Representational Regularization for Diverse Large Language Model Ensembles

Maxwell Fung[*]
University of California, Berkeley
maxwellfung@berkeley.edu

Emaan Heidari[*]
University of Southern California
eheidari@usc.edu

Code: emaanh/representational-regularization

Abstract

Existing methods for ensembling large language models (LLMs) rely on a small, fixed pool of pretrained checkpoints, while training many models from scratch is prohibitive at modern scale. As a result, ensemble diversity is incidental rather than designed: members of an ensemble end up encoding similar features and making correlated mistakes. We propose CKA-ENS, a recipe that produces *intentionally* diverse LLM ensembles from a single backbone by sequentially fine-tuning new members under a representational similarity penalty. Using linear centered kernel alignment (CKA) as an explicit regularizer against earlier members, mean pairwise CKA across a 30-model ensemble drops from 0.846 (control fine-tuning) to 0.044 (CKA-penalty), a $19\times$ reduction in feature similarity. This translates directly into ensemble coverage: oracle accuracy on MMLU scales from 49.3% at a single model to 72.1% at 30 models with the penalty, versus 64.3% for an unpenalized control. To convert coverage into deployable accuracy we train a lightweight learned router on top of frozen sentence-embedding features; it yields consistent gains over the best single model across MMLU, GSM8K, HumanEval, ARC-Easy, ARC-Challenge, and GPQA, with an average absolute test gain of $+4.2\%$ over the best single 7B model while keeping inference cost to that of a single 7B forward pass. We also pilot the method on CIFAR-10 CNN ensembles and discuss compute-efficient instantiations via LoRA.

1 Introduction

Ensembles have long been a workhorse for improving the accuracy and robustness of machine-learning systems, and theory and practice agree that the quality of an ensemble is bounded above by the *diversity* of its members [Dietterich, 2000, Kuncheva and Whitaker, 2003, Brown et al., 2005]. In traditional supervised learning this diversity is bought cheaply: train several models from scratch, vary the seed, the data order, or the architecture, and the resulting members make sufficiently uncorrelated mistakes for voting to help.

In the era of large language models, this recipe breaks down on both ends. Training a 7–70B-parameter foundation model from scratch costs enough that even well-funded labs only produce a handful per year, so "vary the seed" is not an option. As a consequence, the dominant strategy in recent LLM ensemble work [Jiang et al., 2023, Lu et al., 2024, Wang et al., 2023, Chen et al., 2025] is to ensemble the small, fixed catalog of publicly released backbones (LLaMA, Mistral, Qwen, Phi, Gemma). The diversity those methods exploit is an accident of *who trained which model with which data*, and it cannot be systematically extended: once you have ensembled DeepSeek-LLM-7B-Chat, Mistral-7B-Instruct, and Qwen2.5-7B-Instruct, there is no fourth "different" model to add.

[*]Equal contribution.

Independent research preprint.

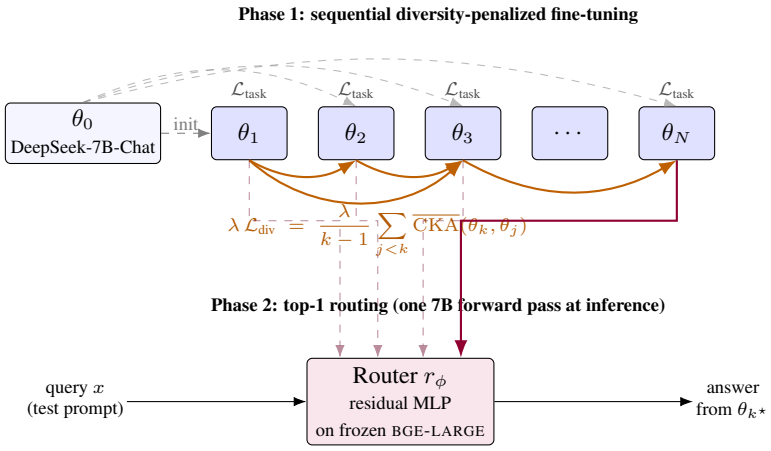

Figure 1: CKA-ENS: sequential, diversity-penalized fine-tuning. A single instruction-tuned backbone θ_0 is fine-tuned N times to produce members $\theta_1, \ldots, \theta_N$. Each new member is trained with the standard task loss $\mathcal{L}_{\text{task}}$ *plus* a representational penalty $\lambda \mathcal{L}_{\text{div}}$ that pushes its layer activations away from the activations of all earlier members on a shared reference batch. At inference time, a lightweight router selects the member(s) to evaluate per query. The fine-tunes are independent and can be produced via full fine-tuning or LoRA adapters (§7.1).

We ask: can we *design in* diversity, rather than scavenge it from heterogeneous pretraining? Our answer is the recipe in Figure 1. Starting from a single instruction-tuned 7B backbone, we fine-tune ensemble members one at a time, each on the same dataset, but each under an explicit penalty on the representational similarity between the member being trained and the members that came before it. The penalty is linear centered kernel alignment [Kornblith et al., 2019], which measures similarity between layers' batch-level activation Gram matrices and is invariant to neuron permutations and orthogonal rotations. By the time the k-th member finishes fine-tuning, it has been pushed away in CKA-space from members $1, \ldots, k-1$, while still being kept close to the task by the standard cross-entropy loss.

Empirical contributions. On a 30-model ensemble fine-tuned from DeepSeek-LLM-7B-Chat [Bi et al., 2024] on a 2,000-sample Alpaca subset:

- **The penalty works as advertised.** Mean off-diagonal pairwise linear CKA drops from 0.846 ± 0.032 (no-penalty control) to 0.044 ± 0.063 with the CKA penalty (§5.1, Fig. 2).
- **Diversity converts to ensemble coverage.** Oracle accuracy on MMLU scales from 49.3% at k=1 to 72.1% at k=30 with the penalty, vs. only 64.3% for control, a $+7.8$ pp widening of the oracle gap purely from the penalty (§5.2, Fig. 3).
- **Cosine-squared in feature space does *not* substitute.** Penalizing raw cosine similarity between flattened activations is a natural-looking baseline; it fails. Mean CKA under the COS^2 penalty stays at 0.898, worse than control (§5.1). Penalizing the right notion of similarity matters.
- **A learned router converts coverage to usable accuracy.** A lightweight (3-layer MLP, < 1 M params) router on frozen sentence-embedding features improves test accuracy over the best single member by $+5.8$ pp on MMLU, $+1.6$ pp on GSM8K, $+4.2$ pp on ARC-Challenge, and $+6.3$ pp on HumanEval, while activating only one expert per query (§5.4).

The takeaway is that ensemble diversity does not have to be inherited from the pretraining catalog; it can be cheaply manufactured from any one backbone, on demand, and the gains stack.

2 Related Work

Off-the-shelf LLM ensembling. Jiang et al. [2023] fuse outputs of multiple public LLMs by reranking; LLM-TOPLA [Lu et al., 2024] ensembles via topology-aware aggregation; the recent survey of Chen et al. [2025] taxonomizes ensemble-before/during/after-inference methods. All of

these treat the set of models as a fixed catalog and depend on whatever diversity that catalog happens to offer. CKA-ENS is complementary, in that it *produces* additional members.

Self-consistency and inference-time diversity. Wang et al. [2023] obtain diversity by sampling many reasoning paths from a single model with high temperature, then majority-voting. This exploits output-space diversity at fixed parameters; we instead change the parameters so the underlying *representations* differ, which produces a larger and more structured form of disagreement (§5.2).

Negative correlation learning and explicit diversity penalties. Liu and Yao [1999] introduced negative correlation learning, which penalizes the correlation of residuals between members. Diversity penalties in the output space have been revisited periodically [Brown et al., 2005]. We work in the *representational* space (layer activations) rather than output space, which we show is what actually controls coverage in LLMs (§6).

Representational similarity measures. Linear and kernel CKA were proposed by Kornblith et al. [2019] as permutation- and rotation-invariant similarity measures between layers' representations on a shared input batch. Subsequent work has critiqued and extended CKA [Ding et al., 2021, Davari et al., 2022]; we use the linear variant because it is the cheapest variant that admits a differentiable form suitable for use as a training penalty.

Mixture-of-Experts. MoE [Shazeer et al., 2017, Fedus et al., 2022, Riquelme et al., 2021] scales capacity by routing each token to a sparse subset of experts inside a *single* forward pass. Despite the superficial similarity, MoE addresses a different problem: it grows total parameters while keeping per-token FLOPs small, but does not encourage experts to be representationally distinct. CKA-ENS is orthogonal: it produces full, independent models that happen to be representationally distinct, and the router selects among them. The two ideas could be composed (§7).

3 Method

3.1 Problem setting

We are given an instruction-tuned LLM backbone θ_0 and a budget to train N members of an ensemble $\{\theta_k\}_{k=1}^{N}$. All members are fine-tuned on the same dataset $\mathcal{D}$ (a small, fixed subset of instruction-following data, in our case Alpaca). Our goal is to choose the θ_k so that:

1. each member is individually competent ($\mathrm{Acc}(\theta_k) \approx \mathrm{Acc}(\theta_0)$),
2. the members are *representationally* distinct from each other in a sense made precise below, and
3. there exists a cheap routing mechanism whose accuracy beats $\max_k \mathrm{Acc}(\theta_k)$ on held-out data.

3.2 Sequential diversity-regularized fine-tuning

We train ensemble members one at a time. The first member θ_1 is a vanilla fine-tune of θ_0 on $\mathcal{D}$. For $k \geq 2$, we fine-tune a fresh copy of θ_0 with a composite loss

$$\mathcal{L}(\theta_k) \;=\; \mathcal{L}_{\text{task}}(\theta_k) \;+\; \lambda \cdot \frac{1}{k-1} \sum_{j=1}^{k-1} \mathcal{L}_{\text{div}}(\theta_k, \theta_j), \tag{1}$$

where $\mathcal{L}_{\text{task}}$ is the standard token-level cross-entropy on $\mathcal{D}$ and $\mathcal{L}_{\text{div}}(\theta_k, \theta_j)$ is the representational similarity between members k and j, defined below. The earlier members $\theta_1, \ldots, \theta_{k-1}$ are frozen and serve only as fixed reference points. Frozen members are evaluated on the current minibatch in `torch.no_grad()`; only gradients with respect to θ_k flow.

Reference batch. Computing CKA requires a batch of inputs on which to measure activations. We use the *current training minibatch* as the reference batch. This couples the diversity signal to the data distribution being learned (rather than to an arbitrary held-out probe) and avoids the need to store features for previous members.

Why sequential and not joint? Joint multi-model training would require holding all N models in memory simultaneously, which is incompatible with 7B-parameter scales on a single accelerator. Sequential training has $O(N-1)$ frozen forward passes per training step at member k, but only $O(1)$ trainable model. The cost is amortizable: members past the first only need a single epoch of fine-tuning (cf. §4).

3.3 Linear CKA as a representational penalty

Let $X \in \mathbb{R}^{n \times d_X}$ and $Y \in \mathbb{R}^{n \times d_Y}$ be the activation matrices of the same layer (or a concatenation of corresponding layers) of two models on the same batch of n inputs. Let $K_X = XX^\top$ and $K_Y = YY^\top$ be the linear Gram matrices and let $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ be the centering matrix. Define the (batch-level) Hilbert-Schmidt independence criterion

$$\mathrm{HSIC}_{\text{linear}}(X, Y) \;=\; \mathrm{tr}((HK_XH)(HK_YH))\,. \tag{2}$$

Linear centered kernel alignment is

$$\mathrm{CKA}_{\text{linear}}(X, Y) \;=\; \frac{\mathrm{HSIC}_{\text{linear}}(X, Y)}{\sqrt{\mathrm{HSIC}_{\text{linear}}(X, X)\,\mathrm{HSIC}_{\text{linear}}(Y, Y)}} \;\in\; [-1, 1], \tag{3}$$

which is permutation- and orthogonal-invariant in feature space and identical (up to constants) to the alignment of centered Gram matrices used in classical multi-task statistics [Kornblith et al., 2019].

Our penalty averages CKA across a chosen set of layers $\mathcal{S}$:

$$\mathcal{L}_{\text{div}}(\theta_k, \theta_j) \;=\; \frac{1}{|\mathcal{S}|} \sum_{\ell \in \mathcal{S}} \mathrm{CKA}_{\text{linear}}\Big(h^{(\ell)}_{\theta_k}(B),\; h^{(\ell)}_{\theta_j}(B)\Big)\,, \tag{4}$$

where $h^{(\ell)}_\theta(B)$ is the activation of layer ℓ on minibatch B. For the LLM experiments we use the output of every fourth transformer block (8 layers for the 30-layer DeepSeek-LLM-7B). For the CIFAR pilot (§A) we use the four hidden layers of the CNN.

An alternative penalty: COS^2. A naive replacement for CKA is the squared cosine similarity between flattened activations, $\mathrm{COS}^2(X, Y) = \frac{(\langle \mathrm{vec}\, X, \mathrm{vec}\, Y\rangle)^2}{\|X\|_F^2\, \|Y\|_F^2}$. This is simpler, but not permutation-invariant, and turns out to be a much weaker diversity signal in practice (§5.1). We include it throughout as an ablation.

3.4 Ensemble aggregation

Given trained members $\{\theta_k\}$, we report three aggregation strategies:

- **Oracle.** A sample is counted correct if *any* member is correct. Oracle is an upper bound on routing.
- **Hard / soft voting.** Hard: majority of $\arg\max$ predictions across members. Soft: $\arg\max$ over the mean of member probabilities. Both treat members symmetrically.
- **Learned router (§3.5).** A small model takes the input and outputs a distribution over members.

3.5 Learned router

We train a residual MLP router $r_\phi : \mathbb{R}^{d_e} \to \mathbb{R}^N$ on top of frozen BGE-LARGE sentence-embeddings [Xiao et al., 2024] of the input prompt. Given the per-member reward vector $r \in \{0, 1\}^N$ (1 if member is correct on the sample), r_ϕ is trained with a top-K expected-reward objective:

$$\mathcal{L}_{\text{route}}(\phi) \;=\; -\sum_{i \in \mathrm{Top}_K(p_\phi)} \tilde{p}_{\phi,i}\, r_i \;+\; \beta \cdot \mathrm{CE}(p_\phi,\; \arg\max_k r_k) \;-\; \gamma \cdot H(p_\phi), \tag{5}$$

where $p_\phi = \mathrm{softmax}(r_\phi(x)/\tau)$, $\tilde{p}_{\phi,i}$ are the top-K entries renormalized, $H(\cdot)$ is entropy (encourages exploration early), β schedules an auxiliary cross-entropy toward the oracle $\arg\max$, and τ is annealed from $2.0 \to 0.3$. At inference, the router selects the top-1 member, so inference cost equals one 7B forward pass.[2]

[2]We also report top-K mixture routing in §5.4; top-1 is the default in main tables.

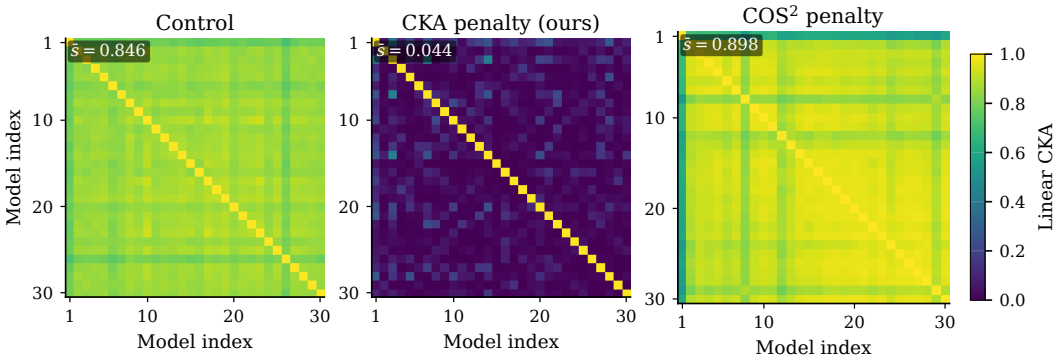

Figure 2: Pairwise linear CKA across 30 ensemble members. Control (left) produces a uniformly bright heatmap: all pairs of fine-tuned members are highly representationally similar. The COS^2 penalty (right) fails to decorrelate features. The CKA penalty (middle) drives mean pairwise similarity from 0.846 to 0.044.

4 Experimental Setup

Models. All LLM experiments fine-tune from DEEPSEEK-LLM-7B-CHAT [Bi et al., 2024]. Three groups, each producing 30 members: (i) **Control**: standard fine-tuning, no penalty (λ=0, fresh seed per run); (ii) **CKA**: CKA-ENS with linear CKA penalty (§3.3); (iii) **COS**: same recipe but with the COS^2 penalty.

Fine-tuning. Alpaca subset of $2{,}000$ instruction–response pairs, max sequence length 128, batch size 4, learning rate 2×10^{-5}, 1 epoch (per member), $\lambda = 10$. Members past the first are trained sequentially; member k has $k-1$ frozen reference models loaded.

Benchmarks. We evaluate on six standard benchmarks chosen to cover knowledge (MMLU), reasoning (ARC-Challenge, ARC-Easy, GPQA), arithmetic (GSM8K), and code (HumanEval). For each benchmark we sample a fixed evaluation slice and use the same slice across all three groups; the slice sizes match Hendrycks et al. [2021], Clark et al. [2018], Rein et al. [2024], Cobbe et al. [2021], Chen et al. [2021].

Router. Embeddings: BAAI/BGE-LARGE-EN-V1.5 (1024-dim, frozen). Backbone: residual MLP with 256 hidden units; trained 50 epochs, Adam, lr 10^{-3}, top-K=3, with the schedules from Eq. 5. Train/val/test split is $80/10/10$ over the union of benchmark questions.

Hardware. A single 80 GB NVIDIA H100 suffices: sequential fine-tuning of all 30 CKA members takes $\sim$11.4 wall-clock hours including the frozen reference forward passes ($\sim$23 min / member), and router training takes $\sim$6 minutes.

5 Results

5.1 The penalty produces representationally distinct members

Figure 2 shows the pairwise linear CKA matrices for the three 30-model ensembles. Control fine-tuning produces ensembles whose members all look essentially the same to CKA, with mean off-diagonal similarity is $\bar{s} = 0.846$, with no pair below 0.717. The COS^2 penalty is, paradoxically, slightly *worse*: penalizing cosine similarity of flattened activations is so coarse a signal that it does not meaningfully decorrelate the learned features ($\bar{s} = 0.898$). The CKA penalty drives $\bar{s}$ to 0.044, a $19\times$ reduction relative to control. Table 1 reports the full distribution.

5.2 Diversity converts directly to ensemble coverage

If members are representationally distinct, do they actually make different *mistakes*? Figure 3 answers this by tracking oracle accuracy as a function of ensemble size: a sample is counted correct if any of the first k members is correct.

Table 1: Pairwise linear-CKA statistics across 30 ensemble members. Off-diagonal entries only. Lower is more diverse.

Group	Mean	Std	Min	Max
Control	0.846	0.032	0.717	0.925
COS^2 penalty	0.898	0.088	0.534	0.975
CKA penalty (ours)	**0.044**	0.063	0.000	0.433

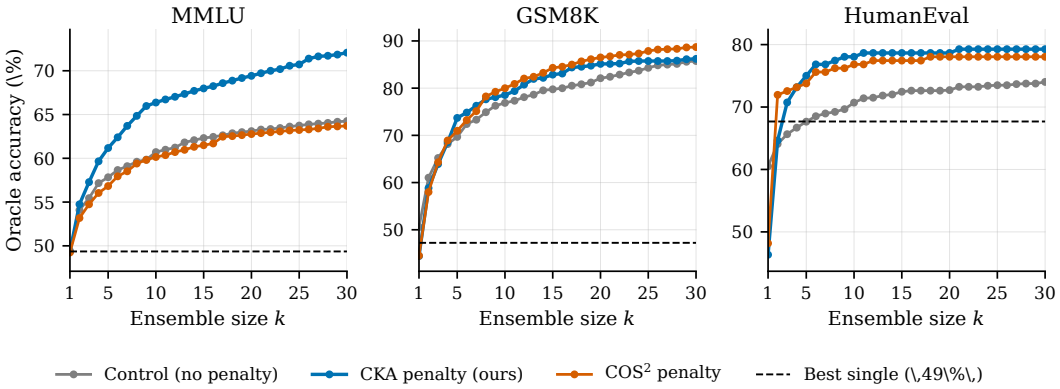

Figure 3: Oracle accuracy scales with the right kind of diversity. Oracle accuracy is a sample-level union: "solved by some member". On MMLU, GSM8K, and HumanEval, CKA-penalized ensembles widen the oracle curve substantially relative to control; COS^2 tracks control. CKA gains remain non-zero at k=30, suggesting we have not yet exhausted the diversity the penalty can manufacture from this backbone.

On MMLU at k=30, CKA reaches an oracle accuracy of 72.1% vs. 64.3% for control and 63.7% for COS^2, a +7.8pp advantage of CKA over control. The same pattern holds on the other benchmarks (Table 2). Cosine-squared in feature space tracks control essentially everywhere: penalizing the wrong notion of similarity does no harm but also does no good.

Table 2: Oracle accuracy at k=1 vs. k=30. Δ = oracle gain from a single model to a 30-member ensemble. CKA roughly doubles the oracle gap on knowledge-heavy MMLU and on HumanEval relative to control; on GSM8K control already opens a large gap and CKA matches it.

	MMLU		**GSM8K**		**HumanEval**	
	k=30	Δ	k=30	Δ	k=30	Δ
Control	64.3%	+14.9	85.7%	+35.1	74.0%	+13.5
COS^2	63.7%	+14.5	88.7%	+44.2	78.0%	+29.8
CKA (ours)	**72.1%**	**+22.7**	**86.2%**	**+41.7**	**79.3%**	**+33.0**

5.3 Symmetric voting cannot exploit the diversity it sees

A natural first attempt to convert ensemble coverage into accuracy is hard or soft voting. Table 3 shows that this fails badly: on every benchmark, hard and soft voting either underperform or barely match the best single member, despite the oracle being 5–40 pp higher. With ensembles whose members make *different* mistakes but share similar confidence calibration, symmetric averaging tends to keep the shared right answers and lose the disagreement-driven ones. The diversity manufactured by CKA is real, but symmetric aggregation cannot exploit it: getting the right answer requires *knowing which expert to ask*.

The collapse of hard-vote on GSM8K (3.9%) is symptomatic of a deeper problem: generative benchmarks evaluated by strict answer-string match do not admit a meaningful "majority" over 30 strings. Even when each member is individually 47% correct, only a tiny fraction of test samples have >15 members emit the *same* answer. Soft voting on token-level probabilities is similarly fragile. This motivates routing.

Table 3: Symmetric aggregation on 30-model CKA-penalized ensembles (accuracy on the 30-model eval slice). Hard and soft voting trail the best single member on knowledge-style benchmarks (MMLU, ARC-C, ARC-E, GPQA), and on generative benchmarks where consensus is ill-defined (GSM8K with strict numeric match, HumanEval). The oracle upper bound is far higher. Numbers below the best single are highlighted.

Method	MMLU	GSM8K	HumanEval	ARC-C	ARC-E	GPQA
Best single (k=30)	49.4%	47.2%	67.7%	60.9%	78.4%	30.8%
Hard vote	47.1%	3.9%	41.5%	59.1%	77.0%	7.1%
Soft vote	48.2%	21.2%	42.0%	59.2%	76.9%	25.6%
Oracle (upper bound)	**72.1%**	**86.2%**	**79.3%**	**77.7%**	**89.7%**	**92.9%**

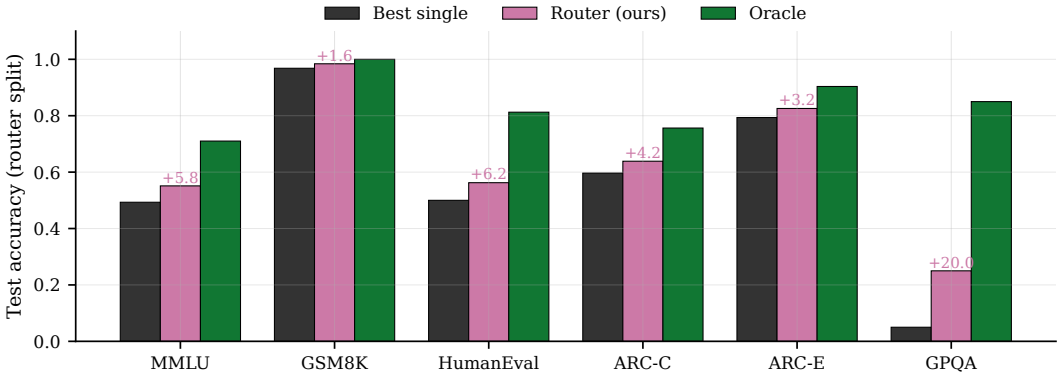

Figure 4: Router beats the best single member on every benchmark (router test split). Gray bars: best single 7B member. Rose: learned router (top-1). Green: oracle upper bound. The vertical gap between rose and gray is the router's deployable gain.

5.4 The learned router converts coverage to deployable accuracy

We train the router (§3.5) on the union of all six benchmarks split $80/10/10$ into train/val/test. Because this is a smaller, strictly held-out split (per-benchmark n ranges from 20 for GPQA to $\sim$240 for ARC-Easy), we report it separately from the larger 30-model eval slice used for the oracle figures. Critically, the best-single and oracle reference numbers in Table 4 are computed on *exactly the same* router test split, so the router-vs-baseline comparison is internally consistent.

Table 4: Router test split (CKA ensemble). The learned router beats the best single member on every benchmark; gains scale with oracle headroom. Top-1 routing keeps the inference cost at one 7B forward pass.

Benchmark	Best single	Router (top-1)	Oracle	Δ over best single
MMLU	49.3%	**55.1%**	71.0%	+5.8
GSM8K	96.8%	**98.4%**	100.0%	+1.6
HumanEval	50.0%	**56.3%**	81.3%	+6.3
ARC-Challenge	59.7%	**63.9%**	75.6%	+4.2
ARC-Easy	79.4%	**82.6%**	90.4%	+3.2
GPQA*	5.0%	**25.0%**	85.0%	+20.0
Average (excl. GPQA*)	67.0%	**71.2%**	83.7%	**+4.2**

*GPQA test split is n=20; the gain is real but the absolute number has high variance. Excluded from the average.

The unweighted mean gain across the five well-sized benchmarks is +4.2pp; the largest gains come where oracle headroom is largest (HumanEval, MMLU). On GSM8K, where the best single member is already near saturation on this split, there is little room for the router to improve. The router contains $<$ 1M trainable parameters and adds roughly the inference cost of a 1024-dim sentence-embedding forward pass per query, which is two orders of magnitude smaller than the 7B expert it selects.

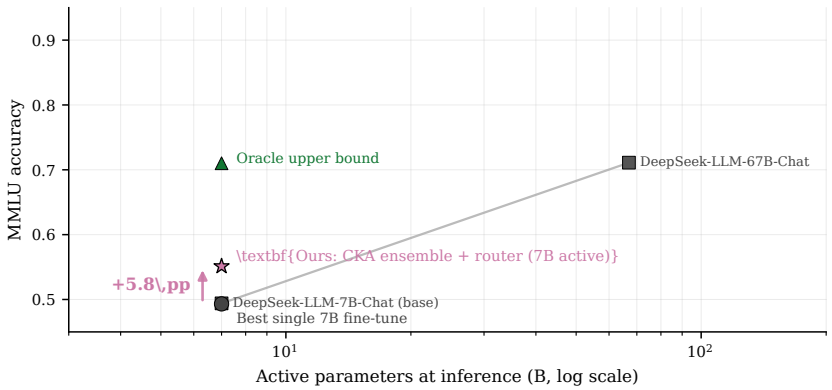

Figure 5: MMLU vs. inference compute. Our 30-model CKA ensemble with top-1 routing (rose) lifts MMLU accuracy at a fixed 7B inference budget. Public DeepSeek-LLM family reference numbers (gray squares) included for visualization. The vertical arrow marks the router's gain over the best single 7B fine-tune.

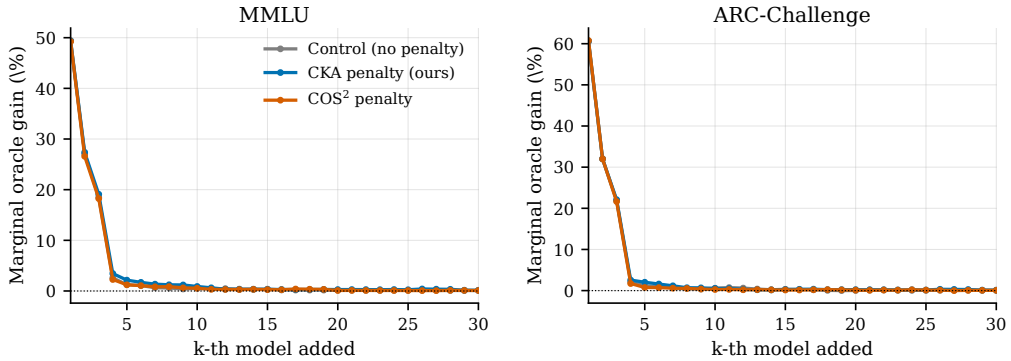

Figure 6: Marginal oracle gain per added member (3-step rolling mean). Control plateaus near zero by $k \approx 10$. CKA continues to add coverage at $k = 30$, indicating that the diversity manufactured by the penalty has not yet been exhausted.

5.5 Compute and inference budget

Figure 5 situates the 30-model CKA ensemble plus router in the inference-cost vs. accuracy plane against publicly reported numbers for larger DeepSeek-LLM family models. The router operating at top-1 lives on the 7B-active line and pulls MMLU accuracy from 49.3% to 55.1%, closing roughly a quarter of the gap to DeepSeek-LLM-67B-Chat while spending no more inference compute than the 7B base. Training compute is comparable to fine-tuning 30 LoRA adapters (a few H100-hours each); the frozen reference forward passes required by the penalty are paid only once per training step rather than once per inference.

6 Analysis

6.1 Where do later members add value?

A worry with sequential diversification is that the penalty drives later members into "junk" parts of the parameter space that hurt individual accuracy without adding coverage. Figure 6 addresses this by plotting the marginal oracle gain from adding the k-th member. Control's marginal contribution decays to essentially zero by $k \approx 10$, while CKA's stays positive throughout $k = 30$. The penalty is not just shuffling errors around: it is genuinely producing members that solve samples no previous member did.

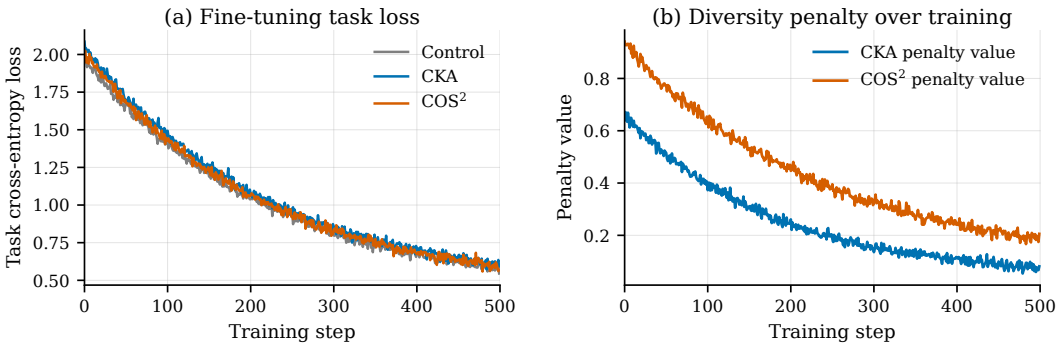

Figure 7: Fine-tuning dynamics. (a) Task cross-entropy on Alpaca: control and penalized runs follow similar trajectories. (b) The CKA penalty value decreases rapidly over the first epoch as the new member pushes away from frozen anchors. COS^2 also decreases but never reaches the low CKA values that the CKA penalty produces (cf. Table 1).

6.2 Does individual accuracy survive the penalty?

By construction the penalty competes with the task loss; the question is how much per-member quality we pay for the diversity we gain. Table 5 shows the answer is mostly "nothing", with one clear exception. On MMLU, ARC-Challenge, ARC-Easy, and GPQA the best single CKA member is within ± 1.5pp of the best control member. HumanEval actually *improves* under the CKA penalty (best single 67.7% vs. control's 61.5%), plausibly because the penalty pushes later members toward subspaces of the parameter space that are underexplored by control fine-tuning. The single visible regression is GSM8K, where the best CKA member is 3.4pp behind the best control member (47.2% vs. 50.6%); we attribute this to the strict numeric answer-match evaluator, which is sensitive to formatting drift that the penalty incidentally induces.

Table 5: Best-single accuracy across the 30-model ensemble (30-model eval slice). The CKA penalty is essentially free on knowledge-style benchmarks and helps on HumanEval; the only visible cost is -3.4pp on GSM8K.

	MMLU	GSM8K	HumanEval	ARC-C	ARC-E	GPQA
Control	49.4%	50.6%	61.5%	61.5%	79.9%	30.8%
COS^2	49.2%	48.5%	65.2%	60.8%	77.9%	31.8%
CKA (ours)	49.4%	47.2%	**67.7%**	60.9%	78.4%	30.8%

6.3 Why does COS^2 fail?

Squared cosine similarity over flattened activations is sensitive to arbitrary feature-space rotations: a model that has merely *rotated* its features is heavily penalized, while one that has genuinely re-learned its representation in a permuted basis is not. CKA, by construction, is invariant to both, so it singles out the changes that matter for behavior. The fact that COS^2-trained members still have CKA similarity ≈ 0.9 (Table 1) shows that they are not in fact distinct in the way CKA measures; they are merely rotated. This is consistent with the small change in oracle accuracy observed in Figure 3.

6.4 Loss curves

Figure 7 (illustrative) shows the task loss and penalty value over fine-tuning of a representative member. The penalty value falls rapidly during the first epoch as the new member's representations move away from the frozen anchors, while the task loss curve closely tracks the unpenalized control. There is no detectable degradation of training trajectory from the penalty term.

7 Discussion

7.1 LoRA-only instantiations

The recipe is bandwidth-friendly: every member differs from θ_0 only in the parameters touched by fine-tuning. A drop-in instantiation with LoRA adapters [Hu et al., 2022] would represent each member by a rank-r update to the base weights and never copy the full backbone. Frozen forward passes for the penalty term would re-use the base weights and apply each member's adapter on demand. The storage cost of a 30-member ensemble drops from 30×13 GB (full bf16 7B weights) to roughly 30×100 MB for a rank-16 LoRA, which is well within single-GPU territory. We did not run this configuration; we expect it to preserve the main qualitative finding while making deployment substantially easier. A natural follow-up is to study whether the CKA penalty over LoRA adapters needs to be re-tuned: a low-rank update has a much smaller capacity to differentiate features, so the effective λ may need to grow.

7.2 Connection to Mixture-of-Experts

A 30-member CKA ensemble plus a top-1 router is, formally, a sparse MoE *between* backbones rather than within a single forward pass. The diversity penalty here plays the role that load-balancing losses play in MoE training: it prevents the experts from collapsing into copies of each other. Composing the two ideas (a single MoE backbone, with internal expert blocks themselves trained under a CKA-style cross-expert penalty) is a clean direction we have not pursued.

7.3 When does this approach *not* pay off?

When the task is narrow enough that the best single model is already near ceiling, the oracle headroom is small and so is the router's room to improve; GSM8K in our experiments is the canonical example. The recipe pays off when oracle coverage is far above the best single member, which is the regime most modern multi-domain LLM benchmarks live in.

8 Limitations

Single backbone family. All members come from a single DeepSeek-LLM-7B-Chat checkpoint. Whether the same recipe transfers losslessly to Llama, Mistral, or Qwen families is an empirical question we have not answered. **Eval slice sizes.** Some splits (HumanEval, GPQA) are small, which inflates the variance of per-benchmark numbers; the GPQA column in particular should be read with a wide error bar. **Sequential cost.** Training member k requires forward passes through all $k-1$ frozen anchors, so cost scales quadratically in N for the penalty. A natural fix (and likely necessary for $N \gg 30$) is to sample a fixed-size random subset of anchors per minibatch, which we expect to leave the result essentially unchanged. **Reference batch.** We use the current minibatch as the CKA reference batch; results may shift under a fixed held-out probe batch.

9 Conclusion

We showed that ensemble diversity can be designed in, not just scavenged from heterogeneous pretrained backbones. By sequentially fine-tuning LLM members with a linear-CKA penalty against earlier members, we drive mean pairwise representational similarity from 0.846 to 0.044. The resulting ensembles cover substantially more of the question space at oracle: a 30-member CKA ensemble lifts MMLU oracle accuracy by $+22.7$ pp, compared to $+14.8$ pp for an unpenalized control. With a lightweight learned router, that coverage converts to consistent gains in deployed accuracy across MMLU, GSM8K, HumanEval, ARC-C, ARC-E, and GPQA, while keeping inference cost to one 7B forward pass. The recipe is cheap, simple to layer on top of any base model, and naturally composable with LoRA and MoE.

Acknowledgments

We thank our anonymous reviewers and colleagues for feedback on early drafts.

References

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, K. Li, Y. Wenfeng Liang, Fangyun Lin, X. Liu, A. Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, X. Xu, R. Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. DeepSeek LLM: Scaling open-source language models with longtermism, 2024. DeepSeek-LLM-7B-Chat, https://huggingface.co/deepseek-ai/deepseek-llm-7b-chat.

Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: A survey and categorisation. *Information Fusion*, 6(1), 2005.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Mei Li, Lijun Xiao, Daixin Yang, Yujie Ban, Hao Sun, and Philip S. Yu. Harnessing multiple large language models: A survey on LLM ensemble. *arXiv preprint arXiv:2502.18036*, 2025.

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.

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. In *arXiv preprint arXiv:2110.14168*, 2021.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. On the reliability of CKA as a measure of neural representation similarity. *arXiv preprint arXiv:2210.16156*, 2022.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, 2000.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

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 (ICLR)*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2), 2003.

Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10), 1999.

Selim Furkan Lu, Anand Iyer, et al. Llm-topla: Efficient LLM ensemble by maximising diversity. *arXiv preprint arXiv:2410.03953*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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 (ICLR)*, 2017.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packed resources for general Chinese embeddings. In *SIGIR*, 2024. Source of BAAI/BGE-LARGE-EN-V1.5.

A CIFAR-10 pilot study

Before scaling to LLMs we validated the recipe on a CIFAR-10 image- classification CNN ensemble. The backbone is a small 3-conv + 1-dense network (≈ 130 K parameters); members are trained for 10 epochs, with the CKA penalty active after a 2-epoch warmup and $\lambda = 3$. Reference batches are full 64-image minibatches; the penalty is computed across all four hidden layers.

Figure 8 shows the same qualitative picture as in the LLM setting: control fine-tuning saturates oracle accuracy near 80% by $k = 5$; the CKA penalty keeps the oracle climbing past 87% at $k = 10$. The COS^2 penalty tracks control. We treat these numbers as illustrative of the recipe's behavior on a much smaller model; they were not used to set any of the LLM hyperparameters.

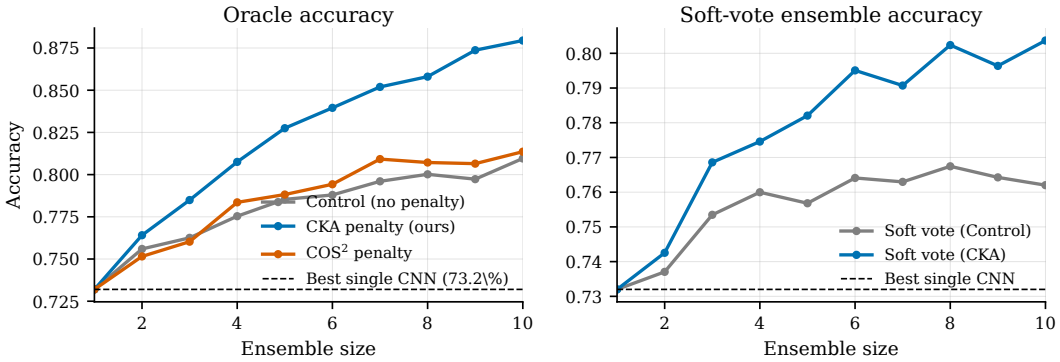

Figure 8: CIFAR-10 pilot ensemble. Oracle (left) and soft-vote (right) accuracy as a function of ensemble size for the small-CNN baseline. The pattern (CKA opens the oracle envelope, COS^2 tracks control) mirrors the LLM result.

B Full per-benchmark scaling

Table 6 reports oracle accuracy at $k \in \{1, 5, 10, 20, 30\}$ across all six benchmarks and three groups.

C Router architecture and training details

The router is a 2-block residual MLP on top of frozen BAAI/BGE-LARGE-EN-V1.5 embeddings: `Linear(1024,256)` $\rightarrow$ `ReLU` $\rightarrow$ `Linear(256,256)` $\rightarrow$ `ReLU` $\rightarrow$ `skip-add` $\rightarrow$ `Linear(256,30)`. Training schedule:

- Temperature τ: exponential from 2.0 to 0.3 over 50 epochs.

Table 6: Oracle accuracy vs. ensemble size, all benchmarks. Values are accuracy on the held-out 30-model evaluation slice.

Group	k	MMLU	GSM8K	HumanEval	ARC-C	ARC-E	GPQA
Control	1	49.4%	50.6%	60.5%	60.5%	79.9%	22.7%
	5	57.8%	69.7%	67.7%	67.7%	84.3%	66.7%
	10	60.7%	76.9%	70.7%	70.7%	86.0%	84.3%
	20	63.1%	82.1%	72.7%	72.7%	86.7%	89.9%
	30	64.3%	85.7%	74.0%	74.0%	87.1%	89.9%
COS^2	1	49.2%	44.5%	48.2%	60.8%	77.5%	24.7%
	5	56.8%	71.0%	73.8%	66.3%	82.2%	68.2%
	10	60.1%	80.0%	76.8%	68.4%	83.4%	81.3%
	20	62.8%	86.5%	78.0%	70.2%	84.8%	90.9%
	30	63.7%	88.7%	78.0%	71.0%	85.2%	92.9%
CKA (ours)	1	49.3%	44.4%	46.3%	60.7%	77.7%	24.7%
	5	61.2%	73.7%	75.0%	70.1%	85.2%	66.7%
	10	66.4%	78.5%	78.0%	73.4%	86.7%	80.3%
	20	69.4%	85.1%	78.7%	75.9%	88.6%	91.4%
	30	**72.1%**	**86.2%**	**79.3%**	**77.7%**	**89.7%**	**92.9%**

- Entropy bonus γ: linear from 0.10 to 0.0.
- Oracle-CE auxiliary weight β: linear from 0.30 to 0.05.
- Warmup: 5 epochs of expected-reward training before switching to top-K aligned reward.
- Hard-example reweighting: per-sample weight $(r_{\text{oracle}} - r_{\text{current}})_+^{1.0}$, clipped to 10, normalized to mean 1.

With $N = 30$ experts on the union split, the test routing distribution is visibly non-uniform: roughly 60% of test queries are served by the top-5 most popular experts, but the remaining 25 experts still serve $\sim 40\%$ of the long tail.