bar{s}$ by $\bar{s}_T$ for this common local budget:

$$\begin{aligned} \bar{s}_T &= \sum_k p_k s_{k,T}, \\ \Theta_{D,0} &= (\bar{s}_T, e_{1,T}, \ldots, e_{E,T}, \phi_0). \end{aligned} \tag{A.3}$$

The trajectory label $a = D$ denotes MoE-DisCo and $a = B$ the baseline; j indexes global steps and S is their total budget. The second stage applies ordinary SGD from $\Theta_{D,0}$, while the baseline starts from $\Theta_{B,0} = \Theta^{\text{init}}$:

$$\Theta_{a,j+1} = \Theta_{a,j} - \alpha G_{a,j}, \qquad a \in \{D, B\}. \tag{A.4}$$

Assumption A.1 (Optimization geometry)**.** *Each f_k is finite, differentiable, L-smooth, and μ-strongly convex in the Euclidean norm, where $0 < \mu \le L$. The complete objective F is finite, differentiable, and Euclidean μ_F-strongly convex. It also satisfies*

$$F(\Theta + d) \le F(\Theta) + \langle \nabla F(\Theta), d \rangle + \frac{L_F}{2}\|d\|_p^2, \tag{A.5}$$

where

$$\begin{aligned} \|d\|_p^2 &= \|d_s\|^2 + \sum_k p_k \|d_{e_k}\|^2 + \|d_\phi\|^2, \\ \|\zeta\|_{p,*}^2 &= \|\zeta_s\|^2 + \sum_k p_k^{-1} \|\zeta_{e_k}\|^2 + \|\zeta_\phi\|^2. \end{aligned} \tag{A.6}$$

The step sizes satisfy $0 < \eta < 1/L$ and $0 < \alpha < 1/L_F$, and all initial parameters are finite.

Because $\|d\|_p \le \|d\|$, Equation (A.5) also supplies the Euclidean descent inequality used by ordinary SGD. The geometry of F is an independent condition and does not follow from parameter concatenation.

Assumption A.2 (Stochastic gradients and within-cluster noise)**.** *For filtrations $\mathcal{F}_t$ and $\mathcal{G}_{a,j}$ containing the corresponding training histories,*

$$\begin{aligned} \mathbb{E}[g_{k,t} \mid \mathcal{F}_t] &= \nabla f_k(u_{k,t}), \\ \mathbb{E}[\|g_{k,t} - \nabla f_k(u_{k,t})\|^2 \mid \mathcal{F}_t] &\le w_k, \\ \mathbb{E}[G_{a,j} \mid \mathcal{G}_{a,j}] &= \nabla F(\Theta_{a,j}), \\ \mathbb{E}[\|G_{a,j} - \nabla F(\Theta_{a,j})\|^2 \mid \mathcal{G}_{a,j}] &\le \sigma_F^2. \end{aligned} \tag{A.7}$$

The constants w_k are uniform over the local trajectories, σ_F^2 is valid for both complete-model trajectories, and

$$\mathcal{W} = \sum_k p_k w_k. \tag{A.8}$$

The quantity $\mathcal{W}$ includes gradient noise from the entire single-expert submodel. The main theorem does not require independence among local sampling streams.

Let $u_k^\star = (s_k^\star, e_k^\star) = \arg\min f_k$, $f_k^\star = f_k(u_k^\star)$, $z_H^\star = \arg\min H$, and $H_\star = H(z_H^\star)$. Choose a fixed reference backbone s^{ref} and let $e_k^{\text{ref}} = \arg\min_e f_k(s^{\text{ref}}, e)$. The backbone-residual statistic is

$$\mathcal{C}_s = \sum_k p_k \|\nabla_s f_k(s^{\text{ref}}, e_k^{\text{ref}})\|^2. \tag{A.9}$$

For any fixed approximate experts $\widehat{e}_k$, define the alternative statistic

$$\widehat{\mathcal{C}}_s = \sum_k p_k \|\nabla f_k(s^{\text{ref}}, \widehat{e}_k)\|^2 \tag{A.10}$$

The two statistics are not ordered in general. Lemma A.3 gives both $\Gamma \le \mathcal{C}_s/(2\mu)$ and $\Gamma \le \widehat{\mathcal{C}}_s/(2\mu)$, so either may be used in the subsequent bounds.

Assumption A.3 (Router-statistic compatibility)**.** *Let $\widetilde{G}_H(z;\xi)$ be the gradient of $\ell(s, e_{C(\xi)}; \xi)$ embedded in the complete parameter space, with unselected expert blocks and the router block set to zero. Let $G_F(z, \phi_0; \xi) = \nabla_\Theta \ell_F(z, \phi_0; \xi)$. Assume*

$$\mathbb{E}_\xi \|G_F(z_H^\star, \phi_0; \xi) - \widetilde{G}_H(z_H^\star; \xi)\|^2 \le \mathcal{R}. \tag{A.11}$$

A uniform bound on a deterministic region containing $z_H^\star$ is sufficient. The statistic $\mathcal{R}$ includes changes in predictions, gradients, and routing auxiliary terms.

Finally, the initial local-gradient term used in Equation (A.13) is

$$\mathcal{I}_0 = \frac{1}{\mu^2} \sum_k p_k \|\nabla f_k(u_{k,0})\|^2. \tag{A.12}$$

For completeness, define the quantities used by the main-text statements:

$$\begin{aligned} q &= 1 - \eta\mu, \qquad \kappa = L/\mu, \qquad D_0 = \mathcal{R}/(2\mu_F), \\ R_T &= q^T \mathcal{I}_0 + \frac{\eta \mathcal{W}}{\mu}(1 - q^T), \\ A_T &= \frac{L}{2} R_T + \frac{\kappa - 1}{2\mu} \mathcal{C}_s, \\ B_T &= \left(\sqrt{D_0} + \sqrt{\frac{L_F}{\mu} A_T} \right)^2. \end{aligned} \tag{A.13}$$

Let

$$\mathcal{E}_0 = F(\Theta^{\text{init}}) - F_\star, \qquad \underline{\mathcal{E}}_0 = \frac{\|\nabla F(\Theta^{\text{init}})\|_{p,*}^2}{2L_F}, \tag{A.14}$$

and define the persistent first-stage term

$$A_\infty = \frac{L\eta\mathcal{W}}{2\mu} + \frac{\kappa - 1}{2\mu} \mathcal{C}_s. \tag{A.15}$$

The sufficient statistical condition for finite-step local pretraining is

$$D_0 < \underline{\mathcal{E}}_0, \qquad A_\infty < \frac{\mu}{L_F} \left(\sqrt{\underline{\mathcal{E}}_0} - \sqrt{D_0} \right)^2. \tag{A.16}$$

For the second stage, let $\rho = 1 - \alpha\mu_F$ and $\nu = L_F \alpha \sigma_F^2/(2\mu_F)$. Then

$$\begin{aligned} \mathbb{E}[F(\Theta_{D,S}) - F_\star] &\le \nu + \rho^S (B_T - \nu), \\ \mathbb{E}[F(\Theta_{B,S}) - F_\star] &\le \nu + \rho^S (\mathcal{E}_0 - \nu). \end{aligned} \tag{A.17}$$

A.2 Preliminary Lemmas

In this section, $\mathbb{E}_t$ denotes expectation conditional on the information available immediately before the corresponding update. The quadratic lower bound of any finite-valued, continuous, strongly convex function implies bounded sublevel sets, attainment of its minimum, and uniqueness of its minimizer. The condition $p_k > 0$ ensures that the weighted norm is nondegenerate.

Lemma A.1 (Basic inequalities for smooth strongly convex functions)**.** *Let f be Euclidean L-smooth and μ-strongly convex, with minimizer $u^\star$. Then, for any u,*

$$\frac{\|\nabla f(u)\|^2}{2L} \le f(u) - f(u^\star) \le \frac{\|\nabla f(u)\|^2}{2\mu}, \qquad \|u - u^\star\|^2 \le \frac{\|\nabla f(u)\|^2}{\mu^2}. \tag{A.18}$$

When $0 < \eta \le 1/L$,

$$\|u - \eta\nabla f(u) - u^\star\|^2 \le (1 - \eta\mu)\|u - u^\star\|^2. \tag{A.19}$$

Proof. Smoothness gives

$$f\left(u - \frac{1}{L}\nabla f(u)\right) \le f(u) - \frac{1}{2L}\|\nabla f(u)\|^2. \tag{A.20}$$

The left-hand side is at least $f(u^\star)$, which proves the first inequality in Equation (A.18). Strong convexity gives

$$f(v) \ge f(u) + \langle \nabla f(u), v - u \rangle + \frac{\mu}{2}\|v - u\|^2. \tag{A.21}$$

Taking the infimum over v yields $f(u^\star) \ge f(u) - \|\nabla f(u)\|^2/(2\mu)$. Strong monotonicity of the gradient and $\nabla f(u^\star) = 0$ further give

$$\mu\|u - u^\star\|^2 \le \langle \nabla f(u), u - u^\star \rangle \le \|\nabla f(u)\| \|u - u^\star\|, \tag{A.22}$$

which proves the distance bound, including the trivial case $u = u^\star$.

Finally, strong convexity implies

$$\langle \nabla f(u), u - u^\star \rangle \geq f(u) - f(u^\star) + \frac{\mu}{2}\|u - u^\star\|^2. \tag{A.23}$$

Expanding the squared distance and using $\|\nabla f(u)\|^2 \leq 2L[f(u) - f(u^\star)]$ gives

$$\begin{aligned}\|u - \eta\nabla f(u) - u^\star\|^2 &\leq (1 - \eta\mu)\|u - u^\star\|^2 \\ &\quad - 2\eta(1 - \eta L)[f(u) - f(u^\star)] \\ &\leq (1 - \eta\mu)\|u - u^\star\|^2. \end{aligned} \tag{A.24}$$

□

Lemma A.2 (Finite-step local SGD error)**.** *Under Assumptions A.1 and A.2, the update in Equation* (A.2) *satisfies*

$$\sum_k p_k \mathbb{E}\|u_{k,T} - u_k^\star\|^2 \leq R_T. \tag{A.25}$$

Proof. Let $\omega_{k,t} = g_{k,t} - \nabla f_k(u_{k,t})$. Conditional unbiasedness removes the cross term in the squared-distance expansion, and Lemma A.1 gives

$$\begin{aligned}\mathbb{E}_t\|u_{k,t+1} - u_k^\star\|^2 &= \|u_{k,t} - \eta\nabla f_k(u_{k,t}) - u_k^\star\|^2 + \eta^2\mathbb{E}_t\|\omega_{k,t}\|^2 \\ &\leq q\|u_{k,t} - u_k^\star\|^2 + \eta^2 w_k. \end{aligned} \tag{A.26}$$

Iterating the recursion and using $1 - q = \eta\mu$ yields

$$\mathbb{E}\|u_{k,T} - u_k^\star\|^2 \leq q^T\|u_{k,0} - u_k^\star\|^2 + \frac{\eta w_k}{\mu}(1 - q^T). \tag{A.27}$$

By Equation (A.18),

$$\sum_k p_k\|u_{k,0} - u_k^\star\|^2 \leq \frac{1}{\mu^2}\sum_k p_k\|\nabla f_k(u_{k,0})\|^2 = \mathcal{I}_0. \tag{A.28}$$

Taking the weighted sum of Equation (A.27) proves the claim. The same recursion also guarantees the required finite second moments of the parameters. □

Lemma A.3 (Auxiliary-objective geometry and backbone heterogeneity)**.** *Define*

$$\Gamma = H_\star - \sum_k p_k f_k^\star, \quad s_p^\star = \sum_k p_k s_k^\star, \quad \mathcal{H}_p = \sum_k p_k\|s_k^\star - s_p^\star\|^2. \tag{A.29}$$

Then H is μ-strongly convex in the weighted norm on z, and

$$0 \leq \Gamma \leq \frac{\mathcal{C}_s}{2\mu}, \qquad \mathcal{H}_p \leq \frac{2\Gamma}{\mu}, \qquad H(z) - H_\star \geq \frac{\mu}{2}\|z - z_H^\star\|_p^2. \tag{A.30}$$

The first upper bound remains valid with $\widehat{\mathcal{C}}_s$ in place of $\mathcal{C}_s$.

Proof. Applying strong convexity to each f_k and weighting by p_k gives

$$\begin{aligned}H(z') &\geq H(z) + \langle \nabla H(z), z' - z\rangle \\ &\quad + \frac{\mu}{2}\left[\|s' - s\|^2 + \sum_k p_k\|e_k' - e_k\|^2\right]. \end{aligned} \tag{A.31}$$

This proves weighted strong convexity and the last inequality in Equation (A.30), and it also guarantees a unique minimizer of H. Since $H(z) \geq \sum_k p_k f_k^\star$ for every z, we have $\Gamma \geq 0$.

Let $u_k^{\mathrm{ref}} = (s^{\mathrm{ref}}, e_k^{\mathrm{ref}})$. Conditional optimality gives $\nabla_e f_k(u_k^{\mathrm{ref}}) = 0$. Lemma A.1 implies

$$\begin{aligned}\Gamma &\leq \sum_k p_k[f_k(u_k^{\mathrm{ref}}) - f_k^\star] \\ &\leq \frac{1}{2\mu}\sum_k p_k\|\nabla f_k(u_k^{\mathrm{ref}})\|^2 = \frac{\mathcal{C}_s}{2\mu}. \end{aligned} \tag{A.32}$$

Retaining the full gradient norm for any $\widehat{e}_k$ gives the alternative bound.

Applying strong convexity of each local objective at $z_H^\star = (s_H^\star, e_{H,1}^\star, \dots, e_{H,E}^\star)$ yields

$$\begin{aligned}\Gamma &\geq \frac{\mu}{2}\sum_k p_k \left[\|s_H^\star - s_k^\star\|^2 + \|e_{H,k}^\star - e_k^\star\|^2\right] \\ &\geq \frac{\mu}{2}\sum_k p_k \|s_H^\star - s_k^\star\|^2 \geq \frac{\mu}{2}\mathcal{H}_p. \end{aligned} \tag{A.33}$$

For any backbone s_{test}, the last step uses the identity

$$\sum_k p_k \|s_{\text{test}} - s_k^\star\|^2 = \|s_{\text{test}} - s_p^\star\|^2 + \mathcal{H}_p. \tag{A.34}$$

□

Lemma A.4 (Shared-backbone averaging and expert concatenation)**.** *Let* $z_T = (\bar{s}_T, e_{1,T}, \dots, e_{E,T})$. *Then*

$$\mathbb{E}[H(z_T) - H_\star] \leq A_T, \qquad \mathbb{E}\|z_T - z_H^\star\|_p^2 \leq \frac{2A_T}{\mu}. \tag{A.35}$$

Proof. Because $\nabla f_k(u_k^\star) = 0$, smoothness gives

$$f_k(\bar{s}_T, e_{k,T}) - f_k^\star \leq \frac{L}{2}\left[\|\bar{s}_T - s_k^\star\|^2 + \|e_{k,T} - e_k^\star\|^2\right]. \tag{A.36}$$

The squared-distance decomposition and Jensen's inequality give

$$\begin{aligned}\sum_k p_k \mathbb{E}\|\bar{s}_T - s_k^\star\|^2 &= \mathcal{H}_p + \mathbb{E}\|\bar{s}_T - s_p^\star\|^2 \\ &= \mathcal{H}_p + \mathbb{E}\left\|\sum_k p_k (s_{k,T} - s_k^\star)\right\|^2 \\ &\leq \mathcal{H}_p + \sum_k p_k \mathbb{E}\|s_{k,T} - s_k^\star\|^2. \end{aligned} \tag{A.37}$$

Adding the expert errors and applying Lemma A.2 yields

$$\mathbb{E}H(z_T) - \sum_k p_k f_k^\star \leq \frac{L}{2}(\mathcal{H}_p + R_T). \tag{A.38}$$

Subtracting Γ and applying Lemma A.3 gives

$$\begin{aligned}\mathbb{E}[H(z_T) - H_\star] &\leq \frac{L}{2}R_T + \frac{L}{2}\mathcal{H}_p - \Gamma \\ &\leq \frac{L}{2}R_T + (\kappa - 1)\Gamma \\ &\leq \frac{L}{2}R_T + \frac{\kappa - 1}{2\mu}\mathcal{C}_s = A_T. \end{aligned} \tag{A.39}$$

Weighted strong convexity of H gives the parameter-distance bound. The expert errors are not averaged, and the proof does not use independence among the local training streams. □

B Proofs of the Main-Text Results

B.1 Proof of Theorem 3.1

Let

$$P = \operatorname{diag}(I_s, p_1 I_{e_1}, \dots, p_E I_{e_E}, I_\phi). \tag{B.1}$$

Taking the trial displacement $d = -L_F^{-1} P^{-1} \nabla F(\Theta)$ in Equation (A.5) gives

$$F\left(\Theta - \frac{1}{L_F} P^{-1} \nabla F(\Theta)\right) \le F(\Theta) - \frac{1}{2L_F} \|\nabla F(\Theta)\|_{p,*}^2. \tag{B.2}$$

Since the left-hand side is at least $F_\star$,

$$\|\nabla F(\Theta)\|_{p,*}^2 \le 2L_F [F(\Theta) - F_\star]. \tag{B.3}$$

This is a trial point used only in the proof and does not alter the ordinary SGD update used by the algorithm.

Define $\Theta^{\mathrm{ref}} = (z_H^\star, \phi_0)$ and $\Delta = F(\Theta^{\mathrm{ref}}) - F_\star$. Because $\nabla H(z_H^\star) = 0$, Assumption A.3 and Jensen's inequality imply

$$\begin{aligned} \|\nabla F(\Theta^{\mathrm{ref}})\|^2 &= \left\| \mathbb{E}_\xi [G_F(z_H^\star, \phi_0; \xi) - \widetilde{G}_H(z_H^\star; \xi)] \right\|^2 \\ &\le \mathcal{R}. \end{aligned} \tag{B.4}$$

Euclidean strong convexity of the full objective and Lemma A.1 give

$$\Delta \le \frac{\|\nabla F(\Theta^{\mathrm{ref}})\|^2}{2\mu_F} \le D_0. \tag{B.5}$$

The merged point and the reference point have the same router initialization, so Lemma A.4 gives

$$\mathbb{E}\|\Theta_{D,0} - \Theta^{\mathrm{ref}}\|_p^2 \le \frac{2A_T}{\mu}. \tag{B.6}$$

Applying weighted smoothness at the reference point and then the dual-norm Cauchy–Schwarz inequality yields

$$\begin{aligned} \mathbb{E}[F(\Theta_{D,0}) - F_\star] &\le \Delta + \|\nabla F(\Theta^{\mathrm{ref}})\|_{p,*} \sqrt{\mathbb{E}\|\Theta_{D,0} - \Theta^{\mathrm{ref}}\|_p^2} \\ &\quad + \frac{L_F}{2} \mathbb{E}\|\Theta_{D,0} - \Theta^{\mathrm{ref}}\|_p^2 \\ &\le \Delta + \sqrt{2L_F \Delta} \sqrt{\frac{2A_T}{\mu}} + \frac{L_F A_T}{\mu} \\ &= \left(\sqrt{\Delta} + \sqrt{\frac{L_F}{\mu} A_T} \right)^2 \le B_T. \end{aligned} \tag{B.7}$$

This proves Equation (3.4). Equation (B.3) implies $\underline{\mathcal{E}}_0 \le \mathcal{E}_0$. Therefore, when $B_T < \underline{\mathcal{E}}_0$,

$$F(\Theta^{\mathrm{init}}) - \mathbb{E}F(\Theta_{D,0}) = \mathcal{E}_0 - \mathbb{E}[F(\Theta_{D,0}) - F_\star] \ge \underline{\mathcal{E}}_0 - B_T > 0. \tag{B.8}$$

Lemma A.3 already establishes the alternative upper bound based on $\widehat{\mathcal{C}}_s$; the remaining steps are unchanged. □

B.2 Finite-Step Sufficient Condition

Equations (A.13) and (A.15) give

$$A_T = A_\infty + \frac{L}{2} q^T \left(\mathcal{I}_0 - \frac{\eta \mathcal{W}}{\mu} \right) \le A_\infty + \frac{L}{2} q^T \mathcal{I}_0. \tag{B.9}$$

Define the positive statistical margin

$$\delta = \frac{\mu}{L_F} \left(\sqrt{\underline{\mathcal{E}}_0} - \sqrt{D_0} \right)^2 - A_\infty. \tag{B.10}$$

If $(L/2) q^T \mathcal{I}_0 < \delta$, then

$$A_T < \frac{\mu}{L_F} (\sqrt{\underline{\mathcal{E}}_0} - \sqrt{D_0})^2. \tag{B.11}$$

Since $D_0 < \underline{\mathcal{E}}_0$, taking nonnegative square roots gives

$$\sqrt{D_0} + \sqrt{\frac{L_F}{\mu} A_T} < \sqrt{\underline{\mathcal{E}}_0}. \tag{B.12}$$

Squaring proves $B_T < \underline{\mathcal{E}}_0$, so Theorem 3.1 applies. Because $q \in (0,1)$, $q^T \mathcal{I}_0 \to 0$, and hence a finite T exists. When $\mathcal{I}_0 > 0$, it is sufficient to choose

$$T > \max\left\{0, \frac{\log(L\mathcal{I}_0/(2\delta))}{-\log q}\right\}. \tag{B.13}$$

When $\mathcal{I}_0 = 0$, the transient term is identically zero. □

B.3 Proof of Theorem 3.2

Fix $a \in \{D, B\}$ and write $\Theta_j = \Theta_{a,j}$ for this proof. Each training process uses the filtration generated by the full pretraining history and its current parameters; $\mathbb{E}_j$ denotes expectation conditional on $\mathcal{G}_{a,j}$. Since $\|d\|_p \le \|d\|$, the full objective satisfies the Euclidean descent inequality

$$F(\Theta - \alpha G) \le F(\Theta) - \alpha \langle \nabla F(\Theta), G \rangle + \frac{L_F \alpha^2}{2} \|G\|^2. \tag{B.14}$$

Conditional unbiasedness and the variance bound give

$$\begin{aligned} \mathbb{E}_j F(\Theta_{j+1}) &\le F(\Theta_j) - \alpha \left(1 - \frac{L_F \alpha}{2}\right) \|\nabla F(\Theta_j)\|^2 + \frac{L_F \alpha^2}{2} \sigma_F^2 \\ &\le F(\Theta_j) - \frac{\alpha}{2} \|\nabla F(\Theta_j)\|^2 + \frac{L_F \alpha^2}{2} \sigma_F^2. \end{aligned} \tag{B.15}$$

Euclidean strong convexity implies $\|\nabla F(\Theta_j)\|^2 \ge 2\mu_F [F(\Theta_j) - F_\star]$. With $\mathcal{U}_j = \mathbb{E}[F(\Theta_j) - F_\star]$, we obtain

$$\mathcal{U}_{j+1} \le \rho \mathcal{U}_j + \frac{L_F \alpha^2}{2} \sigma_F^2. \tag{B.16}$$

Unrolling the recursion gives

$$\mathcal{U}_S \le \rho^S \mathcal{U}_0 + \frac{L_F \alpha^2 \sigma_F^2}{2} \frac{1 - \rho^S}{1 - \rho} = \nu + \rho^S (\mathcal{U}_0 - \nu). \tag{B.17}$$

Using $\mathcal{U}_0 \le B_T$ for MoE-DisCo and $\mathcal{U}_0 = \mathcal{E}_0$ for the baseline proves Equation (A.17).

For a target $\varepsilon > \nu$ and an initial-error upper bound $\mathcal{B} \ge 0$, define the sufficient iteration count

$$\overline{N}(\mathcal{B}, \varepsilon) = \begin{cases} 0, & \mathcal{B} \le \varepsilon, \\ \left\lceil \dfrac{\log((\mathcal{B} - \nu)/(\varepsilon - \nu))}{-\log \rho} \right\rceil, & \mathcal{B} > \varepsilon. \end{cases} \tag{B.18}$$

If $\mathcal{B} \le \varepsilon$, the initial point already meets the target according to its upper bound, so zero steps suffice. If $\mathcal{B} > \varepsilon > \nu$, solving

$$\nu + \rho^S (\mathcal{B} - \nu) \le \varepsilon \tag{B.19}$$

gives $S \ge \log((\mathcal{B} - \nu)/(\varepsilon - \nu))/(-\log \rho)$, proving Equation (B.18). This sufficient count is nondecreasing in $\mathcal{B}$. Since $B_T < \underline{\mathcal{E}}_0 \le \mathcal{E}_0$,

$$\overline{N}(B_T, \varepsilon) \le \overline{N}(\mathcal{E}_0, \varepsilon). \tag{B.20}$$

When $\mathcal{E}_0 > B_T > \varepsilon > \nu$, the difference between the unrounded counts is

$$\frac{\log((\mathcal{E}_0 - \nu)/(B_T - \nu))}{-\log \rho} > 0. \tag{B.21}$$

If this difference is at least one, the rounded counts also differ by at least one. □

C Partition Statistics and Representation Quality

Here $h(\xi)$ is the mean-pooled sequence representation defined in Section 3.2. Let

$$\begin{aligned} \bar{h}_k &= \mathbb{E}[h(\xi) \mid C = k], \\ W_{h,k} &= \mathbb{E}[\|h(\xi) - \bar{h}_k\|^2 \mid C = k], \\ W_h &= \sum_k p_k W_{h,k}. \end{aligned} \tag{C.1}$$

W_h is the distortion of the fixed partition when conditional means are used as centroids. The realized distortion for any other centroids upper-bounds this value.

Proposition C.1 (Representation distortion controls within-cluster gradient noise)**.** *Assume that local SGD samples independently with replacement within each cluster. For every relevant local parameter vector u, suppose*

$$\nabla_u \ell(u; \xi) = \psi_u(h(\xi)) + r_u(\xi), \tag{C.2}$$

where ψ_u is L_g-Lipschitz and

$$\mathbb{E}[\|r_u(\xi)\|^2 \mid C = k] \le \epsilon_k^2 \tag{C.3}$$

uniformly on the relevant parameter region. For single-sample SGD, one may take

$$w_k = (L_g \sqrt{W_{h,k}} + \epsilon_k)^2. \tag{C.4}$$

With $\epsilon_g^2 = \sum_k p_k \epsilon_k^2$,

$$\mathcal{W} \le (L_g \sqrt{W_h} + \epsilon_g)^2. \tag{C.5}$$

Consequently, when $D_0 < \underline{\mathcal{E}}_0$, finite-step local pretraining yields a strictly improved initialization under the sufficient condition

$$\frac{L\eta}{2\mu}(L_g \sqrt{W_h} + \epsilon_g)^2 + \frac{\kappa - 1}{2\mu}\mathcal{C}_s < \frac{\mu}{L_F}(\sqrt{\underline{\mathcal{E}}_0} - \sqrt{D_0})^2. \tag{C.6}$$

C.1 Proof of Proposition C.1

Fix a cluster k and a relevant parameter vector u, and let $g = \nabla_u \ell(u; \xi)$. The conditional mean minimizes mean squared deviation, so

$$\mathbb{E}[\|g - \mathbb{E}[g \mid C = k]\|^2 \mid C = k] \le \mathbb{E}[\|g - \psi_u(\bar{h}_k)\|^2 \mid C = k]. \tag{C.7}$$

Using Equation (C.2) and the conditional L^2 triangle inequality,

$$\begin{aligned} &\big(\mathbb{E}[\|g - \psi_u(\bar{h}_k)\|^2 \mid C = k]\big)^{1/2} \\ &\quad \le \big(\mathbb{E}[\|\psi_u(h(\xi)) - \psi_u(\bar{h}_k)\|^2 \mid C = k]\big)^{1/2} + \big(\mathbb{E}[\|r_u(\xi)\|^2 \mid C = k]\big)^{1/2} \\ &\quad \le L_g \sqrt{W_{h,k}} + \epsilon_k. \end{aligned} \tag{C.8}$$

Thus, w_k can be chosen as in Equation (C.4). Cauchy–Schwarz further gives

$$\begin{aligned} \sum_k p_k w_k &= L_g^2 W_h + 2L_g \sum_k p_k \sqrt{W_{h,k}} \epsilon_k + \epsilon_g^2 \\ &\le L_g^2 W_h + 2L_g \sqrt{W_h} \epsilon_g + \epsilon_g^2 \\ &= (L_g \sqrt{W_h} + \epsilon_g)^2. \end{aligned} \tag{C.9}$$

Substituting this upper bound into A_∞ shows that Equation (C.6) implies Equation (A.16). For an independent with-replacement minibatch of size b_k, the conditional variance of the average gradient is further divided by b_k; the single-sample bound above remains valid but may be loose. □

C.2 Within–Between Decomposition of Expert Gradients

At a common single-expert reference model u, define

$$g_e(\xi; u) = \nabla_e \ell(u; \xi), \quad m_{e,k}(u) = \mathbb{E}[g_e(\xi; u) \mid C = k], \quad \bar{m}_e(u) = \sum_k p_k m_{e,k}(u). \tag{C.10}$$

The corresponding within- and between-cluster dispersions are

$$\begin{aligned} W_e(\mathcal{P}; u) &= \sum_k p_k \mathbb{E}[\|g_e - m_{e,k}\|^2 \mid C = k], \\ B_e(\mathcal{P}; u) &= \sum_k p_k \|m_{e,k} - \bar{m}_e\|^2. \end{aligned} \tag{C.11}$$

Proposition C.2 (Between-cluster gradient differences and specialization potential)**.** *When the second moment of the gradient is finite,*

$$\mathrm{Var}(g_e(\xi; u)) = W_e(\mathcal{P}; u) + B_e(\mathcal{P}; u). \tag{C.12}$$

Further, fix a backbone s, let $J_k(e; s) = f_k(s, e)$, and assume that each $J_k(\cdot; s)$ is μ_e-strongly convex and L_e-smooth in e. Define

$$\begin{aligned} e^\star_{\mathrm{pool}|s} &= \arg\min_e \sum_k p_k J_k(e; s), \\ e^\star_{k|s} &= \arg\min_e J_k(e; s), \\ \mathcal{S} &= \sum_k p_k [J_k(e^\star_{\mathrm{pool}|s}; s) - J_k(e^\star_{k|s}; s)]. \end{aligned} \tag{C.13}$$

Then the between-cluster gradient dispersion $B^\star_e$ at the common-expert optimum satisfies

$$\frac{B^\star_e}{2L_e} \le \mathcal{S} \le \frac{B^\star_e}{2\mu_e}. \tag{C.14}$$

Proof. Expand the square after writing $g_e - \bar{m}_e = (g_e - m_{e,C}) + (m_{e,C} - \bar{m}_e)$. The first parenthesized term has zero conditional mean given the cluster label, so the expected cross term vanishes, proving Equation (C.12). Moreover,

$$\sum_k p_k \nabla_e J_k(e^\star_{\mathrm{pool}|s}; s) = 0, \tag{C.15}$$

and therefore $B^\star_e = \sum_k p_k \|\nabla_e J_k(e^\star_{\mathrm{pool}|s}; s)\|^2$. Applying the two objective-gap inequalities in Lemma A.1 to every $J_k(\cdot; s)$ and taking their weighted sum proves Equation (C.14). □

Equation (C.14) characterizes the optimal objective improvement from replacing a common expert by cluster-specific experts. It is not a comparison between the training speed of MoE-DisCo and end-to-end MoE training, because end-to-end training can also learn specialization. Without control along the optimization trajectory, a dispersion decomposition at a single reference point cannot replace Assumption A.2.

D Arbitrary Fixed Aggregation Weights

Suppose that the shared backbone is instead merged as $\bar{s}^\gamma_T = \sum_k \gamma_k s_{k,T}$, where $\gamma_k \ge 0$ and $\sum_k \gamma_k = 1$. The weights may be determined by a fixed data partition but do not depend on the random local-SGD trajectories. Define

$$d_\gamma = \chi^2(\gamma \| p) = \sum_k \frac{(\gamma_k - p_k)^2}{p_k}, \qquad c_\gamma = \sum_k \frac{\gamma_k^2}{p_k} = 1 + d_\gamma. \tag{D.1}$$

Proposition D.1 (Initialization bound under weight mismatch)**.** *Let*

$$A_T^\gamma = \frac{L}{2}\left(\frac{\sqrt{d_\gamma \mathcal{C}_s}}{\mu} + \sqrt{(1+d_\gamma)R_T}\right)^2 + \frac{\kappa-1}{2\mu}\mathcal{C}_s,$$
$$B_T^\gamma = \left(\sqrt{D_0} + \sqrt{\frac{L_F}{\mu}A_T^\gamma}\right)^2. \tag{D.2}$$

Theorems 3.1 and 3.2 remain valid for this merging rule when B_T is replaced by B_T^γ. Setting $\gamma = p$ exactly recovers the main-text bound.

Proof. Because $\sum_k(\gamma_k - p_k) = 0$, define

$$w_\gamma = \sum_k(\gamma_k - p_k)s_k^\star = \sum_k(\gamma_k - p_k)(s_k^\star - s_p^\star). \tag{D.3}$$

Weighted Cauchy–Schwarz and Lemma A.3 give

$$\|w_\gamma\|^2 \le d_\gamma \mathcal{H}_p \le \frac{2d_\gamma\Gamma}{\mu} \le \frac{d_\gamma \mathcal{C}_s}{\mu^2}. \tag{D.4}$$

Let $\delta s_k = s_{k,T} - s_k^\star$ and $\delta e_k = e_{k,T} - e_k^\star$, and construct

$$X_T = \left(\sum_k \gamma_k \delta s_k, \sqrt{p_1}\delta e_1, \ldots, \sqrt{p_E}\delta e_E\right). \tag{D.5}$$

Since $c_\gamma \ge 1$, weighted Cauchy–Schwarz yields

$$\begin{aligned}\mathbb{E}\|X_T\|^2 &\le c_\gamma \sum_k p_k \mathbb{E}\|\delta s_k\|^2 + \sum_k p_k \mathbb{E}\|\delta e_k\|^2 \\ &\le c_\gamma R_T.\end{aligned} \tag{D.6}$$

The L^2 triangle inequality therefore gives

$$\mathbb{E}\|(w_\gamma, 0, \ldots, 0) + X_T\|^2 \le \left(\frac{\sqrt{d_\gamma \mathcal{C}_s}}{\mu} + \sqrt{c_\gamma R_T}\right)^2. \tag{D.7}$$

Decomposing the squared backbone distance into $\mathcal{H}_p$ and the squared distance from $s_p^\star$ gives

$$\begin{aligned}&\sum_k p_k \mathbb{E}[\|\bar{s}_T^\gamma - s_k^\star\|^2 + \|e_{k,T} - e_k^\star\|^2] \\ &\qquad \le \mathcal{H}_p + \left(\frac{\sqrt{d_\gamma \mathcal{C}_s}}{\mu} + \sqrt{c_\gamma R_T}\right)^2.\end{aligned} \tag{D.8}$$

Repeating the smoothness step of Lemma A.4, subtracting Γ, and then using $\mathcal{H}_p \le 2\Gamma/\mu$ and $\Gamma \le \mathcal{C}_s/(2\mu)$ yields $\mathbb{E}[H(z_T^\gamma) - H_\star] \le A_T^\gamma$. The remaining steps in the proofs of Theorems 3.1 and 3.2 are unchanged. □

This extension covers uniform merging for imbalanced clusters. In that case, uniform weights cannot simply be identified with p_k, but the analysis also does not need a separate unknown offset for the backbone center.

E Optional Bias–Variance Refinement

This appendix is used only to make the variance reduction from shared-backbone averaging explicit; it is not required by the main theorems. We additionally assume that every f_k is twice continuously differentiable with an L_{Hess}-Lipschitz Hessian. To average independent replicas, we also assume

that the local random sampling streams are mutually independent conditional on the fixed data, partition, and initialization.

Define

$$r_{k,0} = \|u_{k,0} - u_k^\star\|, \quad V_{k,t} = \mathbb{E}\|u_{k,t} - \mathbb{E}u_{k,t}\|^2, \quad b_{k,t} = \mathbb{E}u_{k,t} - u_k^\star. \tag{E.1}$$

Let

$$\begin{aligned} V_{k,\infty}^{\mathrm{ub}} &= \frac{\eta^2 w_k}{1-q^2} = \frac{\eta w_k}{\mu(2-\eta\mu)}, \\ \widehat{V}_{k,T} &= V_{k,\infty}^{\mathrm{ub}}(1-q^{2T}), \\ \beta_k(T) &= q^T r_{k,0} + \frac{L_{\mathrm{Hess}} V_{k,\infty}^{\mathrm{ub}}}{2\mu}(1-q^T). \end{aligned} \tag{E.2}$$

Proposition E.1 (Finite-step variance and mean bias)**.** *Under these additional conditions, $V_{k,T} \le \widehat{V}_{k,T}$ and $\|b_{k,T}\| \le \beta_k(T)$. Let $V_{s,k,T}$ and $V_{e,k,T}$ denote the variance of the shared and expert blocks, respectively, and define*

$$\mathcal{Q}_T = \sum_k p_k \beta_k(T)^2 + \sum_k p_k^2 V_{s,k,T} + \sum_k p_k V_{e,k,T}. \tag{E.3}$$

Then A_T in the main text can be replaced by

$$A_T^\dagger = \min\left\{ A_T, \frac{L}{2}\mathcal{Q}_T + \frac{\kappa-1}{2\mu}\mathcal{C}_s \right\}, \tag{E.4}$$

and the two main theorems hold with the corresponding $B_T^\dagger = (\sqrt{D_0} + \sqrt{L_F A_T^\dagger/\mu})^2$.

Proof. Fix k. The deterministic gradient map $D_k(u) = u - \eta\nabla f_k(u)$ is q-Lipschitz. Indeed, $\mu I \preceq \nabla^2 f_k \preceq LI$ and $\eta \le 1/L$, so integration along the line segment gives

$$\|D_k(u) - D_k(v)\| \le q\|u - v\|. \tag{E.5}$$

The law of total variance and conditional unbiasedness imply

$$\begin{aligned} V_{k,t+1} &= \mathrm{Var}(D_k(u_{k,t})) + \eta^2 \mathbb{E}\|g_{k,t} - \nabla f_k(u_{k,t})\|^2 \\ &\le q^2 V_{k,t} + \eta^2 w_k. \end{aligned} \tag{E.6}$$

Here, $\mathrm{Var}(D_k(U)) \le \mathbb{E}\|D_k(U) - D_k(\mathbb{E}U)\|^2 \le q^2\,\mathrm{Var}(U)$. The fixed initialization gives $V_{k,0} = 0$. Unrolling the recursion proves $V_{k,T} \le \widehat{V}_{k,T}$, with no nondecaying initial-distance term.

A Taylor expansion of the gradient at $\mathbb{E}u_{k,t}$ and the Hessian-Lipschitz condition give

$$\|\mathbb{E}\nabla f_k(u_{k,t}) - \nabla f_k(\mathbb{E}u_{k,t})\| \le \frac{L_{\mathrm{Hess}}}{2} V_{k,t}. \tag{E.7}$$

The mean recursion therefore satisfies

$$\|b_{k,t+1}\| \le q\|b_{k,t}\| + \frac{\eta L_{\mathrm{Hess}}}{2} V_{k,t} \le q\|b_{k,t}\| + \frac{\eta L_{\mathrm{Hess}}}{2} V_{k,\infty}^{\mathrm{ub}}. \tag{E.8}$$

Unrolling it and using $1 - q = \eta\mu$ proves $\|b_{k,T}\| \le \beta_k(T)$.

Let $b_{s,k} = \mathbb{E}s_{k,T} - s_k^\star$ and $b_{e,k} = \mathbb{E}e_{k,T} - e_k^\star$. Independence of local training gives the exact identity

$$\mathbb{E}\|\bar{s}_T - s_p^\star\|^2 = \left\| \sum_k p_k b_{s,k} \right\|^2 + \sum_k p_k^2 V_{s,k,T}. \tag{E.9}$$

Jensen's inequality gives

$$\left\| \sum_k p_k b_{s,k} \right\|^2 + \sum_k p_k \|b_{e,k}\|^2 \le \sum_k p_k \|b_{k,T}\|^2 \le \sum_k p_k \beta_k(T)^2. \tag{E.10}$$

Thus, the parameter-error sum in Lemma A.4 can be controlled by $\mathcal{H}_p + \mathcal{Q}_T$. Repeating the remaining steps yields Equation (E.4). Both the basic and refined bounds hold, so their minimum can be used. □

In particular, if $p_k = 1/E$, $w_k \le w$, and $V_\infty^{\mathrm{ub}} = \eta w/[\mu(2-\eta\mu)]$, then

$$\limsup_{T\to\infty} \mathbb{E}\|\bar{s}_T - s_p^\star\|^2 \le \left(\frac{L_{\mathrm{Hess}} V_\infty^{\mathrm{ub}}}{2\mu}\right)^2 + \frac{V_\infty^{\mathrm{ub}}}{E}. \tag{E.11}$$

Thus, with the other constants fixed, the bound for the shared block is $O(\eta^2 + \eta/E)$. The expert blocks are concatenated, so their variances remain weighted by p_k, rather than p_k^2. This result does not imply a uniform $1/E$ improvement in the optimization error of the entire MoE.

F Decaying Step Sizes and Nonempty Conditions

F.1 Full-Objective Convergence with Decaying Step Sizes

Proposition F.1 (Vanishing final-iterate error bound)**.** *Fix $a \in \{D, B\}$ and abbreviate $\Theta_j = \Theta_{a,j}$. Retain the assumptions in the main text and replace the second-stage step size by*

$$\alpha_j = \frac{2}{\mu_F(j + j_0)}, \qquad j_0 \ge \max\{3, 2L_F/\mu_F\}. \tag{F.1}$$

If the expected objective error at the start of the second stage is at most $\mathcal{B}$, then

$$\mathbb{E}[F(\Theta_S) - F_\star] \le \frac{(j_0-2)(j_0-1)\mathcal{B} + (2L_F\sigma_F^2/\mu_F^2)S}{(S + j_0 - 2)(S + j_0 - 1)}. \tag{F.2}$$

MoE-DisCo and the baseline may use $\mathcal{B} = B_T$ and $\mathcal{B} = \mathcal{E}_0$, respectively.

Proof. The chosen step sizes are at most $1/L_F$, so the variable-step recursion used in the proof of Theorem 3.2 remains valid. Let $\mathcal{U}_j = \mathbb{E}[F(\Theta_j) - F_\star]$ and $c = 2L_F\sigma_F^2/\mu_F^2$. Then

$$\mathcal{U}_{j+1} \le \frac{j + j_0 - 2}{j + j_0}\mathcal{U}_j + \frac{c}{(j + j_0)^2}. \tag{F.3}$$

Multiplying by $(j + j_0 - 1)(j + j_0)$ gives

$$\begin{aligned}
&(j + j_0 - 1)(j + j_0)\mathcal{U}_{j+1} \\
&\quad \le (j + j_0 - 2)(j + j_0 - 1)\mathcal{U}_j + c\frac{j + j_0 - 1}{j + j_0} \\
&\quad \le (j + j_0 - 2)(j + j_0 - 1)\mathcal{U}_j + c.
\end{aligned} \tag{F.4}$$

Summing from $j = 0$ to $S - 1$ makes the successive terms telescope. Using $\mathcal{U}_0 \le \mathcal{B}$ proves Equation (F.2). □

For a fixed first stage, this bound is $O(\mathcal{B}/S^2) + O(\sigma_F^2/S)$ and vanishes asymptotically. Reducing $\mathcal{B}$ improves the initial-error term but does not change the common stochastic-error order.

F.2 The Statistical Conditions Are Nonempty

The following example checks only the compatibility of the conditions; it is not a task model for the main theorem. Let every cluster have the deterministic local loss

$$f_k(s, e) = \tfrac{1}{2}\|s\|^2 + \tfrac{1}{2}\|e - a_k\|^2, \qquad F(z, \phi) = H(z) + \tfrac{1}{2}\|\phi\|^2, \tag{F.5}$$

where at least one $a_k \neq 0$. Set $s_0 = e_{k,0} = \phi_0 = 0$ and $s^{\mathrm{ref}} = 0$. Use exact gradients as zero-noise SGD, and let the auxiliary and full sample losses differ only by $\|\phi\|^2/2$. Then

$$\mathcal{W} = \mathcal{C}_s = \mathcal{R} = 0, \quad L = \mu = L_F = 1, \quad \mu_F = p_{\min}, \quad \underline{\mathcal{E}}_0 = \mathcal{E}_0 = \tfrac{1}{2}\sum_k p_k\|a_k\|^2 > 0. \tag{F.6}$$

Here $p_{\min} = \min_{1\le k\le E} p_k$. The statistical margin is therefore strictly positive. We have $B_T = (1-\eta)^T\mathcal{E}_0 < \mathcal{E}_0$ for every $T \ge 1$. The actual merging error is $(1-\eta)^{2T}\mathcal{E}_0$, consistent with the upper bound. Representations that are constant within clusters and distinct across clusters can make the K-means distortion zero. This example shows that the conditions are logically satisfiable; it does not assign the same properties to an arbitrary real MoE.

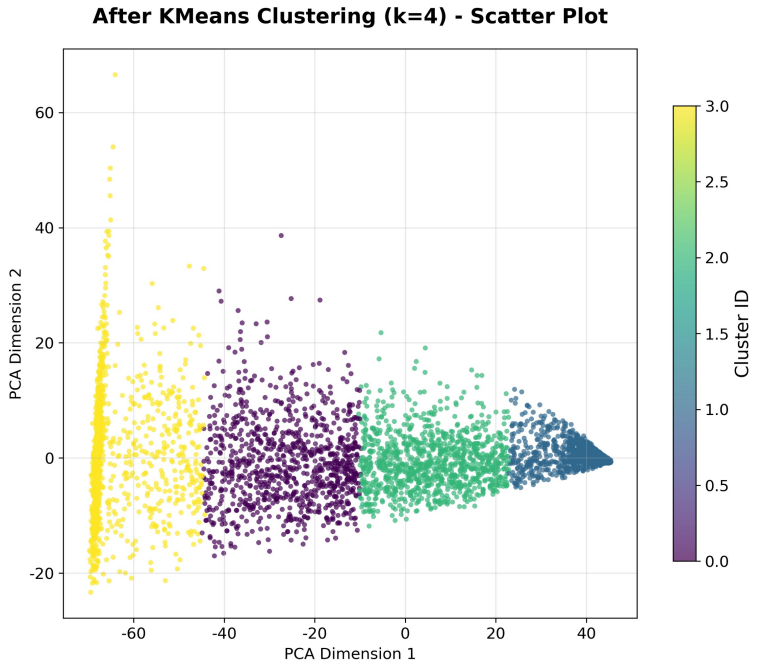


Figure 7: C4 sentence representations assigned to four semantic clusters and projected to two dimensions with PCA.

G More on Dataset Partitioning

MoE-DisCo uses unsupervised clustering to allocate semantically similar subsets of the training data to individual expert submodels. For a sentence $x = (x_1, \ldots, x_{N_x})$, we reuse the embedding table of the corresponding public checkpoint and compute the sentence representation $h(x)$ by mean pooling:

$$h(x) = \frac{1}{N_x} \sum_{i=1}^{N_x} \text{Embedding}(x_i) \tag{G.1}$$

where N_x is the number of tokens in x. K-means groups these vectors into E nonempty clusters, one per expert.

The clustering objective minimizes the within-cluster sum of squared distances to the respective centroids:

$$\min_{\mathcal{P}=\{\mathcal{D}_k\}_{k=1}^{E}} \sum_{k=1}^{E} \sum_{x \in \mathcal{D}_k} \|h(x) - \bar{h}_k\|^2, \tag{G.2}$$

Here $\mathcal{P}$ is a partition into E nonempty subsets $\mathcal{D}_k$, $n_k = |\mathcal{D}_k|$, and $\bar{h}_k = n_k^{-1} \sum_{x \in \mathcal{D}_k} h(x)$ is the mean centroid, matching $\mathbb{E}[h(\xi) \mid C = k]$ under uniform empirical sampling. Finally, the resulting clusters are used to partition the original dataset into E semantically coherent sub-datasets, each of which is assigned to train a distinct single-expert submodel.

The resulting clusters train distinct expert submodels. Figure 7 visualizes their geometry. We measure cluster purity as the average cosine similarity between each sample embedding and its assigned centroid. Table 5 reports the cluster sizes and purity values. Because the sample counts are imbalanced, the integration stage uses the same counts as weights for shared-backbone averaging.

H Hyperparameter Settings

All experiments in this work are conducted with the following hyperparameters :

Table 5: C4 partition statistics. Purity is the average cosine similarity between sample embeddings and their assigned centroid.

Cluster	Samples	Purity
1	56,233	99.24%
2	131,578	88.73%
3	59,018	99.83%
4	57,171	96.43%
Total / clustering time	**304,000**	**1.32 min**

Table 6: Hyperparameter settings used in submodel training.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
Learning rate scheduler	constant
Batch size	8
Data Type	`bfloat16`
Sequence length	1024
Number of GPUs	1

I SUBMODEL TRAINING DETAILS

In MoE-DisCo, each submodel is trained on a low-cost single-GPU setups. For the k-th submodel, we retain the full shared backbone and keep only the k-th expert in every MoE layer. This results in a standard dense model with the same depth and hidden dimensionality as the original MoE. Although sub-expert training requires a non-trivial amount of time, it incurs relatively low monetary cost, as each submodel can be trained independently on affordable hardware and fully in parallel. A detailed breakdown of training steps and cost is provided in Table 8.

Table 8: Maximum training steps and wall-clock time among all sub-expert models. As sub-expert models are trained fully in parallel, we report the maximum value across experts, which determines the overall training time of this stage.

Model	Dataset	Max Steps	Max Time (h)
Qwen	C4	4,200	2.42
	WikiText	4,330	0.97
	OpenWebText	11,845	12.69
Llama	C4	4,810	1.71
	WikiText	5,136	1.66
	OpenWebText	8,670	6.92

Table 7: Hyperparameter settings used in full-parameter training and MoE-DisCo fine-tune.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	0.01
Warmup ratio	0.03
Learning rate scheduler	Cosine
Global Batch size	16
Data Type	`bfloat16`
Sequence length	1024
Tensor parallel size (TP)	2
Data parallel size (DP)	2
Number of GPUs	4

J GATING NETWORK ARCHITECTURE

The MoE architecture employed in this work adopts top-K_{act} sparsely-gated routing. For each input token, the gating network selects K_{act} experts from the E available experts, where $1 \leq K_{\mathrm{act}} \leq E$, thereby maintaining high model capacity with controlled computational cost. During the submodel training phase, only a single expert is assigned per submodel; consequently, the gating network is not utilized, and each submodel is trained as a dense model with a fixed expert. In the subsequent fine-tune stage, the full gating network is introduced and requires only a small number of optimization steps to learn effective and coordinated routing across experts, achieving efficient and stable adaptation without extensive retraining.