DENSE CREDIT ASSIGNMENT FOR ADAPTIVE MOE ROUTING VIA LAYER-WISE MOE-δ

Anonymous authors
Paper under double-blind review

ABSTRACT

MoE LLMs are widely used to scale capacity under limited inference compute. Their activated expert count k^* is a primary lever for that cost. Because expert-count demand varies across tokens and layers, static Top-K overallocates routed compute. Recent work therefore studies token- and layer-wise count allocation. Among these, Ada-K (Yue et al., 2025a) trains a thin K-selector with reinforcement learning from next-token probability. That objective optimizes generation quality via a terminal likelihood signal; it does not directly score the count decision at each (token, layer). We argue that denser MoE-latent credit better matches k^* optimization, and propose **δ-Credit**: a reward that scores each layer's count decision by how closely the resulting MoE-block update (the change in hidden state across that layer's expert computation) matches a full-capacity (Top-K) teacher. Unlike Ada-K's single terminal likelihood signal broadcast to every depth, this credit is a dense per-layer score along the student trajectory. With this δ training target on Gemma 4 26B-A4B (reasoning-mode decoding, 64k context), δ-Credit outperforms Ada-K on GPQA Diamond / MMLU-Pro / MMMU-Pro at a lower mean activated count $\bar{k}^*$ (82.8/83.7/63.1 vs. 76.3/82.0/61.7; 5.60 vs. 6.15), reaching 98.8%/100%/94.7% of BF16 Top-8 accuracy at $\sim$70% routed MoE cost. The same recipe on Qwen3-30B-A3B (Qwen Team, 2025) (text-only) is again ahead of Ada-K on both GPQA Diamond and MMLU-Pro at comparable routed cost. These results support MoE-δ as an effective action-sensitive credit signal for learning dynamic k^* under our evaluation protocol, and show that when the training signal aligns with count allocation, adaptive routing can cut activated experts more aggressively while better preserving downstream accuracy.

1 INTRODUCTION

Sparse MoE transformers enlarge model capacity with a large expert pool, while activating only a few experts per token under a fixed Top-K to keep inference FLOPs in check (Shazeer et al., 2017; Fedus et al., 2022). Because expert-count demand varies across tokens and layers, static Top-K overallocates routed compute, and a uniform expert budget is a poor match (Fedus et al., 2022; Zhou et al., 2022). Ada-K (Yue et al., 2025a) therefore freezes a pretrained MoE and trains a post-hoc count allocator that chooses how many of the router's selected experts to run ($k^* \leq K$). With a small per-layer PPO allocator under $\mathcal{L}_{\mathrm{RL}} + \lambda\mathcal{L}_{\mathrm{reg}}$, it reports large FLOP cuts on Mixtral (Jiang et al., 2024). However, its credit remains a single last-layer next-token $\log P$ gap (R; Ada-K Eq. 6), broadcast to every depth with discount γ (A_ℓ; Ada-K Eq. 9). That signal is natural for language modeling, but the decision being optimized is not next-token identity; it is how many MoE experts to run at each (t, ℓ). Because a single scalar $\log P$ gap lacks depth-wise attribution, every layer inherits only a γ-rescaled copy of the same terminal score and gets little per-layer feedback for its count decision at that depth. Under a shared compression penalty, early- and late-layer k^* are then hard to credit distinctly, so the allocator can compress without a local MoE-update target that matches the action being optimized. We argue that count allocation needs credit tied to the MoE update itself, not a terminal LM scalar. **δ-Credit** (dense, layer-wise MoE-δ credit) keeps Ada-K's allocator interface and compression term, and replaces the broadcast $\log P$ advantage with dense per-(t, ℓ) MoE-block residual credit $\delta{=}h_{\mathrm{out}}{-}h_{\mathrm{in}}$ (Sec. 4; Fig. 1). Main results are on Gemma 4

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

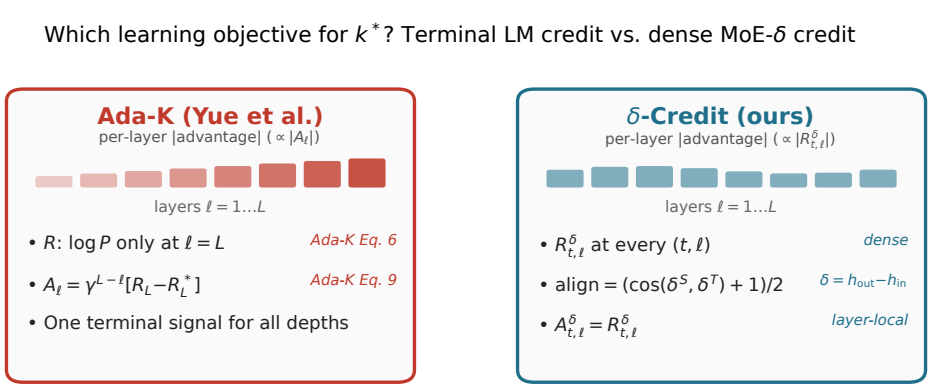


Figure 1: Ada-K vs. δ-Credit: same allocator and $\lambda\mathcal{L}_{\text{reg}}$; different PPO advantage ($R/A_\ell \to R^\delta/A^\delta$). Left: Ada-K R/A_ℓ (Ada-K Eqs. 6, 9; one last-layer $\log P$ signal, γ-scaled across depth). Right: δ-Credit R^δ/A^δ with $\text{align} = (\cos(\delta^S, \delta^T) + 1)/2$ (Eqs. (6)–(7)). Bars show per-layer absolute advantage ($|A_\ell|$ or $|R^\delta|$; darker/taller = larger).

26B-A4B (Gemma Team, 2026); we further test transfer to a routed-only MoE without a shared expert (Qwen3-30B-A3B, text-only) in Sec. 6 and App. A.11.

Contributions. (1) **Credit objective for k^*:** we identify a mismatch between Ada-K's generation-quality credit (R/A_ℓ) and the count action, and replace the broadcast last-layer $\log P$ gap with dense per-(t, ℓ) MoE-latent δ similarity against a Top-K teacher (R^δ/A^δ; Fig. 2, Tab. 1), so each layer's allocator receives dense credit from teacher–student MoE-update agreement along the current student trajectory rather than a γ-rescaled terminal LM scalar. (2) **Action-sensitive feature:** a frozen probe finds MoE-δ's mean drop-per-k $\sim$1.6$\times$ larger than full-hidden h (Fig. 3); as PPO credit, this signal punishes cutting k^* and resists the same $\lambda\mathcal{L}_{\text{reg}}$, yielding a gentler accuracy–compute trade-off (82.8% GPQA / $k^*\approx$5.6 vs. matched full-h 65.2% / $k^*\approx$4.6; App. A.4, Tab. 11). We do *not* claim that δ predicts a task-optimal k^*; the probe and ablation motivate δ as count credit rather than full-hidden similarity. (3) **Matched evaluation evidence:** on Gemma 4 and on a routed-only MoE without a shared expert (Qwen3-30B-A3B), the same recipe improves the observed accuracy–compute trade-off over Ada-K—lower $\bar{k}^*$ on Gemma, and matched or lower routed cost on Qwen—with higher matched decode throughput on Gemma (Sec. 5; Tabs. 2, 5, 19).

2 Related Work

Sparse MoE and scaling. Conditional computation routes each token to a subset of experts (Shazeer et al., 2017). Switch Transformer (Fedus et al., 2022), GShard (Lepikhin et al., 2021), and ST-MoE (Zoph et al., 2022) scale sparse MoE to large language models; GLaM (Du et al., 2022) and Mixtral (Jiang et al., 2024) show strong quality at sub-linear active compute. Expert Choice routing (Zhou et al., 2022) lets experts select tokens rather than each token picking a fixed Top-K. Soft MoE (Puigcerver et al., 2024) replaces hard Top-K with a differentiable weighted mix over experts, so it does not hard-select a discrete expert set. By contrast, Ada-K / δ-Credit keep the router's selected experts and learn only how many of them to run ($k^* \leq K$). Recent open MoEs also explore shared-expert designs (a shared feed-forward that is always active alongside Top-K routing), as in Gemma 4 (Gemma Team, 2026).

Expert importance and compression. Prior work prunes or merges experts offline, or identifies a small set of indispensable *Super Experts* whose removal collapses reasoning (Su et al., 2026). We instead keep expert IDs and the router's selected set fixed, and learn only how many of those selected experts to run per (t, ℓ).

Dynamic and adaptive routing. The closest prior for per-token, per-layer expert *counts* is Ada-K (Yue et al., 2025a): a frozen Top-K router and experts, with only a small count allocator trained post hoc by PPO under expected-count $\mathcal{L}_{\text{reg}}$ and nucleus warm-start (**P-Warm**) (Holtzman et al., 2020) (Sec. 3). δ-Credit keeps that interface and changes the credit. Other variable-count schemes do

not learn such a PPO allocator: router-confidence thresholds (Huang et al., 2024), null experts inside Top-K (Zeng et al., 2024), jointly trained gates that can grow or prune the pool (Guo et al., 2025), threshold routers that redesign the MoE (Yang et al., 2024), edge sensitivity gating (Zhong et al., 2024), layer skipping (Elbayad et al., 2020), static per-layer budget reallocation (Li et al., 2026), straight-through masks over expert IDs (Wu et al., 2026), and training-free router signals (Yang et al., 2026).

RL for discrete routing and teacher credit. Integer counts k^* are non-differentiable, so Ada-K and this work both train allocators with PPO (Schulman et al., 2017; 2015) rather than backprop through the discrete action, freezing the router and choosing how many of the selected experts to run. Both use PPO's clipped importance-ratio objective (Schulman et al., 2017) with a scalar advantage A; what differs is how A is built on the rollout. Concurrent MoE+RL work improves *which* experts fire via exploration-driven routing (Kim & Kang, 2025); we instead keep the router's selected set fixed and learn only the count k^* under a dense MoE-δ credit. Representation distillation matches student and teacher features (Hinton et al., 2015; Kornblith et al., 2019); we extend that idea into PPO by scoring, at each token and MoE layer, how the MoE representation changes under the chosen k^*, and use that score as a dense per-(t, ℓ) reward R^δ for optimizing k^* without updating backbone weights (feature choice: Sec. 4).

3 Background

Gemma 4 default routing and compute. Gemma 4 26B-A4B (Gemma Team, 2026) is a shared-expert MoE: each layer has 128 routed experts plus one shared expert that is always active ($3\times$ the FFN width of a routed expert). By default the router activates a fixed Top-8 of the routed experts for every token, using the same K on easy and hard tokens alike. We report **Active** as the forward-pass parameter count (billions): non-MoE weights and the shared expert are held fixed at the Top-8 baseline, and only the routed MoE block is scaled by the mean activated fraction $K/8$

$$\text{active}(K) \approx 1.84 + 0.54 + 1.43 \cdot (K/8) \quad [\text{B}], \tag{1}$$

so Top-8 is ≈3.81B (1.84 non-MoE + 0.54 shared + 1.43 routed) and **Cost** $= K/8$ is that routed fraction relative to Top-8. Ada-K / δ-Credit keep the router's selected experts (Top-8 on Gemma) and learn $k^* \in \{1, \ldots, 8\}$; tables plug $K{=}\bar{k}^*$ into Eq. (1) (Tab. 2, 3). We modify the llama.cpp inference stack so unused expert FFN slots skip compute and MoE cost scales with that ratio.

Ada-K baseline. We reproduce Ada-K (Yue et al., 2025a) on Gemma for a matched comparison: same count machinery, different credit. The backbone and router stay frozen; a per-layer linear allocator chooses how many of the selected experts to keep ($k^* \le K$; on Gemma $K{=}8$, $L{=}30$):

$$P_{\text{alloc}}(x_i) = \text{Softmax}(W_{\text{alloc}}\, x_i), \qquad k^* \sim P_{\text{alloc}}(x_i) \quad \text{(Ada-K Eqs. 4–5).} \tag{2}$$

State s_ℓ is the layer-ℓ hidden of token x_i; action $\hat{c}_\ell \equiv k^*$. Discrete counts use PPO with a clipped surrogate (Ada-K Eqs. 7–8; App. A.14) after a short supervised warm-start that we call **P-Warm**: nucleus labeling of expert counts (Holtzman et al., 2020) with cross-entropy (Ada-K Eqs. 12–13; $p^* = 0.23$; Sec. 6).

Ada-K's reward is last-layer next-token $\log P$ only (Ada-K Eq. 6), with γ-broadcast advantage (Eq. 9) and expected-count regularizer (Eqs. 10–11). We change that credit:

$$R(\hat{c}_\ell, s_\ell) = \log P(x_i \mid x_1, \ldots, x_{i-1}) \cdot \mathbf{1}_{\ell=L}, \tag{3}$$

$$A_\ell = \gamma^{L-\ell}\big[R(\hat{c}_L, s_L) - R(c_L^*, s_L^*)\big], \tag{4}$$

$$\mathcal{L}_{\text{reg}} = \tfrac{1}{L}\sum_{\ell=1}^{L}\sum_{n=1}^{N} n\, P_{\theta_\ell}(n), \qquad \mathcal{L} = \mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\text{reg}}, \tag{5}$$

where $*$ is forced Top-K and $N{=}K{=}8$. Every depth gets the same terminal $\log P$ gap, re-scaled by $\gamma^{L-\ell}$. Reproduction: $\gamma{=}0.99$, $\lambda{=}0.003$. Symbol tables for Ada-K and δ-Credit are in App. A.2.

4 METHOD: δ-CREDIT

Overview. Tab. 1 contrasts two credit signals for the same count decision $k^*_{t,\ell}$. We hold the outer loss $\mathcal{L}{=}\mathcal{L}_{\mathrm{RL}} + \lambda\mathcal{L}_{\mathrm{reg}}$, per-layer count allocator (Eq. (5)), and PPO framework fixed (λ is method-specific; App. A.7), isolating credit design as the sole variable: Ada-K's single terminal $\log P$ gap, γ-broadcast to every depth, is swapped out for dense per-(t,ℓ) MoE-δ scores $R^{\delta}_{t,\ell}$ along the student trajectory. Below we define the MoE-δ capture, the align reward, and the PPO update.

Table 1: Same outer loss; methods differ in reward target and in how densely credit is assigned across layers.

	Ada-K (Yue et al., 2025a)	δ-Credit (ours)
Instantaneous reward	R (Eq. (3)): $\log P$ only if $\ell{=}L$	$R^{\delta}_{t,\ell}$ (Eq. (7)): align at *every* ℓ
Advantage in $\mathcal{L}_{\mathrm{RL}}$	A_ℓ (Eq. (4)): γ-scaled terminal gap	$A^{\delta}_{t,\ell}{=}R^{\delta}_{t,\ell}$ (no γ broadcast)
What is scored	next-token $\log P$ vs. Top-K baseline	MoE-δ vs. Top-8 teacher
Per-(t,ℓ) MoE score	No	Yes

Notation. Let t index tokens and $\ell \in \{1,\dots,L\}$ layers ($L{=}30$; plots use $0,\dots,29$). Hidden $h_{t,\ell}$ is the allocator state (Ada-K's s_ℓ: the per-layer hidden fed to the count head); $k^*_{t,\ell} \in \{1,\dots,8\}$ is the sampled count action (Ada-K's $\hat{c}_\ell$), not an expert-ID choice. **Teacher** T uses the same frozen weights with forced Top-K (here $K{=}8$); **student** S uses allocator-sampled $k^* \le K$. Both share frozen weights and the same router parameters, but they run *independent* forwards: each applies the router to its own layer input and forms a Top-K ranking; the student then keeps only the first k^* slots and renormalizes remaining weights, $\tilde{w}_i = w_i \mathbf{1}[i \le k^*] / \sum_{j \le k^*} w_j$. Selected expert IDs can therefore diverge once h^S and h^T diverge across depth; δ-Credit scores the resulting MoE-update match under those forwards, not a same-state counterfactual that holds IDs fixed. Because upstream states may already differ, $R^{\delta}_{t,\ell}$ is dense trajectory credit—teacher–student MoE-update agreement at (t,ℓ) under the current student path—not a causal estimate of the isolated effect of $k^*_{t,\ell}$.

Capture points. Define $h^{\mathrm{in}}_{t,\ell}$ as the hidden at the MoE input (post-attention / router input) and $h^{\mathrm{out}}_{t,\ell}$ as the decoder-layer output after the MoE residual add. The scored vector is the MoE-block residual update $\delta_{t,\ell} = h^{\mathrm{out}}_{t,\ell} - h^{\mathrm{in}}_{t,\ell}$, not the full residual stream h^{out}, and not a routed-only residual obtained by subtracting the shared expert (Fig. 2a). So δ is defined from MoE-block input/output alone—one $(h^{\mathrm{in}}, h^{\mathrm{out}})$ pair per MoE layer with a count allocator—and is not tied to a shared-expert layout; the same capture applies whenever pre- and post-expert hiddens are available (e.g. MoE without a shared FFN). On Gemma the MoE block runs the shared feed-forward and the Top-k^* routed experts in parallel from the same post-attention residual; k^* changes only the routed branch, and we do *not* subtract the shared contribution, so δ still includes it (Fig. 2b).

Why MoE-δ, not full-hidden cosine. A representation reward could instead compare student and teacher full hiddens h^{out} after each MoE block—that is what the next sublayer actually reads. The problem is that h^{out} still carries most of the pre-MoE residual stream, so when we reduce k^* the cosine barely moves: the routed update is a small perturbation on a vector that is already close to the teacher. MoE-δ isolates that perturbation (Sec. 4), so similarity to the teacher reflects the MoE block's contribution rather than the inherited residual.

We answer two questions in order and keep them separate: (i) without training, which feature falls faster when we force smaller k^* (frozen probe); (ii) under the same λ, what happens if we plug that feature into PPO as the credit (matched train logs in App. A.4).

(i) Frozen probe: δ reacts more to smaller k^*. On Gemma 4 we sweep forced student $k^* \in \{1,\dots,8\}$ against a Top-8 teacher on the same text batch (no PPO, identical tokens). Both $\cos(h^S, h^T)$ and $\cos(\delta^S, \delta^T)$ decrease as k^* drops, but the mean drop-per-k is $\sim$1.6$\times$ larger for MoE-δ (Fig. 3; App. A.4): removing routed experts hurts δ-similarity more than full-h similarity, so a δ-based reward gives PPO a clearer signal when the allocator cuts count. In the range we actually train ($\bar{k}^* \approx$ 5–6; probe $k^* \in \{4,\dots,7\}$), $\cos(\delta)$ stays high enough to optimize ($\approx$0.89–0.96) yet still spans a wider band than $\cos(h)$.

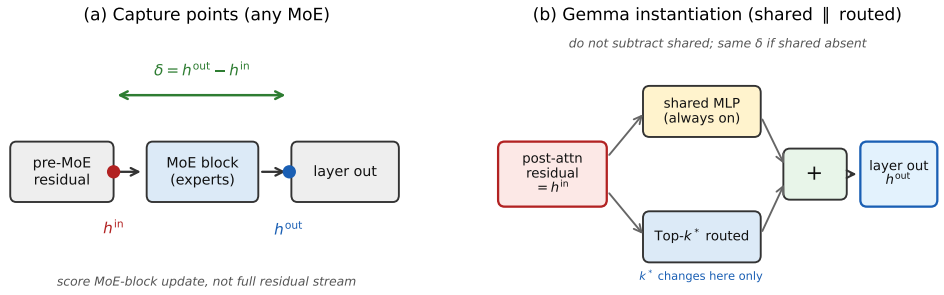


Figure 2: Where δ-Credit looks. (a) General capture from MoE input/output: h^{in}, h^{out}; $\delta{=}h^{\text{out}}{-}h^{\text{in}}$ (not tied to shared experts). (b) Gemma instantiation: shared MLP and Top-k^* routed experts run in parallel from the same post-attention residual (shared contribution not subtracted).

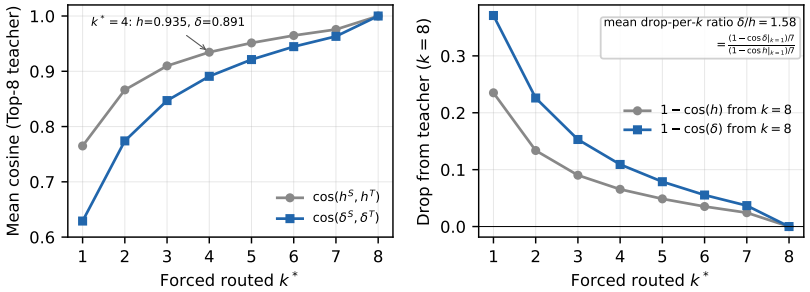


Figure 3: **Probe** (not a training curve): forced student k^* vs. Top-8 teacher. Left: mean $\cos(h)$ vs. $\cos(\delta)$. Right: drop from $k^*=8$ and the δ/h mean-drop-per-k ratio ($\sim$1.6$\times$)—the stronger count penalty. Learning under the same λ: Fig. 6 (App.).

Near our trained regime, e.g. at forced $k^*=4$, mean $\cos(h)$ is 0.935 while $\cos(\delta)$ is 0.891; the right panel's $\sim$1.6$\times$ is the ratio of those mean drops from the $k^*=8$ teacher.

(ii) Same $\lambda\mathcal{L}_{\text{reg}}$: weak h-credit lets the compression term over-compress. A matched PPO swap (only the cosine feature; App. A.4, Fig. 6) shows what the probe gap implies once we train. Because MoE-δ falls faster when k^* is cut, the PPO reward itself punishes over-compression and resists the same $\lambda\mathcal{L}_{\text{reg}}$: matched δ-Credit stays near $\bar{k}^*{\approx}5.6$ with a non-collapsed LM-gap and holds GPQA Diamond at 82.8% / $k^*{\approx}5.56$. Full-h under-reacts, so in-reward align stays high ($\approx$0.95) even as activation collapses and the LM-gap monitor fails—$\lambda\mathcal{L}_{\text{reg}}$ wins and the selected checkpoint falls to 65.2% / $k^*{\approx}4.61$ (Tab. 11). That larger activation/LM motion in Fig. 6 is this weak-credit failure—*not* evidence that h is "more sensitive"; probe sensitivity is established in (i) / Fig. 3.

We do *not* require local MoE-δ similarity to identify a task-optimal expert count. Its role is an *action-sensitive* reward: variation with k lets PPO distinguish retaining from removing experts; $\lambda\mathcal{L}_{\text{reg}}$ supplies the opposing pressure toward smaller k. The resulting policy is evaluated by downstream accuracy–compute trade-offs, not by whether δ predicts an oracle k^*. The full hidden h contains a large residual that dilutes the routed-k^* effect; we therefore adopt MoE-δ as the sole score in R^{δ}.

We map student/teacher increment cosine into $[0,1]$, mean-center over the rollout batch $\mathcal{B}$ at layer ℓ (scale $\alpha{>}0$), and use that as reward and advantage:

$$\delta_{t,\ell} = h^{\text{out}}_{t,\ell} - h^{\text{in}}_{t,\ell}, \qquad \text{align}_{t,\ell} = \frac{\cos(\delta^S_{t,\ell}, \delta^T_{t,\ell}) + 1}{2} \in [0,1], \tag{6}$$

$$R^{\delta}_{t,\ell} = \alpha \cdot \big(\text{align}_{t,\ell} - \text{mean}_{\mathcal{B},\ell}(\text{align})\big), \qquad A^{\delta}_{t,\ell} = R^{\delta}_{t,\ell}. \tag{7}$$

$R^{\delta}_{t,\ell}$ is high when token t at layer ℓ has higher align than the same-depth batch mean: its MoE-δ is closer (in cosine) to the Top-8 teacher than peers at ℓ. Mean-centering is applied per layer as a baseline for variance reduction and *signed* advantages under PPO clipping: raw MoE-δ align sits in a high, narrow band ($\approx$0.95), so subtracting the layer-batch mean does *not* enlarge pairwise gaps ($a_i - a_j$ is unchanged by a common subtractor) but expresses each score relative to the same-layer

batch mean and centers advantages around zero. Because the mean is taken within each layer, $R^{\delta}_{t,\ell}$ ranks tokens only against peers at depth ℓ; it never reuses a terminal $\log P$ score across depth the way Ada-K's γ-broadcast A_ℓ does ("layer-local" here means per-(t,ℓ) dense scoring, not a claim that $R^{\delta}_{t,\ell}$ isolates $k^*_{t,\ell}$ from upstream drift). We use align alone in R^{δ}: Ada-K's A_ℓ and MoE-δ credit are different axes, and mixing them underperforms on Gemma (App. A.5; hybrid GPQA collapses vs. pure R^{δ}). PPO is thus pushed to raise within-layer relative R^{δ} with as few experts as needed, while $\lambda\mathcal{L}_{\mathrm{reg}}$ penalizes excess expected count. No likelihood term enters the policy-gradient reward (Eqs. (6)–(7)); likelihood is used only for checkpoint selection (Sec. 5). We do not train a value baseline—$A^{\delta}{=}R^{\delta}$ enters the clipped PPO surrogate directly—and teacher δ^T is computed with a Top-8 forward without gradient on the same tokens. Hyperparameters: $\alpha{=}1.0$, $\lambda{=}0.001$; checkpointing in Sec. 5 (Tab. 6; symbols: App. A.2).

PPO update. At each (t,ℓ), sample student $k^*_{t,\ell}$ vs. a forced Top-8 teacher, form mean-centered $R^{\delta}_{t,\ell}{=}A^{\delta}_{t,\ell}$ from δ align, and score only that layer's allocator (pad-masked layer averages in App. A.14). Following Ada-K (Yue et al., 2025a) (Sec. 3), discrete k^* is non-differentiable, so we train allocators with PPO (Schulman et al., 2017): state $h_{t,\ell}$, action $k^*_{t,\ell}$, policy $\pi_{\theta_\ell}(\cdot \mid h_{t,\ell})$. A rollout records $(h, k^*, \log \pi_{\mathrm{old}}, R^{\delta})$; with importance ratio $\rho = \pi_{\theta_\ell}(k^* \mid h)/\pi^{\mathrm{old}}_{\theta_\ell}(k^* \mid h)$ and clip $\varepsilon{=}0.2$,

$$\mathcal{L}_{\mathrm{RL}} = -\mathbb{E}\big[\min\big(\rho A,\ \mathrm{clip}(\rho, 1{-}\varepsilon, 1{+}\varepsilon)\, A\big)\big], \tag{8}$$

where we set $A{=}A^{\delta}_{t,\ell} = R^{\delta}_{t,\ell}$ at every (t,ℓ) (Ada-K Eqs. 7–8 use the same clip form with $A{=}A_\ell$). Outer $\mathcal{L}{=}\mathcal{L}_{\mathrm{RL}} + \lambda\mathcal{L}_{\mathrm{reg}}$ is unchanged.

5 Experiments

Setup. Base model: Gemma 4 26B-A4B instruction-tuned (frozen). Trainable: only the 30 per-layer linear count heads (hidden $2816 \to 8$ logits; $\sim$0.68M parameters); the backbone, router, and shared expert stay frozen. Data: $N{=}9{,}000$ text/agent trajectories (Tab. 9; no images); AdamW (Loshchilov & Hutter, 2019) learning rate 10^{-3}; two PPO update epochs per rollout buffer. After one CE router warm-start epoch, allocator-only PPO runs for one epoch on the same mix. Training is PyTorch on one GPU of a dual NVIDIA RTX PRO 6000 Blackwell Max-Q workstation ($\sim$96 GB VRAM each); Table 2 evaluates BF16 checkpoints through a patched llama.cpp server (App. A.15). Hyperparameters: Tab. 6.

Checkpoints and evaluation. All reported numbers use the **selected training checkpoint** under a fixed log protocol (not a held-out validation set). This is our **common checkpoint rule** for matched Ada-K vs. δ-Credit comparison; Yue et al. report a one-epoch run and do not describe such a rule. After a 5-step warm-up, a step qualifies once its 10-step moving average (MA) of a quality score is defined and satisfies LM-gap MA$\geq$0; among qualifiers we keep the lowest MA of mean activated count (train-time proxy for $\bar{k}^*$). Both methods use the same Top-K-baselined gap $R(\hat{c}_L, s_L) - R(c^*_L, s^*_L)$ as that quality score (Ada-K's training signal; for δ-Credit a logged monitor only, not part of R^{δ}). Dense R^{δ} remains the per-layer PPO credit (trajectory MoE-update scores); the LM-gap is only the checkpoint gate under this common rule. If no step clears LM-gap MA$\geq$0, we take the post-warmup maximizer of (LM-gap MA) $-$ (activation MA)$/K$ (ties: lower activation MA). Training-log evidence is App. A.16–A.17; an Ada-only relaxation to LM-gap MA≥ -0.05 is App. A.16 (ablation).

Compression penalty λ. We choose λ with short 25-step PPO sweeps, scoring the last 20 steps (App. A.7), separately from the checkpoint rule above. A λ is eligible if the mean, over that last-20 window, of the 10-step moving average of $R(\hat{c}_L, s_L) - R(c^*_L, s^*_L)$ exceeds -0.05 (training has not collapsed); among eligible λ we keep the lowest window mean of the same activation moving average. We apply this common LM-gap filter to every method and model so λ remains comparable; once λ is fixed, each full run uses the common checkpoint rule in Sec. 5 (LM-gap MA$\geq$0, then min activation MA). This yields $\lambda{=}0.001$ for δ-Credit and $\lambda{=}0.002$ for Ada-K on Gemma under that common λ rule. Table 2 additionally reports Ada-K at the original paper value $\lambda{=}0.003$ (same common *checkpoint* rule as δ-Credit; the common-λ pick $\lambda{=}0.002$ under that same checkpoint rule is App. A.8). If no sweep candidate clears -0.05, App. A.7 takes the highest window-mean LM-gap MA.

Inference. Allocator weights are embedded in BF16 llama.cpp checkpoints; inference uses a modified llama.cpp build with dynamic k^* masking after Top-8 router selection and unused expert FFN slots skip compute (Sec. 3); the shared expert remains always active.

Evaluation. All methods use the same decoding stack as the Gemma 4 model card's reasoning mode (chain-of-thought enabled): T=1.0, top-p=0.95, top-k=64, seed 42; 64k context, up to 57k generated tokens. **Primary baseline:** BF16 Top-8 (default Gemma 4 routing with fixed K=8 and no allocator) on that same stack (Table 2, first row). **Benchmarks:** GPQA Diamond (Rein et al., 2024) (n=198; not the full GPQA set), MMLU-Pro (Wang et al., 2024) (n=12032), MMMU-Pro (Yue et al., 2025b) (n=1730). Reported $\bar{k}^*$ is the unweighted mean, over the three benchmarks, of each benchmark's per-request mean routed-expert count on the same checkpoint (Active/Cost: Eq. (1)). We also list those per-benchmark means so the compression is not visible only as a three-way average.

Table 2: Main results on Gemma 4 26B-A4B (reasoning-mode decoding, 64k; checkpoint from the common selection rule). $\bar{k}^*$: unweighted mean of per-benchmark request-mean k^* (GPQA Diamond / MMLU-Pro / MMMU-Pro); Active/Cost from Eq. (1). Per-benchmark k^*: Ada-K 6.03/6.15/6.28; δ-Credit 5.56/5.62/5.63. Ada-K row: Yue et al. paper λ=0.003 (Eqs. 6, 9, γ=0.99), common checkpoint rule (App. A.16). Our common-λ Ada pick λ=0.002 is also below δ on every bench (80.8/82.1/62.3 at $\bar{k}^*$=6.35; App. A.8).

Method	$\bar{k}^*$	Active	GPQA	MMLU-Pro	MMMU-Pro
BF16 Top-8	8.00	3.81B	83.8	83.7	66.6
Ada-K (R/A_ℓ)	6.15	3.48B	76.3	82.0	61.7
δ-Credit (ours)	5.60	3.38B	82.8	83.7	63.1

Table 3: Per-token MoE compute vs. default Top-8 (same $\bar{k}^*$/Active as Table 2; Eq. (1)). Absolute FLOPs: App. A.13.

Setting	$\bar{k}^*$	Active	Cost
Top-8	8.00	3.81B	100%
Ada-K	6.15	3.48B	77%
δ-Credit	5.60	3.38B	70%

Results. Table 2 uses Ada-K's published λ=0.003; the common-λ pick λ=0.002 remains below δ on every bench (caption; App. A.8). Under the same allocator pipeline and eval stack, δ-Credit outperforms Ada-K (λ=0.003) on all three benchmarks at lower cost: GPQA Diamond / MMLU-Pro / MMMU-Pro improve by +6.5 / +1.7 / +1.4 percentage points while three-benchmark-mean $\bar{k}^*$ drops from 6.15 to 5.60 (Active 3.38B vs. 3.48B; per-benchmark k^* in Tab. 2 caption; Tab. 3), and matched decode throughput rises (Tab. 19). Paired McNemar vs. Ada-K: MMLU-Pro (n=12,032) favors δ-Credit (598 vs. 404; $p<0.001$) and GPQA Diamond (n=198; 24 vs. 11; $p\approx0.041$) is significant; MMMU-Pro (n=1,730; 149 vs. 125; $p\approx0.16$) remains directional; we discuss this multimodal limitation below. An Ada-only ablation that relaxes the LM-gap floor to MA≥-0.05 on the same Ada-K run yields a stronger Ada checkpoint than Table 2, but it remains below δ-Credit on every bench (App. A.16). Relative to BF16 Top-8, δ-Credit reaches 98.8% / 100% / 94.7% of Top-8 accuracy at ∼70% routed MoE cost (Fig. 4).

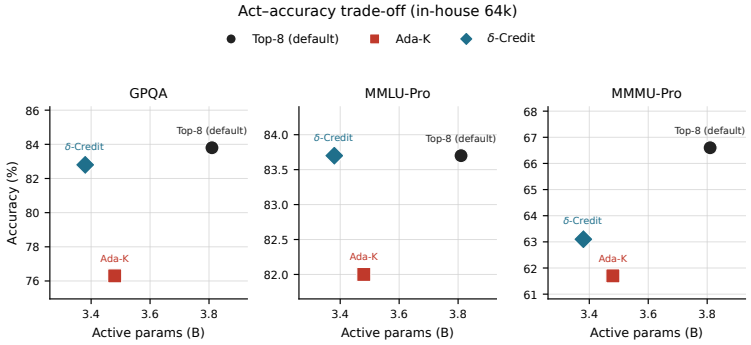


Figure 4: Accuracy vs. active-parameter trade-off (matched reasoning-mode decoding, 64k). Markers: Top-8 / Ada-K / δ-Credit; x-axis uses Table 2 three-bench-mean active parameters (left = cheaper).

6 Analysis and Limitations

Layer-heterogeneous routing (before and after PPO). Gemma routing demand varies by depth, which motivates per-depth R^δ rather than Ada-K's depth-broadcast A_ℓ. We summarize three snapshots on the same depth bands (Tab. 4): (i) router warm-start nucleus labels *before* PPO; (ii) Ada-K $\bar{k}^*$ at its selected training checkpoint; (iii) δ-Credit $\bar{k}^*$ at its selected training checkpoint (Fig. 5; selection rule: App. A.16).

Before PPO (router warm-start). We calibrate a global nucleus threshold $p^* = 0.23$ so the mean nucleus width $\mathbb{E}[n(p^*)]$ *before* clamping to K is $\approx$8.03, matching Top-8 (Ada-K Eq. 13). Allocator CE labels then clamp $n_{t,\ell}(p^*)$ to $\{1, \ldots, 8\}$, with pooled mean $\approx$6.58 over $\sim$1.28M token–layer pairs. Even this warm-start supervision is layer-heterogeneous (early / middle / late $\approx$6.48 / 7.45 / 5.81), in contrast to Ada-K's single terminal $R(\hat{c}_L, s_L)$.

After PPO (learned k^*). Layer-wise mean-centering in Eq. (7) removes each layer's absolute align level from R^δ: advantages rank tokens only against peers at the same depth, so a high early-layer align band and a lower late-layer band do *not* by themselves supply a cross-depth "compress where easy" signal. Fig. 5 (bottom) nevertheless shows a post-training correlation: under δ-Credit, per-layer align (logged monitor) is highest early and lower late, and coincides with lower early / higher late $\bar{k}^*$ (Pearson $r \approx -0.54$ over the 30 layers). We treat that as an empirical depth pattern, not as proof that absolute align levels drive allocation through R^δ. What the matched logs *do* show is that the pattern is not merely leftover P-Warm structure: Ada-K and δ-Credit start from the same nucleus warm-start, yet diverge after PPO—δ-Credit cuts early more than Ada-K (Early: 4.56 vs. 6.12) and keeps slightly more late (Late: 5.54 vs. 5.35; fine bands in Tab. 13), while both keep mid-depth $\bar{k}^*$ high (Tab. 4). Ada-K's broadcast terminal $\log P$ advantage does not recover that early↓/late↑ contrast relative to δ-Credit. A fuller account would need per-layer align-vs-k slopes or centering/warm-start ablations; we do not claim those mechanisms here.

Table 4: Depth-band mean routing before and after PPO: P-Warm ($p^* = 0.23$) vs. selected-checkpoint $\bar{k}^*$ (fine: Tab. 13).

Depth band	P-Warm $\bar{n}$	Ada-K $\bar{k}^*$	δ-Credit $\bar{k}^*$
Early (L0–L9)	6.48	6.12	4.56
Middle (L10–L19)	7.45	7.40	6.70
Late (L20–L29)	5.81	5.35	5.54

Layer-wise expert count (before / after PPO)

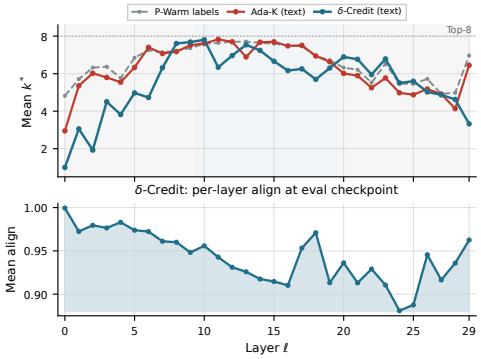


Figure 5: Top: mean k^* by layer (P-Warm; Ada-K / δ-Credit at selected training checkpoints). Bottom: δ-Credit per-layer align monitor (not the PPO advantage after mean-centering; not the checkpoint floor). Higher early-layer align *coincides with* lower k^*, and lower late-layer align with higher k^* (band means: Tab. 4); this is a post-hoc correlation under the learned policy.

Alignment vs. Ada-K R monitor. Under δ-Credit, Ada-K's $R(\hat{c}_L, s_L) - R(c_L^*, s_L^*)$ is logged as a monitor alongside align (App. Figs. 8–10). The two moving averages are largely decoupled ($r \approx 0.08$); $\mathcal{L}_{\mathrm{reg}}$ curves: Fig. 11. On matched Gemma full runs, the same LM-gap MA stays non-negative for most post-warmup δ-Credit steps while activation falls below Ada-K's selected checkpoint, whereas the Ada-K run itself rarely recovers LM-gap MA$\geq$0 once compression proceeds (Fig. 7; App. A.17).

Matched wall-clock latency. Routed MoE cost (Tab. 3) is not wall-clock: on matched GPQA Diamond decode (n=20), δ-Credit raises tokens/s by $+5.9\%$ vs. Top-8 (Ada-K in between); a $\sim$30% routed-MoE FLOP cut yields only single-digit wall speedup because attention, KV, and the shared expert dominate (Tab. 19; App. A.12).

Transfer beyond shared-expert MoEs (Qwen3-30B-A3B, text-only). We apply the same recipe to Qwen3-30B-A3B—a routed-only MoE with no shared expert (48 MoE layers; top-8 of 128; text-only; λ and decoding details in App. A.11). This is a matched transfer test of δ-Credit versus Ada-K, not a causal test of shared-expert effects. In Tab. 5, δ-Credit at our common-λ pick (λ=0.006) sits near Ada-K's paper λ=0.003 cost (80% vs. 78%) at much higher accuracy (GPQA 59.1 vs. 46.0; MMLU-Pro 75.8 vs. 68.1); raising λ to 0.007 cuts cost to 72% while remaining well above Ada-K 0.003. Under the same λ rule, Ada-K's pick (λ=0.002) matches δ's 80% cost but still trails (App. A.11; paired McNemar). Ada-K falls farther below Top-8 on Qwen than on Gemma, widening the paired δ–Ada gap; with no shared expert, Active scales the full routed MoE by mean $k^*/8$ (Eq. (10)).

Table 5: Qwen3-30B-A3B (text-only): accuracy and routing cost. $k^*_{\mathrm{G}}/k^*_{\mathrm{M}}$: per-bench request-mean k^* (same common checkpoint rule as Tab. 2); Active/Cost from Eq. (10). Main rows: Ada-K paper λ=0.003; δ common-λ pick 0.006; δ higher-λ ablation 0.007. Ada-K rule pick 0.002 (matched 80% cost): App. A.11.

Method	λ	GPQA	MMLU-Pro	k^*_{G}	k^*_{M}	Active	Cost
BF16 Top-8	—	61.6	77.0	8.00	8.00	3.35B	100%
Ada-K (R/A_ℓ)	0.003	46.0	68.1	6.31	6.21	2.96B	78%
δ-Credit (ours)	0.006	59.1	75.8	6.51	6.29	2.99B	80%
δ-Credit	0.007	56.1	75.2	5.88	5.70	2.85B	72%

Limitations. Multimodal / early-layer allocation. On MMMU-Pro, δ-Credit trails BF16 Top-8 (-3.5 pp) and Uniform-K=6; $+1{,}000$ vision captions do not close the gap (App. A.9, A.10)—a multimodal limit, not a main claim. **Scope.** Mixtral/DeepSeek-MoE reproductions are out of scope: our comparison targets the credit-design axis (terminal vs. dense reward) under matched allocator/PPO/checkpoint rules, not a validation against Ada-K's published numbers on those architectures. Richer count heads and explaining Ada-K's larger Qwen drop vs. Top-8 are similarly out of scope; App. A.12 is single-request llama.cpp decode (not batched serving).

7 Conclusion

We study optimizing the activated expert count k^* in MoE LLMs by assigning credit at the same token–layer granularity as the count decision, rather than relying solely on terminal generation-quality signals. Broadcast terminal next-token $\log P$ (Ada-K) matches language modeling, but provides indirect credit for deciding how many routed experts to run at each (t, ℓ). δ-Credit addresses this gap by keeping Ada-K's per-layer allocator and $\lambda\mathcal{L}_{\mathrm{reg}}$, while replacing its terminal LM-based PPO credit with dense MoE-δ agreement to a Top-K teacher along the student trajectory. A frozen probe finds MoE-δ's mean drop-per-k $\sim$1.6$\times$ larger than full-hidden cosine (Fig. 3); as PPO credit, this action sensitivity avoids the matched full-h degradation and preserves GPQA (82.8% vs. 65.2%; App. A.4). Within the same allocator and PPO framework, replacing terminal LM credit with MoE-δ improves the observed accuracy–compute trade-off over Ada-K on Gemma 4 (lower $\bar{k}^*$, higher matched decode throughput; Tab. 2, Tab. 19) and on routed-only Qwen3-30B-A3B (Tab. 5). These results support MoE-δ as an effective action-sensitive credit signal for learning dynamic expert counts under our evaluation protocol—not uniformly optimal across modalities (MMMU-Pro gap remains)—and suggest that adaptive MoE routing can benefit from per-layer MoE updates along the student trajectory rather than relying exclusively on terminal LM reward.

ETHICS STATEMENT

We freeze a public base LLM and train only lightweight routing allocators on a mixed corpus (Tab. 9): mostly public instruction and agent data, plus a small **internal** agent Q&A slice ($\approx$11%) from proprietary multi-turn workflows that we do not redistribute. An additional internal vision–ontology caption slice (n=1,000) is used only in App. A.10 and is likewise not redistributed. Benchmarks are standard academic sets; no human subjects. Efficiency gains may reduce inference energy use; misuse risks are inherited from the base model.

REPRODUCIBILITY STATEMENT

Anonymous supplementary material includes training/evaluation configs, the Ada-K-compatible inference patch, scripts to select λ/checkpoints under the paper rules, embed allocator weights into llama.cpp checkpoints, and CPU smoke tests for the reward/PPO stack. Hyperparameters: λ=0.001 (δ-Credit), λ=0.003 (Ada-K Table 2), γ=0.99, router warm-start then PPO, mean-centered alignment advantages. Hardware: dual NVIDIA RTX PRO 6000 Blackwell Max-Q GPUs ($\sim$96 GB VRAM each); details in Sec. 5. Checkpoint selection is App. A.16; training mix is App. A.3. Exact Table 2 training needs that unreleased internal agent slice ($\approx$11%); the supplementary code documents a public-only mix that matches the protocol but is not bit-identical to the paper mix. Base LLM weights, trained `alloc_*.pt`, and raw training JSONL are not in the OpenReview zip (size/license); evaluation benches are public. Eq. (1) and Eq. (7) fully specify Gemma active parameters and the reward; Qwen Active is Eq. (10); MoE FLOP counting is in App. A.13.

AI USE STATEMENT

Generative AI tools (coding assistants and LLM-based editors) were used in this work as follows.

Tools and models. Assistance was delivered through the Cursor IDE agent interface, using (non-exhaustively, across drafting and coding sessions): Cursor Grok 4.6; Claude Opus 5, Opus 4.8, and Opus 4.7; Claude Sonnet 5; Claude Fable 5.1 and Fable 5; and GPT-5.6 Sol and GPT-5.5. We did not use a single fixed model for all tasks; authors selected models per session for coding, editing, or log triage.

Required disclosure (used). We used AI assistants to **implement and refactor** training, evaluation, and tooling scripts (PPO rollout loop, benchmark clients, checkpoint embedding, and figure-generation scripts). We used AI tools to **visualize and organize** experiment logs when drafting the checkpoint protocol and results tables. We also used AI assistants to help **search and triage** recent related work; all cited papers were read and verified by the authors. All AI-assisted scripts were executed, reviewed, and corrected by the authors on real hardware; benchmark numbers in Table 2 come from author-run eval jobs, not from model-generated claims.

Required disclosure (not used / not applicable). We did **not** use generative AI to synthesize training or benchmark datasets, to prove theoretical claims, or to auto-generate core methodological ideas (δ-Credit reward design, PPO credit assignment, and Ada-K baseline reproduction followed published equations and author-implemented code).

Recommended disclosure (used). We used AI assistants to **draft and edit** sections of the English manuscript (introduction framing, related-work paragraphs, reproducibility text), to improve readability and LaTeX structure, and to format references. Figure layouts and plotting scripts were AI-assisted but fed only author-collected metrics.

Author review. We manually verified every scientific claim, equation reference, table entry, and cited result against logs and result JSON. We take full responsibility for the submitted paper and any content produced with AI assistance. See the ICLR 2027 AI Policy for Authors for the disclosure categories above.

REFERENCES

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 P. Bosma,

Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc 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.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020.

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.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770. Google DeepMind.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *International Conference on Learning Representations*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL https://arxiv.org/abs/1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Gyunyeop Kim and Sangwoo Kang. Exploration-driven reinforcement learning for expert routing improvement in mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23592–23605, Suzhou, China, 2025. Association for Computational Linguistics.

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*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

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.

Lie Li, Wen Li, Junxiao Shen, and Guosheng Hu. MAPLE: Moe adaptive plug-and-play layer-wise expert allocation, 2026. URL https://arxiv.org/abs/2608.15299.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

NLP & AI Lab. KULLM-v2: Korean open instruction dataset. Hugging Face Datasets, 2023. URL https://huggingface.co/datasets/nlpai-lab/KULLM-v2. Korean Alpaca-style open instructions.

NVIDIA. Nemotron-SFT-Multilingual-v2. Hugging Face Datasets, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-SFT-Multilingual-v2. CC BY 4.0 subsets used (KO code/stem); CC BY-SA math excluded.

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

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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 *First Conference on Language Modeling (COLM)*, 2024.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Zunhai Su, Qingyuan Li, Hao Zhang, Weihao Ye, Qibo Xue, Yulei Qian, Yuchen Xie, Ngai Wong, and Kehong Yuan. Unveiling super experts in mixture-of-experts large language models. In *International Conference on Learning Representations*, 2026.

taejoon89. Ko-Agent-Trajectories-1.0. Hugging Face Datasets, 2024. URL https://huggingface.co/datasets/taejoon89/Ko-Agent-Trajectories-1.0. CC BY 4.0; Korean multi-turn agent trajectories.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024.

Juntong Wu, Jialiang Cheng, Qishen Yin, Yue Dai, Yuliang Yan, Fuyu Lv, Ou Dan, and Li Yuan. BEAM: Binary expert activation masking for dynamic routing in moe, 2026. URL https://arxiv.org/abs/2605.14438.

Haoqi Yang, Luohe Shi, Qiwei Li, Zuchao Li, Ping Wang, Hao Huang, and Hai Zhao. Faster moe LLM inference for extremely large models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43133–43151, San Diego, California, United States, 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.findings-acl.2140/. Introduces ASET (Adaptive Skipping with Entropy-Penalized Thresholding).

Yuanhang Yang, Shiyi Qi, Wenchao Gu, Chaozheng Wang, Cuiyun Gao, and Zenglin Xu. XMoE: Sparse models with fine-grained and adaptive expert selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11664–11674, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Tongtian Yue, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-k routing: Boosting the efficiency of moe-based llms. In *International Conference on Learning Representations*, 2025a.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Vienna, Austria, 2025b. Association for Computational Linguistics.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. AdaMoE: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6223–6235, Miami, Florida, USA, 2024. Association for Computational Linguistics.

Shuzhang Zhong, Ling Liang, Yuan Wang, Runsheng Wang, Ru Huang, and Meng Li. AdapMoE: Adaptive sensitivity-based expert gating and management for efficient MoE inference. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24)*. Association for Computing Machinery, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022.

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

A Additional Results and Ablations

A.1 Training hyperparameters

Table 6: Training hyperparameters (shared allocator / PPO setup).

Item	Value
Backbone	Gemma 4 26B-A4B IT, frozen (BF16 train)
Trainable	30 linear count heads ($2816 \to 8$; $\sim$0.68M params)
Optimizer	AdamW, learning rate 10^{-3}, effective batch 64
PPO	2 update epochs per rollout buffer; clip ε=0.2
Ada-K credit	Eqs. 6, 9 (`eq9_advantage`); γ=0.99
δ-Credit credit	$A^{\delta}=R^{\delta}$ (no γ broadcast); α=1.0
Router warm-start	$p^{*}=0.23$, 1 CE epoch, then 1 PPO epoch; N=9,000 (Tab. 9)
Teacher	Same weights, forced Top-8, frozen
K / action set	Top-8; $k^{*} \in \{1, \ldots, 8\}$
λ (Ada-K, Table 2)	0.003 (Yue et al., 2025a)
λ (Ada-K Gemma ablation)	0.002 (common λ + common ckpt; App. A.8)
λ (δ-Credit)	0.001 (common λ sweep; App. A.7)
Eval checkpoint	selected training checkpoint (Sec. 5)
Checkpoint warm-up / MA window	5 steps / 10-step MA; floor LM-gap MA$\geq$0 (fallback: App. A.16)
λ sweep	25 steps; score last 20; eligible if window-mean LM-gap MA> -0.05

A.2 Notation

Table 7: Ada-K symbols (Yue et al., ICLR 2025).

Symbol / term	Meaning
allocator / π_{θ_ℓ}, P_{alloc}	per-layer count policy over $\{1, \ldots, K\}$
k^{*}, $\hat{c}_\ell$	sampled expert count (action)
s_ℓ, $h_{t,\ell}$	allocator state (layer hidden)
router / $*$	frozen Top-K ranking; $*$ = forced Top-K baseline
K, L	max routed experts (Gemma K=8); last MoE layer index
R, A_ℓ	Ada-K Eqs. 6, 9: terminal $\log P$ gap; γ-scaled advantage
γ, λ	advantage discount (0.99); weight on $\mathcal{L}_{\text{reg}}$
$\mathcal{L}_{\text{RL}}$, $\mathcal{L}_{\text{reg}}$, $\mathcal{L}$	PPO clip; expected count; $\mathcal{L}_{\text{RL}} + \lambda\mathcal{L}_{\text{reg}}$
P-Warm, p^{*}	supervised nucleus warm-start (Ada-K Eqs. 12–13)

Table 8: δ-Credit symbols (this paper). Ada-K symbols: Tab. 7.

Symbol / term	Meaning
$h_{t,\ell}$, $k^*_{t,\ell}$	state / count (Ada-K s_ℓ, $\hat{c}_\ell$)
T, S	Top-8 teacher; student under sampled k^*
h^{in}, h^{out}	MoE input; post-MoE layer output (Fig. 2)
$\delta_{t,\ell}$	MoE-block increment $h^{\text{out}} - h^{\text{in}}$ (Eq. (6))
$\text{align}_{t,\ell}$	$(\cos(\delta^S, \delta^T) + 1)/2 \in [0, 1]$ (Eq. (6))
$\mathcal{B}$, α	rollout batch for mean-centering; reward scale (α=1)
$R^\delta_{t,\ell}$, $A^\delta_{t,\ell}$	dense reward; advantage (A^δ=R^δ; no γ broadcast; Eq. (7))
ε	PPO clip (0.2; same surrogate form as Ada-K Eqs. 7–8)

A.3 Training data

Table 9: Allocator training mix (N=9,000; text/agent only). Public ≈89%; internal ≈11% (not redistributed). Sources are Korean–English instruction/agent and code/STEM pools (code and STEM prompts are often English-heavy).

Pool	n	Share	Modality	Availability
Multi-turn agent (tools)	4,000	44.4%	agent	public
Code + STEM instruction	2,000	22.2%	text	public
Open instruction	1,959	21.8%	text	public
Domain-specific agent Q&A	1,041	11.6%	agent	internal

Tab. 9 lists the N=9,000 mix used for router warm-start and PPO (text / agent trajectories; no images). Public pools (≈89%, subsampled with a fixed seed): multi-turn agent trajectories with tools (taejoon89, 2024) (CC BY 4.0), Nemotron KO code/stem rows (NVIDIA, 2025) (CC BY 4.0; CC BY-SA math excluded), and open instructions (NLP & AI Lab, 2023). The mix is bilingual in practice (Korean dialogue/instruction plus English-heavy code/STEM text). One **internal** slice (≈11%) is **not** released: domain-specific multi-turn agent Q&A serialized from a proprietary tool-using workflow. The internal vision–ontology relationship / affordance caption slice (n=1,000) used only in App. A.10 is likewise not redistributed. Table 2 benchmarks use only public evaluation sets (Rein et al., 2024; Wang et al., 2024; Yue et al., 2025b).

A.4 δ vs. h: probe penalty, then same-λ learning

Read this appendix in two steps matching Sec. 4: first the frozen probe (which feature's similarity falls faster when k^* is cut), then the matched PPO logs (how that gap resists—or fails to resist—$\lambda\mathcal{L}_{\text{reg}}$). Do not treat larger motion in the train curves as higher sensitivity.

Step 1 — Frozen probe (count sensitivity). Forward-only probe on a held-out text batch: student forced $k^* \in \{1, \ldots, 8\}$ vs. Top-8 teacher; mean over tokens/layers/batches of $\cos(h^S, h^T)$ and $\cos(\delta^S, \delta^T)$ with δ=h^{out}−h^{in} (shared+routed; no shared subtraction). Both fall as k^* decreases (e.g. at $k^* = 1$: h 0.765, δ 0.629; at $k^* = 4$: h 0.935, δ 0.891), but δ's mean drop-per-k is about 1.58× that of h (Fig. 3). So MoE-δ reacts more to reducing the expert count; full-h cosine under-reacts because residual mass dilutes the routed update. Around the learned mean counts in Table 2 ($k^* \in \{5, 6\}$: δ cosine ≈0.921 / 0.945), the representation remains close to the Top-8 teacher while retaining more k^*-slope than h.

Step 2 — Matched PPO (same λ; credit feature only). The probe does not train an allocator. To see the learning consequence we rerun Gemma PPO with λ=0.001, the same P-Warm init, mix, and PPO hyperparameters as Table 2's δ-Credit, changing only the cosine feature from MoE-δ to full layer-output h. Fig. 6 and Tab. 10 summarize the intended reading:

- **MoE-δ:** faster drop under smaller k^* means the PPO reward punishes cutting count, so the same $\lambda\mathcal{L}_{\text{reg}}$ is resisted: LM-gap MA stays near zero at the selected checkpoint (activation MA ≈5.63; dense align log ≈0.94).

- **Full-h:** under-reacts, so in-reward align stays high ($\approx$0.95) even as mean activated count collapses (late activation MA $\approx$3.46) and the LM-gap monitor fails (late ≈ -0.47)—$\lambda\mathcal{L}_{\text{reg}}$ wins and drives a more extreme cut. Under the common checkpoint rule the selected activation MA is still much lower than matched δ-Credit ($\approx$4.97 vs. $\approx$5.63).
- **Do not misread the figure:** the full-h run's larger activation/LM motion is this weak-credit failure (collapse under λ), not evidence that h is more sensitive; probe sensitivity is Step 1 / Fig. 3.
- **Downstream:** under the same reasoning-mode stack as Table 2, selected full-h scores 65.2% GPQA Diamond at $k^* \approx 4.61$ vs. 82.8% / 5.56 for matched δ-Credit (Tab. 11).

This contrast motivates keeping MoE-δ in R^{δ}.

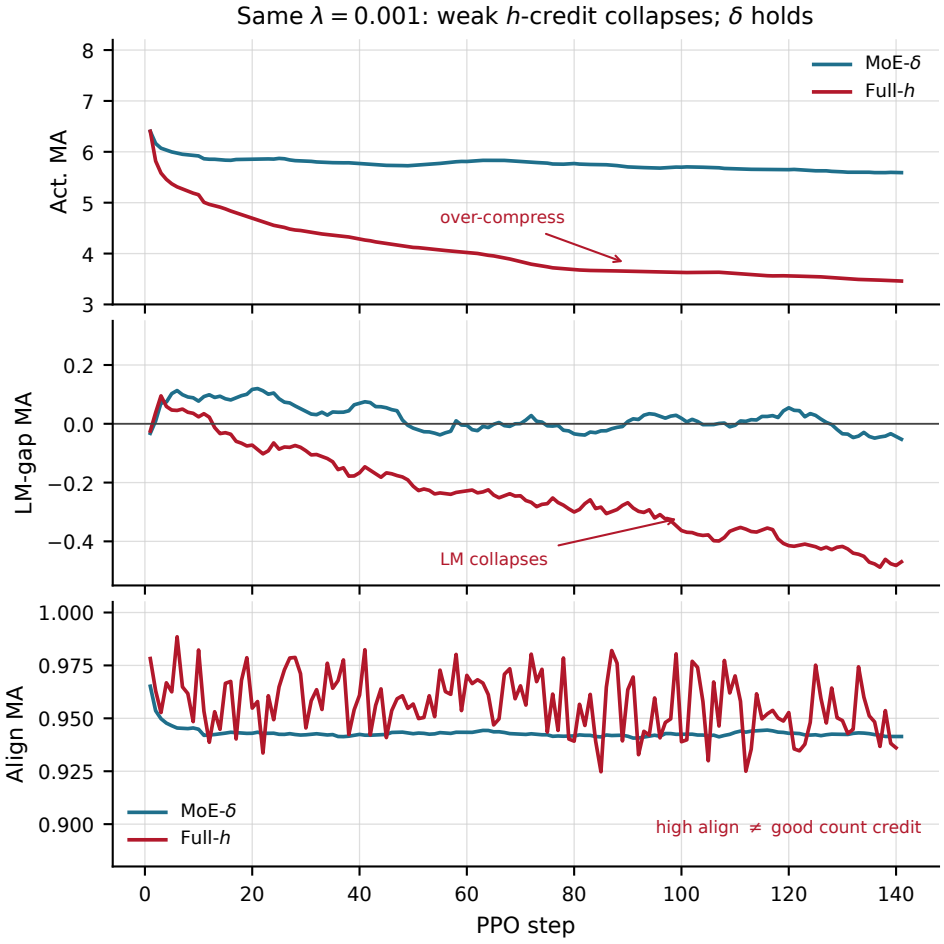


Figure 6: **Learning consequence** of Step 1 under matched PPO (λ=0.001; credit feature only). MoE-δ: reward punishes cutting $k^* \Rightarrow$ resists λ (usable activation / LM-gap). Full-h: high align despite over-compression $\Rightarrow$ λ wins (activation↓, LM-gap↓). Larger full-h motion is that weak-credit failure—not higher sensitivity (Fig. 3). Selected checkpoints: Tab. 10 (common checkpoint rule; Sec. 5).

Table 10: Matched PPO credit ablation (train logs; λ=0.001). Selected under the *common* checkpoint rule (Sec. 5): LM-gap MA$\geq$0 then min activation MA (both methods qualify). Late = last logged full-h step (collapse regime).

Credit feature	Act. MA	LM-gap MA	Align MA
MoE-δ (selected)	5.63	+0.005	$\approx$0.94†
Full-h (selected)	4.97	+0.023	0.939
Full-h (late)	3.46	$-$0.470	$\approx$0.95

$^\dagger\delta$ align MA from a matched dense-align log at the selected checkpoint (activation MA $\geq$5.46; Fig. 8). Late full-h: align remains high while activation and LM-gap collapse—the weak-count-credit failure predicted by Step 1.

Table 11: Matched PPO credit ablation on GPQA Diamond (reasoning-mode decoding; same λ=0.001, stack as Tab. 2). Only the cosine feature differs: full-h vs. MoE-δ (Diamond request-mean k^* 5.56). Full-h k^*: GPQA request-mean from decode count logs (n=198).

Method	Credit	GPQA (%)	k^*
Full-h	$\cos(h^S, h^T)$	65.2	4.61
δ-Credit (ours)	$\cos(\delta^S, \delta^T)$	82.8	5.56

Weak h-credit under-cuts accuracy while compressing harder than matched δ-Credit—the learning failure in Fig. 6 is not rescued at eval.

A.5 Hybrid A_ℓ+R^δ

Ada-K's A_ℓ (terminal $\log P$; Eq. (4)) and R^δ (per-depth MoE-δ align; Eq. (7)) score different things. We tested whether a linear mix still helps:

$$R^{\text{hyb}}_{t,\ell} = \beta\, A_\ell + \alpha\big(\text{align}_{t,\ell} - \text{mean}_{\mathcal{B},\ell}(\text{align})\big), \tag{9}$$

with α=0.2, β=1.0, γ=0.99, and λ=0.009. On GPQA Diamond the hybrid collapses relative to pure δ-Credit (Tab. 12): the two credit axes do not combine usefully, so the main method keeps R^δ alone and does not add Ada-K's R/A_ℓ into the reward.

Table 12: Hybrid A_ℓ+R^δ (Eq. (9)) vs. δ-Credit on GPQA Diamond (selected training checkpoint; not in Table 2). Mixing the two reward axes hurts.

Setting	GPQA (%)	$\bar{k}^*$ (approx.)	vs. δ-Credit
Hybrid A_ℓ+R^δ	52.0	$\approx$6+	$-$30.8 pp
δ-Credit	82.8	5.56†	—

†GPQA Diamond per-request mean routed-expert count (Diamond-only ablation; Table 2's 5.60 is the three-benchmark mean, and Diamond alone is 5.56).

A.6 Router warm-start per-layer profile

At $p^* = 0.23$, the mean nucleus width before clamping to K is $\mathbb{E}[n(p^*)] = 8.03$, and the clamped CE label mean is 6.58 (calibration $\sim$1.28M token–layer pairs). Per-layer P-Warm vs. learned k^* is Fig. 5 (main); early/late layers need fewer experts than mid-depth (Tab. 4). Tab. 13 splits those bands more finely.

Table 13: Fine 5-layer depth bands (complement to Tab. 4): P-Warm labels vs. learned $\bar{k}^*$ at selected training checkpoint.

Depth band	Layers	P-Warm $\bar{n}$	Ada-K $\bar{k}^*$	δ-Credit $\bar{k}^*$
Shallow	L0–L4	5.80	5.13	2.86
Lower-mid	L5–L9	7.16	7.10	6.26
Upper-mid	L10–L14	7.66	7.54	7.18
Lower-late	L15–L19	7.24	7.25	6.22
Upper-late	L20–L24	6.01	5.58	6.38
Tail	L25–L29	5.61	5.11	4.69

A.7 LAMBDA SWEEP (SHORT PPO)

Table 14: Per-method λ from the short PPO sweep (Sec. 5): **common λ rule** on every method/model—25-step run, score the last 20 steps; eligible if the window mean of 10-step LM-gap MA exceeds -0.05, then lowest activation MA. If no candidate clears that gate, take the λ with the highest window-mean LM-gap MA (λ fallback; Qwen Ada-K below). Table 2 reports Gemma Ada-K at the paper value λ=0.003 (common-λ pick 0.002 under the common checkpoint rule: App. A.8). Qwen Tab. 5 reports δ-Credit at the common-rule pick λ=0.006 and a λ=0.007 higher-λ ablation (not eligible under the -0.05 gate); Ada-K paper λ=0.003 is in that table, and Ada-K rule pick 0.002 is App. A.11.

Model	Method	Selected λ	Act. MA
Gemma 4	δ-Credit	0.001	6.07
Gemma 4	Ada-K (R/A_ℓ)	0.002	6.50
Qwen3-30B-A3B	δ-Credit	0.006	6.12
Qwen3-30B-A3B	Ada-K (R/A_ℓ)	0.002	6.19

Gemma δ-Credit / Ada-K and Qwen δ-Credit in Table 14 are eligible under the -0.05 gate (then min activation). Qwen Ada-K never clears that gate on the sweep window, so the λ fallback selects the highest window-mean LM-gap MA (λ=0.002).

A.8 ADA-K λ=0.002 ON GEMMA (ABLATION)

Besides Table 2 (λ=0.003, Ada-K paper (Yue et al., 2025a)), we report the full Gemma run at our **common λ rule** pick λ=0.002 (Sec. 5; App. A.7), using the same depth-discounted advantage as the main Ada-K baseline and the same **common checkpoint rule** as Table 2 (LM-gap MA$\geq$0, then min activation MA; Sec. 5).

Table 15: Ada-K on Gemma at λ=0.002 (common λ pick; common checkpoint rule; not Tab. 2, which uses Ada-K paper λ=0.003).

Setting	$\bar{k}^*$	Active	GPQA	MMLU-Pro	MMMU-Pro
λ=0.002 (common rules)	6.35	3.51B	80.8	82.1	62.3

Per-benchmark k^*: $6.25/6.36/6.44$. Still below Table 2 δ-Credit on every bench ($80.8/82.1/62.3$ vs. $82.8/83.7/63.1$) at higher $\bar{k}^*$ (6.35 vs. 5.60). Main-text McNemar remains vs. Ada-K λ=0.003 only.

A.9 UNIFORM-K ABLATION

Frozen constant-k^* allocators (no PPO; same 64k reasoning-mode BF16 stack as Table 2). This control asks whether Table 2 is just a lower frozen count. Active from Eq. (1) at $\bar{k}^* = K$. GPQA column = GPQA Diamond (n=198). Trained rows repeat Table 2. Matched wall-clock latency: App. A.12. Versus Uniform-K=4, δ-Credit is ahead on all three benches; versus K=5, MMLU-Pro is on par (83.7 vs. 83.4) and GPQA Diamond is ahead (MMMU-Pro ties at 63.1). Versus K=6, GPQA Diamond is meaningfully ahead (82.8 vs. 80.8) and MMLU-Pro is essentially tied (83.7 vs. 84.0); Uniform-K meaningfully wins only on vision Q&A (MMMU-Pro 65.0 vs. 63.1).

Table 16: Frozen uniform-K control (no allocator training).

Method	$\bar{k}^*$ / K	Active	GPQA	MMLU-Pro	MMMU-Pro
BF16 Top-8	8.00	3.81B	83.8	83.7	66.6
Uniform-K (no train)	6	3.45B	80.8	84.0	65.0
Uniform-K (no train)	5	3.27B	78.8	83.4	63.1
Uniform-K (no train)	4	3.10B	73.7	82.1	61.6
Ada-K (λ=0.003)	6.15	3.48B	76.3	82.0	61.7
δ-Credit	5.60	3.38B	82.8	83.7	63.1

A.10 VISION-MIX ABLATION (+1,000 IMAGE CAPTIONS)

Table 2 trains on the N=9,000 text/agent mix (Tab. 9). To test whether the MMMU-Pro gap to Top-8 is mainly missing image supervision, we reran Ada-K and δ-Credit under the same 64k eval / PPO recipe after adding the internal vision–ontology relationship / affordance caption slice (n=1,000; total N=10,000).

Table 17: Same protocol as Table 2; vision slice added (n=1,000). Not a main claim; only probes MMMU-Pro recovery.

Method	$\bar{k}^*$	Active	GPQA	MMLU-Pro	MMMU-Pro
BF16 Top-8 (default)	8.00	3.81B	83.8	83.7	66.6
Ada-K + vision mix	5.88	3.43B	78.8	82.6	61.7
δ-Credit + vision mix	5.73	3.40B	80.8	83.6	63.1
δ-Credit (Table 2)	5.60	3.38B	82.8	83.7	63.1

δ-Credit's MMMU-Pro remains **63.1%** with or without the 1,000 captions (still $-$**3.5** percentage points vs. Top-8); GPQA is slightly worse than the text-only main run. The 1,000 vision slice therefore does *not* close the multimodal recovery gap, though δ-Credit still beats Ada-K on MMMU-Pro under that mix.

A.11 TRANSFER BEYOND SHARED-EXPERT MOES (QWEN3-30B-A3B)

Unlike Gemma 4, Qwen3-30B-A3B (Qwen Team, 2025) has no shared expert (routed-only MoE; 48 MoE layers, top-8 / 128 experts; text-only). We apply the same $\delta{=}h_{\text{out}}{-}h_{\text{in}}$ capture, mean-centered R^{δ}, and PPO+$\lambda\mathcal{L}_{\text{reg}}$ recipe (Sec. 4) as a transfer check to another MoE layout. Eval follows the Qwen model card's reasoning settings (T=0.6, top-p=0.95, top-k=20), BF16 llama.cpp inference, and the same GPQA Diamond / MMLU-Pro clients as Gemma. Short λ sweeps under the **common λ rule** (Sec. 5; App. A.7) pick Ada-K λ=0.002 (LM-gap fallback: no sweep candidate exceeds -0.05) and δ-Credit λ=0.006 (eligible, then min activation). Tab. 5 reports Ada-K at the paper λ=0.003, δ-Credit at 0.006, and a δ λ=0.007 higher-λ ablation (not eligible under the -0.05 gate); Ada-K at our rule pick 0.002 is Tab. 18 below. Full-run checkpoints use the **common checkpoint rule** (Sec. 5); when no post-warmup step clears LM-gap MA$\geq$0, the checkpoint fallback $(\text{LM-gap MA}) - (\text{activation MA})/K$ applies (as for Qwen Ada-K). Against Ada-K paper λ=0.003, δ 0.006 is near cost (80% vs. 78%) at much higher accuracy, and δ 0.007 further cuts cost to 72% while still beating Ada-K. Against Ada-K under our λ rule (0.002; matched 80% cost), δ 0.006 still leads on both benches. Paired McNemar (reasoning mode; same clients as Gemma): vs. Ada-K 0.003, GPQA Diamond n=198 is 41 vs. 15 ($p{\approx}0.0007$) and MMLU-Pro n=12,032 is 1492 vs. 559 ($p{<}0.001$); vs. Ada-K 0.002 (matched 80% cost), 35 vs. 15 ($p{\approx}0.007$) and 968 vs. 578 ($p{<}0.001$); at the δ λ=0.007 ablation vs. Ada-K 0.003, 36 vs. 16 ($p{\approx}0.008$) and 1460 vs. 604 ($p{<}0.001$).

Table 18: Qwen Ada-K under the common λ rule (λ=0.002; not in Tab. 5). Same checkpoint rule and eval stack; Cost matches δ 0.006 at 80% of Top-8.

Method	λ	GPQA	MMLU-Pro	k^*_{G}	k^*_{M}	Active	Cost
Ada-K (R/A_ℓ)	0.002	49.0	72.6	6.43	6.40	3.00B	80%
δ-Credit (Tab. 5)	0.006	59.1	75.8	6.51	6.29	2.99B	80%

Qwen3-30B-A3B has no shared expert (d=2048, L=48, SwiGLU expert width 768), so Active fixes non-MoE and scales the full Top-8 routed MoE block by mean $k/8$:

$$\text{active}_{\text{Qwen}}(k) \approx 1.54 + 1.81 \cdot (k/8) \quad [\text{B}], \tag{10}$$

with k the mean of reported request-mean k^* in Tab. 5 (or Tab. 18) and Cost = $k/8$. Top-8 is ≈3.35B (card: ∼3.3B activated). We do not report a matched tokens/s table on Qwen (Gemma: App. A.12).

A.12 MATCHED WALL-CLOCK LATENCY (GEMMA)

Routed MoE cost (Tab. 3) is not wall-clock. On matched GPQA Diamond prompts (n=20; fixed completion lengths 512 / 4096; Tab. 19), lower decode-time $\bar{k}^*$ is associated with higher tokens/s:

δ-Credit 122.5 / 124.3 vs. Top-8 115.7 / 117.4 ($+5.9\%$ / $+5.9\%$), with Ada-K in between. A $\sim$30% routed-MoE FLOP cut yields only single-digit wall speedup (attention, KV, and the always-active shared expert dominate). Measurements are single-request llama.cpp decode, not batched serving.

Table 19: Matched decode on GPQA Diamond (n=20; lengths 512 / 4096). $\bar{k}^*$ over tokens×layers; $\Delta\%$ vs. Top-8.

Method	$\bar{k}^*$ (512)	tokens/s (512)	$\bar{k}^*$ (4096)	tokens/s (4096)	$\Delta\%$ (512 / 4096)
Top-8	8.00	115.7	8.00	117.4	—
Ada-K	5.95	121.9	6.34	122.0	$+5.4$ / $+3.9$
δ-Credit	5.56	122.5	5.46	124.3	$+5.9$ / $+5.9$

A.13 MoE FLOP counting

Routed MoE feed-forward cost is proportional to $\bar{k}^*$, so Table 3 reports $\bar{k}^*/8$ vs. Top-8. For absolute numbers: with d=2816, routed width 704, shared width 2112, and L=30, a SwiGLU expert costs $6\,d\,n_{\text{ff}}$ FLOPs (gate/up/down, mul+add), hence

$$\text{FLOPs}_{\text{routed}}(\bar{k}^*) = 30 \cdot \bar{k}^* \cdot 6\,d \cdot 704 \tag{11}$$

($\approx$2.85 / 2.19 / 2.00 GFLOP per token at $\bar{k}^* \in \{8, 6.15, 5.60\}$ for Top-8 / Ada-K / δ-Credit), plus a fixed shared term $30 \cdot 6\,d \cdot 2112$. Attention and KV are unchanged by k^*.

A.14 PPO clip details

Sec. 4 states the update; here we expand token/layer averages. We take from Ada-K only the allocator PPO clip (Ada-K Eqs. 7–8) and the compression term $\lambda\mathcal{L}_{\text{reg}}$ in Eq. (5), not their reward Eqs. 6, 9. δ-Credit plugs $A{=}A^{\delta}_{t,\ell} = R^{\delta}_{t,\ell}$ (Eq. (7)) into that clip; the Ada-K baseline reproduction uses $A{=}A_\ell$ (Eq. (4)) for matched comparison. With importance ratio ρ and clip ε=0.2:

$$\ell^{\text{clip}}_{b,t,\ell} = -\min\big(\rho_{b,t,\ell}A_{b,t,\ell},\ \text{clip}(\rho_{b,t,\ell}, 1{-}\varepsilon, 1{+}\varepsilon)\,A_{b,t,\ell}\big), \tag{12}$$

$$L^{\text{clip}}_{\ell} = \frac{\sum_{b,t} m_{b,t}\,\ell^{\text{clip}}_{b,t,\ell}}{\sum_{b,t} m_{b,t}}, \qquad \mathcal{L}_{\text{RL}} = \frac{1}{L}\sum_{\ell} L^{\text{clip}}_{\ell}, \tag{13}$$

where $m_{b,t}$ masks pads. $\mathcal{L}_{\text{reg}}$ is the expected-count term in Eq. (5) (pad-masked mean over the rollout). Rollouts fill a buffer of 64 sequences; each time the buffer is full, we run 2 PPO epochs over $(h, k^*, \log\pi_{\text{old}}, R)$.

A.15 Implementation note

Training uses PyTorch with a frozen Gemma 4 backbone; Table 2 inference uses BF16 llama.cpp with dynamic k^* masking after Top-8 router selection (unused expert FFN slots skip compute).

A.16 Checkpoint selection protocol

λ is chosen first under the common λ rule (App. A.7). Each full run then contributes its **selected training checkpoint** to Table 2 under one **common checkpoint rule** (not an independent held-out validation split, and not a protocol from Yue et al., who report one epoch without an LM-gap gate). In this appendix, **MA** denotes a rolling **10-step moving average**. After a 5-step warm-up, a PPO step **qualifies** once the 10-step MA of reward quality is defined and satisfies LM-gap MA$\geq$0; among qualifying steps we keep the **lowest** activation moving average (mean $\bar{k}^*$ over tokens and layers in the rollout). Both methods use quality $= R(\hat{c}_L, s_L) - R(c^*_L, s^*_L)$ (Ada-K PPO signal; δ-Credit logged monitor only, not in R^{δ}). Dense R^{δ} is the per-layer PPO credit (trajectory MoE-update scores); the LM-gap is only the gate in this common rule. Under the rule, Gemma Ada-K (λ=0.003) has a single post-warmup qualifier (LM-gap MA$\geq$0), which is the Table 2 Ada-K checkpoint; δ-Credit has many qualifiers and selects a lower activation MA. If no step clears LM-gap MA$\geq$0, we take the post-warmup maximizer of (LM-gap MA) $-$ (activation MA)$/K$ (ties: lower activation MA; used on Qwen Ada-K when unmet). **Ablation (not Table 2):** relaxing Ada-K only to LM-gap

MA≥ -0.05 admits later compressed steps on the *same* Ada-K run. That pick improves over the common-rule Ada-K checkpoint on all three benches (GPQA Diamond / MMLU-Pro / MMMU-Pro $79.8/83.1/62.5$ vs. $76.3/82.0/61.7$; $\bar{k}^*\approx 6.00$ vs. 6.15 with per-benchmark $k^*\approx 5.87/6.03/6.10$), but remains below δ-Credit on every bench ($82.8/83.7/63.1$ at $\bar{k}^*=5.60$). **Training stop:** one PPO epoch over $\sim$9,000 trajectories; optional early stop when validation activation stalls (default: full epoch).

A.17 Training-log evidence

Fig. 7 overlays the matched Gemma full runs behind Table 2: Ada-K ($\lambda=0.003$) and δ-Credit ($\lambda=0.001$). After warm-up, Ada-K clears LM-gap MA≥ 0 on only one step (the Table 2 Ada-K pick under the common checkpoint rule), while activation continues to fall; the Ada-only MA≥ -0.05 ablation pick sits just above -0.05 at lower activation than the common-rule Ada checkpoint and is the stronger Ada eval of the two, yet still trails δ-Credit on GPQA / MMLU-Pro / MMMU-Pro. δ-Credit keeps LM-gap MA≥ 0 on a majority of post-warmup steps and reaches a lower selected activation than Ada-K. Separately, within the δ-Credit run, align vs. the Ada-K $R-R^*$ monitor MA has Pearson $r \approx 0.08$ (Figs. 8–10); Fig. 11 shows $\mathcal{L}_{\text{reg}}$ and $\lambda\mathcal{L}_{\text{reg}}$ alongside $\mathcal{L}_{\text{RL}}$.

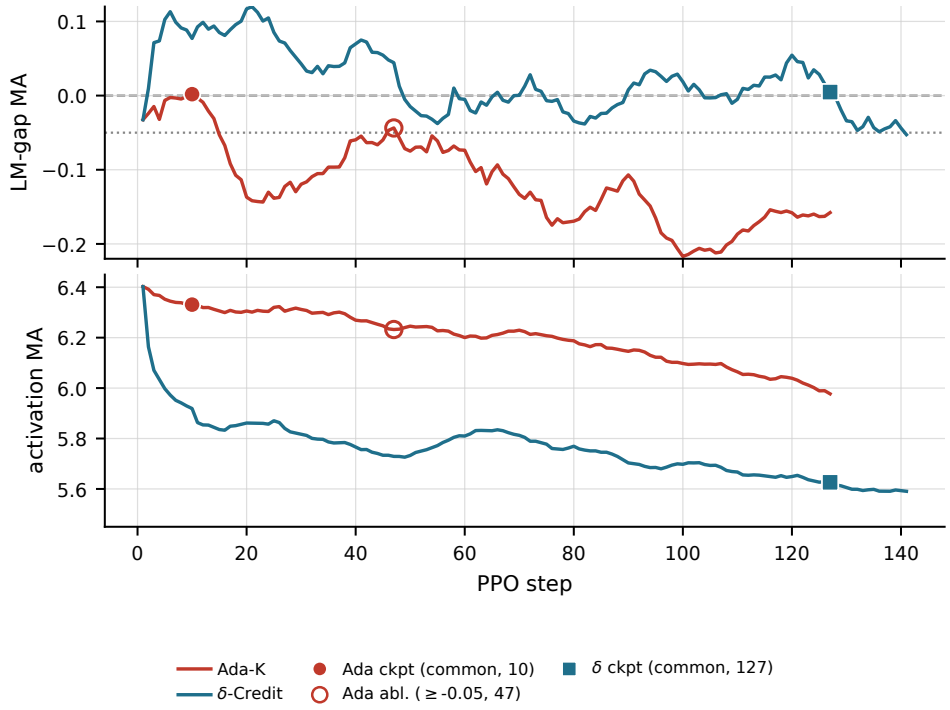


Figure 7: Matched Gemma full runs (10-step MA). **Top:** Top-K LM-gap $R(\hat{c}_L, s_L) - R(c_L^*, s_L^*)$ (Ada-K training signal; δ-Credit monitor only). **Bottom:** activation MA (mean $\bar{k}^*$). Horizontal guides: common-rule threshold LM-gap MA$=0$ and Ada-only ablation threshold -0.05. Markers are *selected eval checkpoints* under the common checkpoint rule (filled: Ada-K / δ-Credit) or the Ada-only MA≥ -0.05 ablation (open)—not early stopping; both runs train a full PPO epoch.

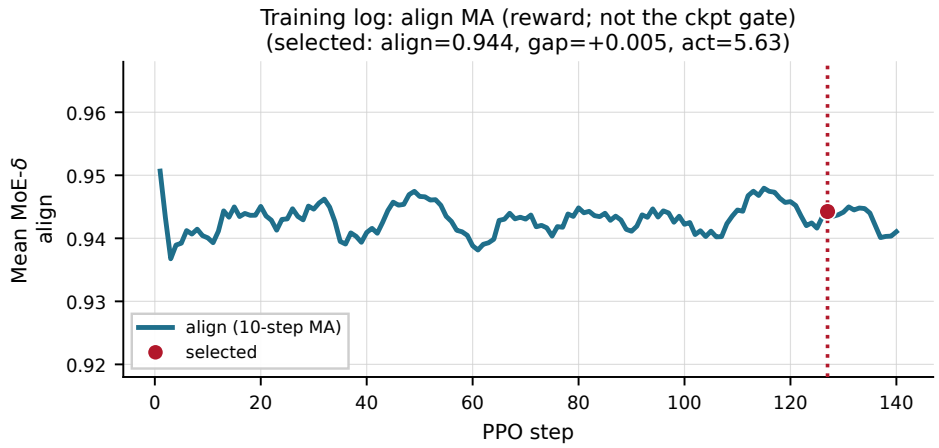


Figure 8: δ-Credit training log (**10-step MA**). Mean MoE-δ align (reward signal; *not* the checkpoint floor). Red = selected checkpoint under LM-gap monitor floor 0 (Sec. 5). Activation MA: Fig. 9; Ada-K monitor: Fig. 10.

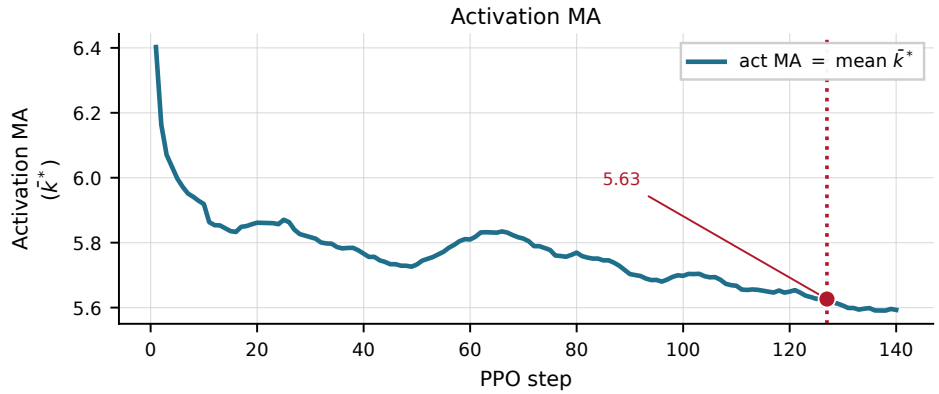


Figure 9: δ-Credit activation MA = mean $\bar{k}^*$ (same run as Fig. 8). Red = selected checkpoint (LM-gap floor 0).

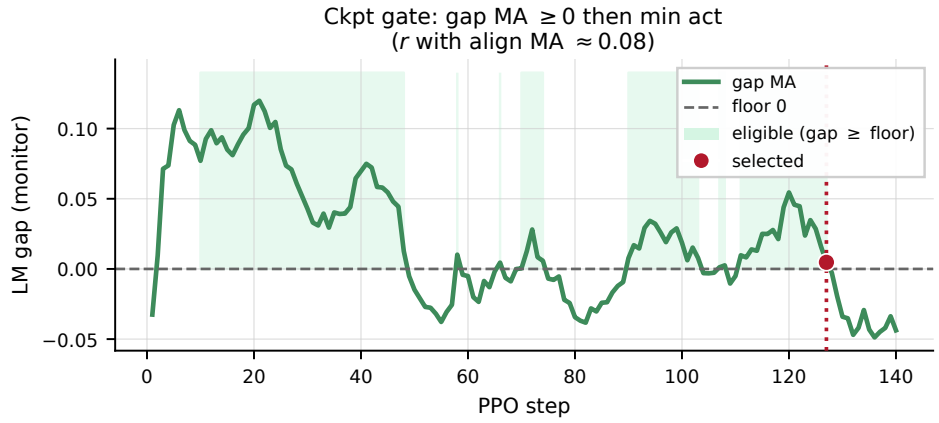


Figure 10: Ada-K $R-R^*$ MA during δ-Credit training ($r \approx 0.08$ with align MA; not in R^δ). Same run as Figs. 8–9; this monitor is δ-Credit's checkpoint quality (floor 0). Per-layer align at the selected checkpoint: Fig. 5 (main).

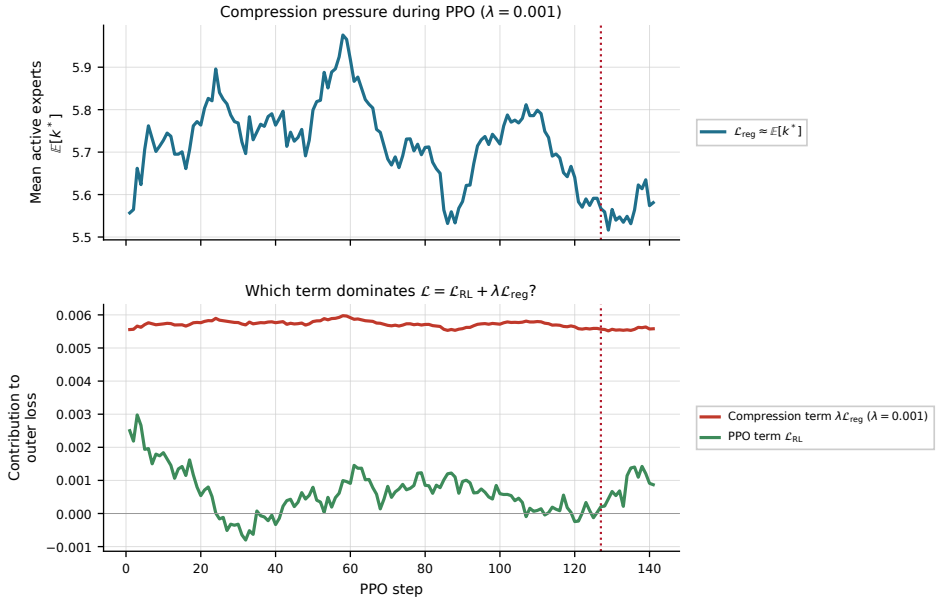


Figure 11: δ-Credit PPO (**10-step MA**). **Top:** $\mathcal{L}_{\text{reg}} \approx \mathbb{E}[k^*]$ (Ada-K Eq. 10), i.e. mean active experts / compression pressure. **Bottom:** same scale for the weighted compression term $\lambda\mathcal{L}_{\text{reg}}$ (λ=0.001) vs. PPO term $\mathcal{L}_{\text{RL}}$ in $\mathcal{L}$=$\mathcal{L}_{\text{RL}} + \lambda\mathcal{L}_{\text{reg}}$ (Ada-K Eq. 11). Dotted red: selected checkpoint.