

Meta-Learning Where to Allocate Experts: Task-Conditioned Layer-Wise Compression for MoEs

Wang Rongfeng^{1,2,3,*} Shichao Weng^{6,*} Zhiqiang Wang^{6,*} Xinyu Liu¹
Yang Yi¹ Peilong Zhou^{1,5} Tang Hongwei^{1,2,3,4,†}

¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

²Nanjing College, University of Chinese Academy of Sciences, Nanjing, China

³Nanjing Institute of Information Superbahn, Nanjing, China

⁴University of Chinese Academy of Sciences, Beijing, China

⁵School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences, Beijing, China

⁶Shenzhen Dobot Corp Ltd, Shenzhen, China

*Equal contribution. †Corresponding author.

Email: tanghongwei@ict.ac.cn

Abstract

Mixture-of-Experts (MoE) models route each token to a subset of expert networks, increasing capacity while keeping per-token computation sparse. Many deployed MoEs use a fixed number of active experts across layers and tasks, despite evidence that layer roles, expert redundancy, and required capacity vary with depth and input difficulty. Existing layer-wise methods usually determine allocations offline and reuse them across tasks, while token-level methods rely on local routing signals without task-level context. We propose MetaNet, a support-set controller that predicts an expert-retention threshold and a bounded routing bias for each layer while keeping the backbone, experts, and original router frozen. On DeepSeek-MoE-16B-Chat, MetaNet provides a tunable accuracy–expert-activation trade-off. Relative to fixed $k=6$, a conservative setting activates 3.61 experts on average (40% fewer) with comparable MMLU accuracy (0.489 versus 0.474); an aggressive setting activates 2.28 experts on average (62% fewer) at 0.438 accuracy. Trained on MMLU, MetaNet transfers to C-Eval without retraining, activating 2.90 experts on average (52% fewer than fixed $k=6$) at 0.386 accuracy.

1 Introduction

Serving large language models (LLMs) at scale is expensive (Touvron et al., 2023; OpenAI, 2023; Ma et al., 2023). Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017; Fedus et al., 2022) reduce per-step compute by routing each token to a small subset of expert subnetworks, enabling large total capacity without activating it all at once. This design is used in recent MoE language models such as DeepSeekMoE (Dai et al., 2024) and

Mixtral (Jiang et al., 2024).

Yet standard pretrained MoE inference typically uses the same number of experts at every layer for every token and every task (Shazeer et al., 2017; Fedus et al., 2022; Dai et al., 2024). This contrasts with evidence that layer-wise capacity demands differ. In dense Transformers, layers have distinct functional roles (Jin et al., 2025; Tenney et al., 2019) and contribute differently to task performance (Men et al., 2025; Lad et al., 2025). In MoE models, lower layers exhibit higher expert redundancy than upper layers (Gao et al., 2025), and heterogeneous cross-layer redundancy has motivated global and domain-specific pruning strategies (Zhang et al., 2026; Yao et al., 2026). Some works exploit depth heterogeneity through layer-wise capacity allocation—OpenELM (Mehta et al., 2024) places fewer parameters in shallow layers, while AlphaLoRA (Qing et al., 2024) and LExI (Chitty-Venkata and Emani, 2026) assign layer-specific expert budgets—but these allocations are fixed offline and shared across tasks. Token-level dynamic routing (Huang et al., 2024; Guo et al., 2025) lets each token activate a variable number of experts, but decisions are made from per-token routing confidence alone, with no access to task context. Expert-pruning methods (Zhang et al., 2026; Yao et al., 2026; Lu et al., 2024; Yang et al., 2024) produce a compressed expert set or skipping policy before query-time inference, rather than inferring a fresh per-layer allocation from a support set at query time. Router-modification or construction methods (Hwang et al., 2024; Do et al., 2023) alter the gating mechanism rather than preserving the original router.

This paper studies task-conditioned layer-wise

expert allocation for frozen MoE inference. Given a small support set for the current task, *MetaNet* predicts a per-layer expert activation policy: a retention threshold $\rho_{\tau,l}$, which determines how many experts are needed to cover the required routing mass at layer l , and a bounded routing bias $\mathbf{b}_{\tau,l}$, whose magnitude is much smaller than typical gate logits and therefore only weakly perturbs the frozen router. The backbone, original router, and full expert set remain unchanged. On DeepSeekMoE-16B-Chat, *MetaNet* reduces the mean number of activated experts by 62% relative to fixed $k=6$ (from 6.00 to 2.28), with MMLU accuracy 0.438 versus 0.474 for fixed $k=6$, and transfers to C-Eval without retraining with a similar reduction in activated experts. With a weaker budget penalty, it keeps 3.61 mean experts, achieves comparable accuracy to fixed $k=6$, and reduces the mean activated-expert count by 40%.

Contributions.

- We formulate *task-conditioned per-layer expert allocation* for frozen MoE inference: infer per-layer active expert counts from a small support set, without modifying backbone parameters.
- We design *MetaNet*, a small two-branch controller that predicts a retention threshold $\rho_{\tau,l}$ and a bounded routing bias $\mathbf{b}_{\tau,l}$ per layer from support-set routing statistics, keeping the backbone, original router, and full expert pool frozen.
- We show through component ablations that the budget and prior branches are complementary: the prior branch helps maintain gate-consistent expert selection under tight budgets, and deep layers are more compressible than shallow layers in our setting.

2 Related Work

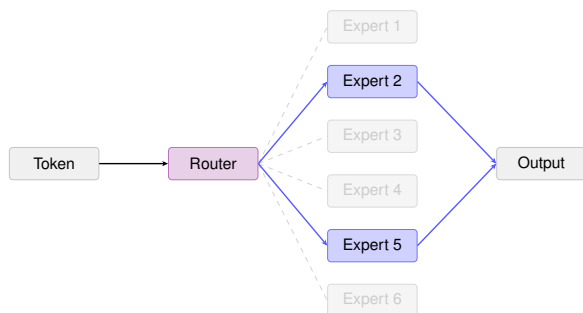


Figure 1: An MoE layer: the router activates 2 of 6 experts per token; inactive experts grayed out.

Mixture-of-Experts and inference compression.

MoE architectures (Shazeer et al., 2017; Fedus et al., 2022) route each input token to k out of E expert FFNs via a gating network (Figure 1), keeping per-step compute proportional to k while allowing large total capacity. DeepSeekMoE (Dai et al., 2024) refines this with fine-grained expert segmentation, and Mixtral (Jiang et al., 2024) shows the effectiveness of sparse expert activation in open LLMs. In these models, however, the active expert count is typically fixed by architecture or inference configuration rather than adapted to the task.

Compressing the inference cost of frozen MoEs.

Existing work on MoE inference compression can be grouped into four lines. (i) *Expert pruning and merging*. NAE (Lu et al., 2024), MoE-I² (Yang et al., 2024), and STUN (Lee et al., 2025) perform post-training expert-level sparsification; MoE-I² additionally applies low-rank decomposition within retained experts. HC-SMoE (Chen et al., 2025) merges experts through hierarchical clustering, whereas ResMoE (Ai et al., 2025) approximates experts using a Wasserstein barycenter and compressed residuals. Most fix policies before deployment, limiting unseen-task adaptation. (ii) *Input-adaptive computation*. Huang et al. (2024), Dynamic MoE (Guo et al., 2025), and AdaMoE (Zeng et al., 2024) vary expert activation using token-level signals. Probe Pruning (Le et al., 2025) is a related batch-wise dynamic structured-pruning method based on model probing rather than MoE router confidence. These approaches make local input-conditioned decisions rather than assigning task-conditioned per-layer capacity from a support set. (iii) *Layer-wise heterogeneous allocation*. MoLA (Gao et al., 2025), AlphaLoRA (Qing et al., 2024), and LExI (Chitty-Venkata and Emani, 2026) assign different expert counts to different layers, but the allocation is determined offline and reused across tasks. (iv) *Router modification or construction*. HyperRouter (Do et al., 2023) and Pre-gated MoE (Hwang et al., 2024) change the routing mechanism, while Read-ME (Cai et al., 2024) refactorizes a pretrained dense LLM into an MoE with a decoupled pre-gating router. *MetaNet* instead keeps the original MoE router frozen and adds only a bounded task-conditioned bias.

Systems support for sparse MoE inference.

Routing sparsity does not by itself guarantee a proportional end-to-end speedup. DeepSpeedMoE (Rajbhandari et al., 2022) uses a specialized

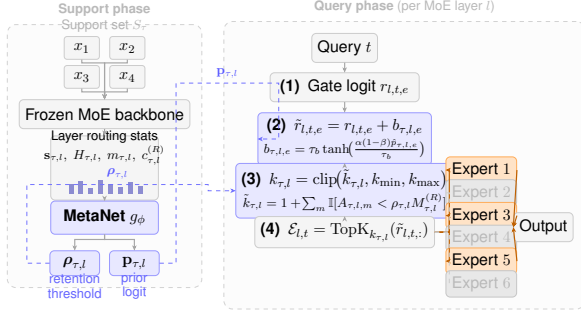


Figure 2: Overview of MetaNet. **Left:** support examples S_τ pass through the frozen backbone; routing statistics are fed to MetaNet g_ϕ , producing a per-layer retention threshold $\rho_{\tau,l}$ and prior score $\mathbf{p}_{\tau,l}$. **Right:** at query time, the corresponding bounded bias is added to gate logits (2); cumulative-mass thresholding with $\rho_{\tau,l}$ determines $k_{\tau,l}$ (3); TopK $_{k_{\tau,l}}$ selects experts (4). Orange: activated; gray: skipped. Backbone parameters are unchanged.

inference stack, Tutel (Hwang et al., 2023) adapts parallelism and pipelining to dynamic expert workloads, and Lina (Li et al., 2023) addresses all-to-all communication and skewed expert popularity. MetaNet changes the routing policy rather than the runtime and is complementary to these system optimizations.

Meta-learning and task-conditioned controllers. MetaNet follows the support-query paradigm (Finn et al., 2017; Vinyals et al., 2016): a controller infers a task representation from a small support set and uses it during query-time inference. Unlike gradient-based model adaptation (Finn et al., 2017), weight-generating hypernetworks (Ha et al., 2017), or parameter-efficient methods that learn task-specific prompt or activation parameters (Liu et al., 2022; Lester et al., 2021), MetaNet performs no task-specific parameter optimization. It maps the support set directly to a routing policy—a per-layer retention threshold ρ_ℓ and bounded bias $\mathbf{b}_\ell$ —that acts on expert selection (Figure 2).

3 Method

MetaNet derives a per-layer expert-activation policy from a support set while keeping the MoE backbone frozen. Figure 2 illustrates the full pipeline.

3.1 Problem setup and support profiling

Let f_θ be a pretrained MoE model with L routed layers, each containing E expert FFNs and a router that activates the top K_{nat} experts per token ($K_{\text{nat}} = 6$ for DeepSeek-MoE-16B-Chat, 8 for OL-

MoE). Following the support-query paradigm (Finn et al., 2017; Vinyals et al., 2016), each task τ has a support set $S_\tau = \{(x_i, y_i)\}_{i=1}^{N_s}$ (used to infer the task policy) and a query set Q_τ (used for evaluation).

MetaNet g_ϕ maps S_τ to a per-layer routing policy $\Pi_\tau = g_\phi(S_\tau)$, replacing the fixed top- K_{nat} with layer-wise activation counts $\mathbf{k}_\tau = (k_{\tau,1}, \dots, k_{\tau,L})$, $k_{\tau,l} \in \{k_{\min}, \dots, k_{\max}\}$. We set $k_{\min} = 1$ and $k_{\max} = 12$ ($= 2K_{\text{nat}}$ for DeepSeek-MoE), so the controller can reduce the budget in redundant layers and retain more experts in layers whose support routing is diffuse.

Let $r_{l,t,e}^{(x)}$ be the gate logit at layer l , expert e , token t of sample x . The per-layer expert score is the token-and-sample average of softmax probabilities:

$$s_{\tau,l,e} = \frac{1}{|S_\tau|} \sum_{x \in S_\tau} \frac{1}{T_x} \sum_{t=1}^{T_x} \text{softmax}(r_{l,t}^{(x)})_e, \quad (1)$$

giving a distribution $\mathbf{s}_{\tau,l} \in \Delta^{E-1}$ that summarizes which experts the task favors at each layer.

Three scalar statistics capture routing concentration and serve as compressibility signals:

$$\begin{aligned} H_{\tau,l} &= -\frac{1}{\log E} \sum_{e=1}^E s_{\tau,l,e} \log s_{\tau,l,e}, \\ m_{\tau,l} &= \max_e s_{\tau,l,e}, \\ c_{\tau,l}^{(R)} &= \sum_{e \in \text{TopKIdx}_R(\mathbf{s}_{\tau,l})} s_{\tau,l,e}. \end{aligned} \quad (2)$$

Here $H_{\tau,l} \in [0, 1]$ is normalized routing entropy ($H = 0$: single expert, $H = 1$: uniform), $m_{\tau,l}$ is the maximum expert probability, $c_{\tau,l}^{(R)}$ is the fraction of routing mass in the top- R experts, and TopKIdx_R returns the index set of top- R entries. Layers with low $H_{\tau,l}$, high $m_{\tau,l}$, and high $c_{\tau,l}^{(R)}$ are concentrated and thus amenable to budget reduction.

The support profile uses only quantities produced by the frozen router and does not require gradients through the backbone during policy inference. The same profile is reused for all query examples from the task, so the profiling cost is amortized over the query set. MetaNet therefore acts as a task-level allocator: it chooses how much expert capacity each layer receives, while the original router still ranks experts for individual tokens.

3.2 MetaNet controller and inference strategy

MetaNet g_ϕ processes the per-layer routing statistics through two independent branches and outputs

two per-layer signals:

$$\begin{aligned}\Pi_\tau &= g_\phi\left(\{s_{\tau,l}, H_{\tau,l}, m_{\tau,l}, c_{\tau,l}^{(R)}\}_{l=1}^L\right) \\ &= \{\rho_{\tau,l} \in (0, 1], \mathbf{p}_{\tau,l} \in \mathbb{R}^E\}_{l=1}^L.\end{aligned}\quad (3)$$

Both branches share a task encoder: a two-layer MLP that maps the support-derived features $[s_{\tau,l}; H_{\tau,l}; m_{\tau,l}; c_{\tau,l}^{(R)}] \in \mathbb{R}^{E+3}$ to a task representation $\mathbf{z}_\tau \in \mathbb{R}^d$; each layer has a learned embedding $\mathbf{e}_l \in \mathbb{R}^d$ ($d=2048$). The **budget branch** encodes $s_{\tau,l}$ and the scalar statistics via two-layer MLPs, concatenates with $[\mathbf{z}_\tau; \mathbf{e}_l]$, and produces discrete budget logits whose soft weighted sum gives the **retention threshold** $\rho_{\tau,l}$. The **prior branch** uses only the scalar-statistics projection (not $s_{\tau,l}$, to avoid a trivial copy) concatenated with $[\mathbf{z}_\tau; \mathbf{e}_l]$ to produce a per-expert score $\mathbf{p}_{\tau,l}$, mapped to the **bounded routing bias** $\mathbf{b}_{\tau,l}$ via scaled tanh.

When multiple experts have similar gate scores, top- k selection is sensitive to small perturbations. The role of $\mathbf{p}_{\tau,l}$ is to provide a weak task-level preference among such experts. We standardize $\mathbf{p}_{\tau,l}$ within each layer ($\hat{p}_{\tau,l,e} = (p_{\tau,l,e} - \mu)/(\sigma + \varepsilon)$) and map it to a bounded routing bias:

$$b_{\tau,l,e} = \tau_b \tanh\left(\frac{\alpha(1 - \beta)\hat{p}_{\tau,l,e}}{\tau_b}\right), \quad (4)$$

where $\tau_b > 0$ is the maximum bias magnitude, $\alpha \in (0, 1)$ is a global scale, and $\beta \in [0, 1)$ is a shrinkage coefficient. Because $|b_{\tau,l,e}| \leq \tau_b$ and τ_b is much smaller than typical gate logit magnitudes, the pretrained router remains the main source of expert ordering. In our main configuration $\alpha(1 - \beta) = 0.005$.

Rather than learning $k_{\tau,l}$ directly (which entangles the budget decision with the shape of the routing distribution), we reformulate budget learning in terms of cumulative probability mass. We combine the support routing distribution with the bounded bias to obtain an adjusted distribution $\bar{\mathbf{q}}_{\tau,l} = \text{softmax}(\log \mathbf{s}_{\tau,l} + \mathbf{b}_{\tau,l})$, used only for budget computation (not for query-phase routing). Here $\log \mathbf{s}_{\tau,l}$ converts the probability vector back to an unnormalized score before adding the bias. Let $M_{\tau,l}^{(R)} = \sum_{e \in \text{TopKIdx}_R(\bar{\mathbf{q}})} \bar{q}_{\tau,l,e}$ be the routing mass covered by the top- R reference experts (here $R = k_{\max} = 12$, matching the reference budget). Let $\bar{q}_{\tau,l,j}^\downarrow$ denote the j -th largest entry of $\bar{\mathbf{q}}_{\tau,l}$ and $A_{\tau,l,m} = \sum_{j=1}^m \bar{q}_{\tau,l,j}^\downarrow$ its prefix sum. The hard acti-

vation count is:

$$\begin{aligned}\tilde{k}_{\tau,l} &= 1 + \sum_{m=1}^E \mathbb{I}[A_{\tau,l,m} < \rho_{\tau,l} M_{\tau,l}^{(R)}], \\ k_{\tau,l} &= \text{clip}(\tilde{k}_{\tau,l}, k_{\min}, k_{\max}).\end{aligned}\quad (5)$$

Intuitively, layers with concentrated routing (low $H_{\tau,l}$) reach any given mass threshold with fewer experts and receive a smaller budget. For training, the non-differentiable indicator is replaced by a sigmoid with temperature T_{bud} (Maddison et al., 2017; Jang et al., 2017; Louizos et al., 2018):

$$k_{\tau,l}^{\text{soft}} = 1 + \sum_{m=1}^E \sigma\left(\frac{\rho_{\tau,l} M_{\tau,l}^{(R)} - A_{\tau,l,m}}{T_{\text{bud}}}\right). \quad (6)$$

At query time, the task-level bias is added to the original gate logit and top- $k_{\tau,l}$ selection is applied:

$$\begin{aligned}\tilde{r}_{\tau,l,t,e} &= r_{l,t,e} + b_{\tau,l,e}, \\ \mathcal{E}_{\tau,l,t} &= \text{TopKIdx}_{k_{\tau,l}}(\tilde{r}_{\tau,l,t,:}).\end{aligned}\quad (7)$$

The policy $\{\rho_{\tau,l}, \mathbf{b}_{\tau,l}\}_{l=1}^L$ is computed once from S_τ and reused across all queries; no backbone parameter is modified.

The formulation separates two decisions. The retention threshold controls *how many* experts are active in a layer, whereas the bounded bias weakly influences *which* experts are preferred when gate scores are close. Fixed top- k baselines keep the first decision constant for all layers and tasks. Token-adaptive methods change k independently for each token, but they do not use a task-level support set. MetaNet keeps the pretrained token router as the fine-grained selector while letting the task profile choose a layer-wise budget before query inference begins.

3.3 Training objective

Training follows an episodic support-query protocol (Finn et al., 2017; Vinyals et al., 2016): each episode samples a task τ , generates Π_τ from S_τ , and evaluates all loss terms on Q_τ . Only MetaNet parameters ϕ are updated. The total loss is:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \lambda_{\text{task}} \mathcal{L}_{\text{task}} + \lambda_{\text{pp}} \mathcal{L}_{\text{pp}} + \lambda_{\text{budget}} \mathcal{L}_{\text{budget}} \\ &\quad + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} \\ &\quad + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}.\end{aligned}\quad (8)$$

Task loss $\mathcal{L}_{\text{task}}$. Standard cross-entropy on Q_τ under the dynamic routing policy of Eq. (7):

$$\mathcal{L}_{\text{task}} = -\frac{1}{|Q_\tau|} \sum_{(x,y) \in Q_\tau} \log p(y | x; \Pi_\tau). \quad (9)$$

Performance-preserving penalty $\mathcal{L}_{\text{pp}}$. We compute a detached reference loss $\mathcal{L}_{\text{ref}}$ by evaluating Q_τ under the original router at native top- K_{nat} . Let $\Delta = \mathcal{L}_{\text{task}} - \mathcal{L}_{\text{ref}} - m_{\text{pp}}$, where $m_{\text{pp}} \geq 0$ is a tolerance margin. The penalty is a one-sided Huber loss: $\mathcal{L}_{\text{pp}} = 0$ when $\Delta \leq 0$, and

$$\mathcal{L}_{\text{pp}} = \begin{cases} \Delta^2/(2\delta_H) & \text{if } 0 < \Delta \leq \delta_H, \\ \Delta - \delta_H/2 & \text{if } \Delta > \delta_H, \end{cases} \quad (10)$$

where $\delta_H > 0$ is the Huber transition point (quadratic below δ_H , linear above).

Budget loss $\mathcal{L}_{\text{budget}}$. Let $\bar{k}_{\tau,l} = (k_{\tau,l}^{\text{soft}} - k_{\min})/(k_{\max} - k_{\min})$ be the normalized soft budget, $u_{\tau,l} = u_0 + (1 - u_0)H_{\tau,l}$ be the per-layer compression target (high-entropy layers get a relaxed target; $u_0 = 0.2$), and $s_{\text{safe}} = \sigma(-\Delta/T_q)$ be a soft safety gate (with temperature $T_q > 0$) that suppresses compression pressure when quality degrades. The budget loss combines compression and rescue objectives:

$$\begin{aligned} \mathcal{L}_{\text{budget}} = & w_{\text{cmp}} s_{\text{safe}} \left[\frac{1}{L} \sum_l \text{ReLU}(\bar{k}_{\tau,l} - u_{\tau,l}) \right. \\ & \left. + w_{\text{compute}} \frac{1}{L} \sum_l \bar{k}_{\tau,l} \right] \\ & + w_{\text{rescue}} (1 - s_{\text{safe}}) \\ & \cdot \frac{1}{L} \sum_l \text{ReLU}(u_{\tau,l} - \bar{k}_{\tau,l}). \end{aligned} \quad (11)$$

When quality is safe ($s_{\text{safe}} \approx 1$) the compression term dominates; when quality degrades ($s_{\text{safe}} \approx 0$) the rescue term takes over and increases the budget.

Alignment and consistency losses. $\mathcal{L}_{\text{align}}$ constrains the routing bias to remain anchored to the support routing distribution, preventing the prior branch from inverting the original gate ordering: $\mathcal{L}_{\text{align}} = \frac{1}{L} \sum_l \text{KL}(s_{\tau,l} \| \text{softmax}(\mathbf{b}_{\tau,l}))$, where the KL is computed with the support distribution as the target so that the learned bias is penalized for deviating from it. $\mathcal{L}_{\text{cons}}$ encourages policy stability: a second policy ($\rho'_{\tau,l}, \mathbf{p}'_{\tau,l}$) is computed from a random half-split of S_τ , and the loss penalizes disagreement with the full-set policy: $\mathcal{L}_{\text{cons}} = \frac{1}{L} \sum_l [(\rho_{\tau,l} - \rho'_{\tau,l})^2 + \|\mathbf{p}_{\tau,l} - \mathbf{p}'_{\tau,l}\|^2]$.

Auxiliary regularizers $\mathcal{L}_{\text{aux}}$. Three small-weight terms stabilize training: a cross-layer budget term $\mathcal{L}_{\text{res}} = \frac{1}{L} \sum_l (\bar{k}_{\tau,l} - \bar{u})^2$, which measures the squared deviation of each normalized soft budget from a reference level $\bar{u}$; a negative-entropy

reward $\mathcal{L}_{\text{ent}}$ on $\text{softmax}(\mathbf{p}_{\tau,l})$ (prevents prior collapse); and a rank loss $\mathcal{L}_{\text{rank}}$ that enforces soft monotonicity between routing entropy and budget:

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}|} \sum_{(l,l') \in \mathcal{P}} \text{ReLU}(\bar{k}_{\tau,l} - \bar{k}_{\tau,l'} + m_r), \quad (12)$$

where $\mathcal{P} = \{(l,l') : H_{\tau,l} < H_{\tau,l'}\}$ and $m_r = 0.05$. Principal hyperparameter settings are summarized in Table 6.

4 Experimental setup

4.1 Models and evaluation

The main backbone is DeepSeek-MoE-16B-Chat (Dai et al., 2024) (27 routed MoE layers, 64 sparse experts per layer, native top- k $K_{\text{nat}}=6$); all backbone parameters are frozen and only MetaNet g_ϕ is updated. Experiments run on three NVIDIA RTX 4090 24 GB GPUs (bfloat16, device_map=auto). MMLU episodes use $N_s = 8$ support and $N_q = 16$ disjoint query examples. For C-Eval, all five labeled development examples per subject form the support set and up to 16 labeled validation examples form the query set. MetaNet is trained for 450 episodic steps on MMLU (Hendrycks et al., 2021) meta-train subjects; the checkpoint with highest meta-validation accuracy is selected and evaluated on held-out meta-test subjects.

We run four experiments: **(1) Main**—MMLU (Hendrycks et al., 2021) meta-test (12 subtasks) vs. fixed top- k and HyperRouter-style; C-Eval (Huang et al., 2023) as zero-shot transfer. **(2) Cross-backbone**—the MMLU-trained checkpoint applied zero-shot to OLMoE-1B-7B (Muenighoff et al., 2025). **(3) Cross-architecture**—GoogLeNet (Szegedy et al., 2015) (frozen) as backbone on two synthetic episodic vision tasks (syn-pattern/syn-pattern-shift), selecting a 3-of-4 branch policy (active ratio 0.75). **(4) Ablations**—component (budget-only, prior-only, full) and layer-position (shallow/middle/deep/all).

Synthetic vision episodes. syn-pattern and syn-pattern-shift are procedural episodic tasks used only as a cross-architecture check. Each class is generated from a simple visual prototype, including color blocks, vertical or horizontal stripes, diagonal patterns, and ring-like structures, with random noise added to each image. The shifted split, syn-pattern-shift, additionally applies horizontal flipping and mild color-statistic perturbation.

Within each episode, support and query images are sampled from the same synthetic class distribution. The support set is used to infer the branch policy, and the query set evaluates whether this policy transfers to unseen samples from the same episode. This experiment tests whether support-conditioned structural control can be applied outside MoE routing.

For MMLU (Hendrycks et al., 2021) and C-Eval (Huang et al., 2023), each subject is treated as a task. The fixed MMLU split contains 34 meta-train, 11 meta-validation, and 12 meta-test subjects. Subject identities are disjoint across the three partitions, so the main MMLU result contains $12 \times 16 = 192$ held-out query predictions rather than additional samples from training subjects. Within each episode, support examples are used only to construct the routing profile and query examples are used for the reported metric.

Evaluation scopes. We distinguish the main transfer evaluation from the additional analyses. C-EVAL-FULL applies the MMLU-trained controller to all 52 C-Eval subjects without parameter updates and contains 828 labeled query examples. The expanded baselines and operating-point sweep use a fixed 10-subject subset with 16 query examples per subject (160 in total), denoted C-EVAL-10. Results from C-EVAL-FULL and C-EVAL-10 are reported in separate panels and are not compared directly. The ten subjects are listed in Appendix B.

Sampling seeds. Unless stated otherwise, results use one episode per subject. In the repeated-sampling check, the subject split remains fixed and the seed changes only support/query example sampling; support and query examples remain disjoint. Two seeds and two episodes per held-out MMLU subject yield 384 query predictions per seed. Fixed $k=6$, fixed $k=12$, and both MetaNet profiles are evaluated under both seeds. Appendix C.2 reports each run and the mean with sample standard deviation.

4.2 Baselines and metrics

Baselines. Fixed top- k uses $k \in \{1, \dots, 6, 12\}$; $k=6$ is the native router budget and $k=12$ is an upper-budget reference rather than a compute-matched baseline. HyperRouter-style (Do et al., 2023) uses a linear head over the same support statistics while keeping top- k fixed at 6. The expanded comparison also includes inference-compatible adaptations inspired by AdaMoE (Zeng

et al., 2024), Dynamic MoE (Guo et al., 2025), and Probe Pruning (Le et al., 2025). The rows are labeled “adapted” because they place the corresponding decision rule in the same frozen DeepSeek harness rather than reproduce each method’s original architecture and training procedure.

Metrics. For each task τ , accuracy is the exact-match rate on the query set Q_τ ; reported accuracies average this value over evaluation tasks. The mean dynamic top- k is

$$\bar{k} = \frac{1}{|\mathcal{T}|L} \sum_{\tau \in \mathcal{T}} \sum_{l=1}^L k_{\tau,l}, \quad (13)$$

where $k_{\tau,l}$ is the layer budget selected from the support set. The active expert ratio, $\bar{k}/E$, is a routed-expert workload proxy rather than total-model FLOPs; for fixed top- k , it is k/E (e.g., $6/64 = 0.094$). Latency is wall-clock time per query under the common three-GPU harness. We report it separately because dispatch, dense layers, device communication, and runtime overhead are absent from $\bar{k}/E$; it does not characterize an optimized MoE serving stack. Ablations report *gate agreement*, the layer/query-averaged Jaccard similarity between expert sets selected by MetaNet’s biased gate and the original gate, and *low-use selection*, the fraction of activated experts in the bottom quartile of support-set routing mass at their layer.

5 Results

5.1 Main results

Table 1 summarizes MMLU and C-Eval-Full results. **Compression vs. accuracy.** MetaNet reaches accuracy 0.438 at mean top- k 2.28 (ratio 0.036, 62% fewer activated experts than fixed $k=6$); fixed $k=3$ has lower accuracy (0.417) and a larger active ratio (0.047), indicating that a uniform budget does not match the layer-wise allocation learned by MetaNet. **Effect of support-conditioned routing.** The HyperRouter-style baseline, matched to ratio 0.094, loses 6.8 points relative to fixed $k=6$ (0.406 vs. 0.474), suggesting that support conditioning alone may be insufficient in this setting without budget control and a bounded interaction with the frozen gate. **Transfer to C-Eval.** The MMLU-trained controller transfers to C-Eval-Full without retraining (2.90 mean experts, ratio 0.045, 52% fewer activated experts than fixed $k=6$), outperforming the closest fixed baseline ($k=2$, ratio 0.031, acc. 0.377) at a nearby expert budget.

Method	Accuracy	Act. ratio	Mean k
MMLU (meta-test, 12 tasks $\times$ 16 queries)			
Fixed $k=6$	0.474	0.094	6.00
Fixed $k=12$	0.495	0.188	12.00
Fixed $k=4$	0.432	0.063	4.00
Fixed $k=3$	0.417	0.047	3.00
HyperRouter-style	0.406	0.094	6.00
MetaNet (aggressive)	0.438	0.036	2.28
MetaNet (conservative)	0.489	0.056	3.61
C-Eval-Full (52 subjects, 828 queries; zero-shot transfer)			
Fixed $k=6$	0.444	0.094	6.00
Fixed $k=2$	0.377	0.031	2.00
Fixed $k=4$	0.443	0.063	4.00
MetaNet (aggressive)	0.386	0.045	2.90
MetaNet (conservative)	0.452	0.056	3.61

Table 1: Main results on MMLU and C-Eval-Full. Bold accuracy marks the stronger MetaNet setting at or below the native $k=6$ budget; bold active-ratio and mean- k values mark the minimum routed-expert workload in each panel. In the MMLU panel, fixed $k=12$ is an upper-budget reference and is excluded from the bold accuracy comparison. The C-Eval fixed- $k=12$ result was measured only on C-Eval-10 and is reported separately in Table 7. The conservative setting uses a lower budget penalty weight λ_{budget} .

Table 2 reports cross-backbone transfer. Applied to OLMoE-1B-7B (Muennighoff et al., 2025) without updating weights, MetaNet exceeds fixed $k=8$ on both MMLU (0.573 vs. 0.521) and C-Eval (0.375 vs. 0.363) with mean top- $k=7.52$. Since this budget is close to OLMoE’s native $k=8$, the improvement may reflect the bounded routing bias rather than expert-count reduction alone (Section 5.4). Table 3 gives the GoogLeNet cross-architecture check: MetaNet reaches 0.66 on syn-pattern-shift vs. 0.58 for static 3/4-branch selection at the same active ratio.

The conservative and aggressive settings show how the objective controls the operating point. The aggressive objective favors smaller budgets and indicates that many layers can operate below the native $k=6$ budget, at the cost of some accuracy. The conservative objective lowers the compression pressure and keeps more experts active when the quality guard predicts risk; this reduces the MMLU gap while still lowering the average active-expert count from 6.0 to 3.61. The two settings use the same controller family with different budget-loss weights.

5.2 Expanded baselines and operating points

Table 7 reports the additional comparisons under a common frozen-inference harness and restores the fixed $k=12$ reference in the C-Eval-10 panel. The MMLU panel retains the 12-subject, 192-query scope, whereas the C-Eval panel uses C-Eval-10 (10 subjects, 160 queries). The AdaMoE and Dynamic MoE adaptations remain close to six experts

Backbone	Method	MMLU	C-Eval	Mean k
DeepSeek-MoE-16B-Chat ($K_{\text{nat}}=6$, 64 exp.)	Fixed $k=6$	0.474	0.444	6.00
	Fixed $k=4$	0.432	0.443	4.00
	Fixed $k=2$	0.365	0.377	2.00
	MetaNet	0.438	0.386	2.28
OLMoE-1B-7B ($K_{\text{nat}}=8$, 64 exp.)	Fixed $k=8$	0.521	0.363	8.00
	Fixed $k=4$	0.479	0.350	4.00
	Fixed $k=2$	0.417	0.350	2.00
	MetaNet	0.573	0.375	7.52

Table 2: Cross-backbone generalization (Exp. II). MetaNet trained on DeepSeek and evaluated zero-shot on OLMoE-1B-7B (Muennighoff et al., 2025). Fixed- k rows report each backbone’s native and lower budgets; the final column reports mean k on MMLU.

on average. MetaNet provides operating points with lower routed-expert workload; the Probe Pruning adaptation is faster but loses substantial accuracy. Section 4 states the different deployment assumptions.

Table 8 gives the denser operating-point sweep. The tags bt0p3, bt0p4, bt0p5, and bt0p8 encode nominal base budget targets $u_0 \in \{0.3, 0.4, 0.5, 0.8\}$ for independently trained controllers in the same family; they are not achieved active ratios. The resulting points are not strictly monotonic, so we report the achieved mean top- k and active ratio for every run rather than describe every candidate as Pareto-optimal. Appendix C.2 reports matched seed-0/seed-1 runs. Conservative MetaNet obtains 0.506 ± 0.022 accuracy, and aggressive MetaNet obtains 0.477 ± 0.022 (mean $\pm$ sample standard deviation).

Dataset	Method	Accuracy	Act. ratio
syn-pattern	Original	1.00	1.00
	Static	1.00	0.75
	MetaNet	0.98	0.75
syn-pattern-shift	Original	0.99	1.00
	Static	0.58	0.75
	MetaNet	0.66	0.75

Table 3: Cross-architecture proof-of-concept results (Experiment III, GoogLeNet backbone). Bold marks the full-compute reference and the best result at active ratio 0.75 for each dataset.

Variant	Acc.	Act. ratio	Mean k	Gate agr.	Low-use
Fixed $k=6$	0.474	0.094	6.00	—	—
Budget-only	0.375	0.045	2.89	0.038	0.734
Prior-only	0.484	0.188	12.00	0.622	0.318
Full method	0.438	0.036	2.28	0.943	0.004

Table 4: Method-component ablation on MMLU. Bold marks the best value in each metric column (higher is better for accuracy and gate agreement; lower is better otherwise). Dashes denote controller diagnostics omitted for fixed routing.

5.3 Ablation studies

Method components. Table 4 ablates the two branches. Removing the prior branch (Budget-only) drops accuracy to 0.375 and gate agreement to 0.038: the budget branch alone selects expert sets with little overlap with the original gate. Removing the budget branch (Prior-only) uses $k=12$ throughout and recovers the highest accuracy (0.484), but at the full $k=12$ routing budget. With both branches, gate agreement rises to 0.943 and low-use selection drops to 0.004, indicating that the prior branch keeps expert selection close to the original gate while the budget branch reduces active experts.

Layer position. Table 5 restricts compression to different layer ranges, holding all other layers at fixed $k=6$. Compressing only the deep layers (layers 18–26) costs 0.005 accuracy points relative to the uncompressed baseline while reaching ratio 0.069, the best efficiency–accuracy trade-off in this sweep. Compressing only the shallow layers (layers 0–8) loses 0.078 accuracy points at ratio 0.087, indicating that early-layer expert selection is less redundant. Middle, FrontBack-10, and Random-9 configurations all reach 0.427. Full-layer compression pushes ratio down to 0.036 and achieves 0.438 accuracy, showing that joint optimization across all layers can partially compensate for the harder

Variant	Acc.	Act. ratio	Mean k
Fixed $k=6$ (all layers)	0.474	0.094	6.00
All layers	0.438	0.036	2.28
Shallow-9 (0–8)	0.396	0.087	5.59
Middle-9	0.427	0.074	4.75
Deep-9 (18–26)	0.469	0.069	4.40
FrontBack-10	0.427	0.081	5.16
Random-9	0.427	0.077	4.94

Table 5: Layer-mask ablation on MMLU. Bold marks the best partial-layer configuration among Shallow-9, Middle-9, Deep-9, FrontBack-10, and Random-9.

shallow-layer budget.

5.4 Mechanistic analysis

Figure 3 visualizes the support-derived routing mass over 27 layers and 64 experts. Expert use changes substantially with depth rather than following one layer-invariant profile. The mean cross-layer Jaccard overlap of activated expert sets is 0.10 (max 0.26), computed by averaging pairwise Jaccard similarity between layer-level activated expert sets within each task; low overlap indicates that different layers often select different expert subsets. Both latency harnesses show modest reductions. In the original harness, MetaNet runs in 491.1 ms per query vs. 508.5 ms for fixed $k=6$ (3.4% lower). In the separate common three-GPU harness, aggressive MetaNet reduces latency from 542.94 to 516.52 ms (4.9%) while reducing mean top- k from 6.00 to 2.28 (62.0% fewer activated experts). Measurements from the two harnesses are not pooled. The gap between the reductions in mean top- k and latency indicates that the current inference stack does not convert routing sparsity proportionally into end-to-end speedup: MetaNet changes routed-expert execution, whereas dense attention, token dispatch and permutation, synchronization, and inter-device communication remain. This distinction is consistent with system-level MoE work that introduces specialized kernels, adaptive execution, or communication scheduling (Rajbhandari et al., 2022; Hwang et al., 2023; Li et al., 2023). The support-profiling pass is computed once per episode and reused across all queries. On OL-MoE, MetaNet exceeds fixed $k=8$ by 5.2 points (0.573 vs. 0.521); because its mean top- k is close to the native $k=8$, this experiment supports cross-backbone transfer but provides only weak evidence for compression. The bounded bias is a plausible contributor, but the experiment does not isolate that mechanism.

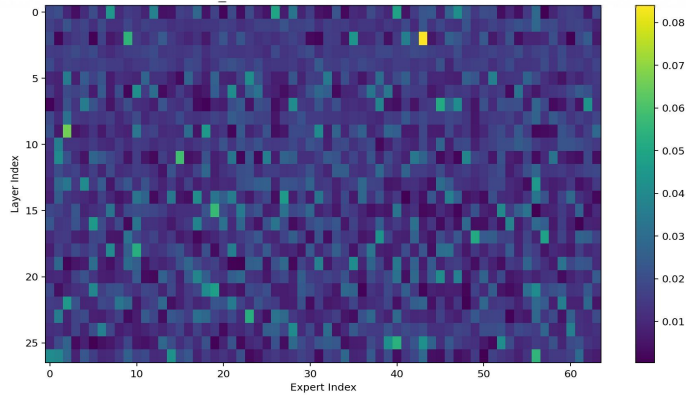


Figure 3: Support-derived layer-expert routing profile for the selected MMLU checkpoint (27 routed layers $\times$ 64 experts; color denotes mean normalized routing mass).

The nonuniform support profile helps explain why one fixed top- k can be inefficient, while the layer-position ablation supplies the direct budget evidence: deep-layer compression is relatively safe, but optimizing all layers jointly gives a more compressed operating point. MetaNet uses the task support set to construct this layer-wise policy rather than relying on one handcrafted layer mask.

6 Conclusion

MetaNet is a support-set controller for frozen MoE inference. It predicts per-layer expert budgets and a bounded routing bias without updating backbone parameters. Across MMLU, C-Eval, a second MoE backbone, and a vision backbone, the results show that expert demand varies by task and depth. Modeling this variation gives better accuracy-expert-activation trade-offs than uniform fixed- k routing.

Limitations

MetaNet reduces activated-expert workload rather than model storage or total end-to-end computation. The active ratio is a routing-level proxy, not a measurement of total model FLOPs, and the full expert pool remains resident in memory. The method therefore does not provide proportional parameter-storage or peak-memory savings in the current implementation. Wall-clock speedup depends on whether the runtime can exploit the smaller routed-expert set; without specialized dispatch and communication support, latency need not decrease in proportion to mean top- k . Integrating MetaNet with optimized MoE runtimes is left for future work.

The main MMLU result uses 192 held-out

queries, and the repeated-sampling check contains only two seeds. We report both runs and their sample standard deviation for the fixed-budget and MetaNet profiles, but two seeds do not support a reliable confidence interval or significance test. C-Eval-Full and C-Eval-10 have different scopes and are reported separately. The adapted baselines share our frozen inference harness but are not exact reproductions of methods that require different training or architectures. In addition, the controller is evaluated mainly on academic multiple-choice benchmarks; open-ended generation, long-context tasks, and safety-sensitive downstream applications may exhibit different routing patterns. Finally, the best layer-wise budget location is treated as a learned outcome rather than a fully characterized principle.

Ethical Considerations

This work studies inference-time efficiency for released MoE models and does not introduce a new pretrained language model, dataset, or user-facing deployment system. Lower activated-expert workload can reduce hardware cost and energy use when paired with suitable sparse dispatch kernels. Potential risks are indirect: cheaper inference may make existing LLM misuse scenarios easier to scale. The method does not add new generative capabilities, collect personal data, or rely on human-subject experiments. We use publicly available research benchmarks and model checkpoints according to their release terms and report aggregate evaluation metrics only. The implementation is available at <https://github.com/Notletterday/MetaNet-MoE>.

Acknowledgments

This work was supported by the Major Science and Technology Project of Henan Province, China, entitled *Research and Industrialization of Key Technologies for Intelligent Computing in Large-Scale Video Scenarios under Digital Social Governance* (Grant No. 241100210100).

References

- Mengting Ai, Tianxin Wei, Yifan Chen, Zhichen Zeng, Ritchie Zhao, Girish Varatkar, Bitu Darvish Rouhani, Xianfeng Tang, Hanghang Tong, and Jingrui He. 2025. [ResMoE: Space-efficient compression of mixture of experts LLMs via residual restoration](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.I*, pages 1–12. Association for Computing Machinery.
- Ruisi Cai, Yeonju Ro, Geon-Woo Kim, Peihao Wang, Babak Ehteshami Bejnordi, Aditya Akella, and Zhangyang Wang. 2024. [Read-ME: Refactorizing LLMs as router-decoupled mixture of experts with system co-design](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 116126–116148. Curran Associates, Inc.
- I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. 2025. [Retraining-free merging of sparse MoE via hierarchical clustering](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 8594–8620. PMLR.
- Krishna Teja Chitty-Venkata and Murali Emani. 2026. [LExI: Layer-adaptive active experts for efficient MoE inference](#). In *AdaptFM: Resource-Adaptive Foundation Model Inference*.
- Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. [DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297. Association for Computational Linguistics.
- Truong Giang Do, Le Khiem, Quang Pham, TrungTin Nguyen, Thanh-Nam Doan, Binh Nguyen, Chenghao Liu, Savitha Ramasamy, Xiaoli Li, and Steven Hoi. 2023. [HyperRouter: Towards efficient training and inference of sparse mixture of experts](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5754–5765. Association for Computational Linguistics.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#). *Journal of Machine Learning Research*, 23(120):1–39.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. [Model-agnostic meta-learning for fast adaptation of deep networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.
- Chongyang Gao, Kezhen Chen, Jinmeng Rao, Ruibo Liu, Baochen Sun, Yawen Zhang, Daiyi Peng, Xiaoyuan Guo, and V. S. Subrahmanian. 2025. [MoLA: MoE LoRA with layer-wise expert allocation](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5112–5127. Association for Computational Linguistics.
- Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. 2025. [Dynamic mixture of experts: An auto-tuning approach for efficient transformer models](#). In *The Thirteenth International Conference on Learning Representations*, pages 79643–79672.
- David Ha, Andrew Dai, and Quoc V. Le. 2017. [HyperNetworks](#). In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. [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)*, pages 12883–12895. Association for Computational Linguistics.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. [C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 62991–63010. Curran Associates, Inc.
- Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, HoYuen Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. 2023. [Tutel: Adaptive mixture-of-experts at scale](#). In *Proceedings of Machine Learning and Systems*, volume 5, pages 269–287.
- Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. 2024. [Pre-gated MoE: An algorithm-system co-design for fast and scalable mixture-of-expert inference](#). In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pages 1018–1031. IEEE.
- Eric Jang, Shixiang Gu, and Ben Poole. 2017. [Categorical reparameterization with Gumbel-Softmax](#). In *International Conference on Learning Representations*.
- 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, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of experts](#). *arXiv preprint arXiv:2401.04088*.
- Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhengting Wang, Wenye Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. [Exploring concept depth: How large language models acquire knowledge and concept at different layers?](#) In *Proceedings of the 31st International*

- Conference on Computational Linguistics, pages 558–573. Association for Computational Linguistics.
- Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. 2025. [Remarkable robustness of LLMs: Stages of inference?](#) In *Advances in Neural Information Processing Systems*, volume 38, pages 130050–130083. Curran Associates, Inc.
- Qi Le, Enmao Diao, Ziyan Wang, Xinran Wang, Jie Ding, Li Yang, and Ali Anwar. 2025. [Probe pruning: Accelerating LLMs through dynamic pruning via model-probing](#). In *The Thirteenth International Conference on Learning Representations*, pages 72494–72513.
- Jaeseong Lee, Seung-won Hwang, Aurick Qiao, Daniel F. Campos, Zhewei Yao, and Yuxiong He. 2025. [STUN: Structured-then-unstructured pruning for scalable MoE pruning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13660–13676. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.
- Jiamin Li, Yimin Jiang, Yibo Zhu, Cong Wang, and Hong Xu. 2023. [Accelerating distributed MoE training and inference with Lina](#). In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 945–959. USENIX Association.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. 2022. [Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc.
- Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. [Learning sparse neural networks through \$L_0\$ regularization](#). In *International Conference on Learning Representations*.
- Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. [Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6172. Association for Computational Linguistics.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. [LLM-Pruner: On the structural pruning of large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 21702–21720. Curran Associates, Inc.
- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. [The concrete distribution: A continuous relaxation of discrete random variables](#). In *International Conference on Learning Representations*.
- Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, and Mohammad Rastegari. 2024. [OpenELM: An efficient language model family with open training and inference framework](#). In *Proceedings of the 2nd Workshop on Efficient Systems for Foundation Models at ICML 2024*.
- Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. [ShortGPT: Layers in large language models are more redundant than you expect](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204. Association for Computational Linguistics.
- Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Øyvind Taffjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2025. [OLMoE: Open mixture-of-experts language models](#). In *The Thirteenth International Conference on Learning Representations*, pages 62061–62121.
- OpenAI. 2023. [GPT-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Peijun Qing, Chongyang Gao, Yefan Zhou, Xingjian Diao, Yaoqing Yang, and Soroush Vosoughi. 2024. [AlphaLoRA: Assigning LoRA experts based on layer training quality](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20511–20523. Association for Computational Linguistics.
- Samy Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. [DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). In *International Conference on Learning Representations*.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. [Going deeper with convolutions](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [LLaMA: Open and efficient foundation language models](#). *arXiv preprint arXiv:2302.13971*.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. [Matching networks for one shot learning](#). In *Advances in Neural Information Processing Systems*, volume 29, pages 3630–3638. Curran Associates, Inc.
- Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. 2024. [MoE-I²: Compressing mixture of experts models through inter-expert pruning and intra-expert](#)

low-rank decomposition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10456–10466. Association for Computational Linguistics.

Juntao Yao, Huiyuan Zheng, Boyang Wang, Xiaohu Yu, Yibo Li, Shaosheng Cao, Donglin Di, Boyan Wang, Haoyun Zheng, Jinze Yu, Anjie Le, and Hongcheng Guo. 2026. [Domain-specific expert pruning for mixture-of-experts LLMs](#). In *Proceedings of the Second AAAI Bridge Program on AI for Medicine and Healthcare*, volume 317 of *Proceedings of Machine Learning Research*, pages 313–321. PMLR.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. 2024. [AdaMoE: Token-adaptive routing with null experts for mixture-of-experts language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6223–6235. Association for Computational Linguistics.

Zeliang Zhang, Nikhil Ghosh, Jiani Liu, Bin Yu, and Xiaodong Liu. 2026. [Does a global perspective help prune sparse MoEs elegantly?](#) *arXiv preprint arXiv:2604.06542*.

A Hyperparameters

Parameter	Value
<i>Backbone & modules</i>	
Model	DeepSeek-MoE-16B-Chat
Frozen	backbone, experts, original gate
Trainable	MetaNet only
Routed experts / layer; native top- k (K_{nat})	64; 6
<i>Retention profile</i>	
Reference top- k (R); routing strategy	12; retention_mass
k_{min} ; k_{max}	1; 12
<i>MetaNet training</i>	
Steps; learning rate	450; 5×10^{-4}
<i>Weak prior</i>	
α ; β ; max bias τ_b	0.05; 0.90; 0.05
<i>Objective</i>	
Budget loss mode	quality_first
Rescue w_{rescue} ; compute w_{compute}	6.0; 0.12
Task scale; layer scale	0.12; 0.18
Overshoot weight; upper tolerance	2.5; 0.05
Base compression target u_0	0.2
Safety gate temperature T_q	0.05
Budget sigmoid temperature T_{bud}	0.05
Perf.-preserving margin m_{pp} ; Huber δ_H	0.01; 0.10

Table 6: Hyperparameter configuration ($R = 12$) used in the main experiments.

B Additional experimental details

MMLU subject split. The fixed split contains 34 meta-train, 11 meta-validation, and 12 meta-test subjects. The meta-train subjects are:

abstract_algebra, anatomy, astronomy, business_ethics, clinical_knowledge, college_biology, college_chemistry, college_computer_science, college_mathematics, college_medicine, computer_security, conceptual_physics, econometrics, elementary_mathematics, formal_logic, high_school_biology, high_school_chemistry, high_school_computer_science, high_school_geography, high_school_macroeconomics, high_school_mathematics, high_school_physics, high_school_statistics, human_aging,

international_law, logical_fallacies, machine_learning, management, marketing, medical_genetics, miscellaneous, nutrition, sociology, us_foreign_policy.

The meta-validation subjects are: college_physics, electrical_engineering, global_facts, high_school_european_history, high_school_government_and_politics, high_school_microeconomics, high_school_psychology, human_sexuality, jurisprudence, moral_disputes, professional_accounting.

The meta-test subjects are: high_school_us_history, high_school_world_history, philosophy, prehistory, moral_scenarios, world_religions, professional_law, professional_medicine, professional_psychology, public_relations, security_studies, virology.

The split is fixed across all sampling seeds.

C-Eval scopes. C-Eval-Full uses all 52 subjects and contains 828 labeled queries. Because public test labels are unavailable, all five development examples in each subject are used as support and up to 16 labeled validation examples are used as queries. C-Eval-10 contains:

middle_school_mathematics, middle_school_physics, middle_school_politics, modern_chinese_history, plant_protection, professional_tour_guide, sports_science, teacher_qualification, urban_and_rural_planner, veterinary_medicine.

Every C-Eval-10 subject contributes 16 queries. MetaNet receives no C-Eval parameter updates in either scope.

Adapted-baseline scope. All adapted rows preserve the DeepSeek backbone and original router. HyperRouter-style uses a support-conditioned linear head at fixed $k=6$; AdaMoE adapted uses the top-score margin; Dynamic MoE adapted uses a variable- k rule inspired by Guo et al. (2025); and Probe Pruning adapted uses a support-conditioned expert mask. These are inference-compatible adaptations, not exact reproductions of methods whose original versions require different training or pruning procedures; Probe Pruning latency therefore reflects a different execution path.

C Additional experimental results

Method	Acc.	Act. ratio	Mean k	Lat. (ms)
MMLU: 12 subjects, 192 queries				
Fixed $k=6$	0.474	0.094	6.00	542.94
Fixed $k=12$	0.495	0.188	12.00	557.65
HyperRouter-style*	0.406	0.094	6.00	539.32
Probe Pruning adapted	0.188	0.094	6.00	180.45
AdaMoE adapted*	0.479	0.090	5.76	577.90
Dynamic MoE adapted*	0.453	0.094	6.00	572.18
MetaNet (aggressive)	0.438	0.036	2.28	516.52
MetaNet (conservative)	0.489	0.056	3.61	536.34
C-Eval-10: 10 subjects, 160 queries				
Fixed $k=6$	0.493	0.094	6.00	532.35
Fixed $k=12$	0.581	0.188	12.00	531.36
HyperRouter-style*	0.443	0.094	6.00	487.80
Probe Pruning adapted	0.281	0.094	6.00	146.15
AdaMoE adapted*	0.513	0.090	5.76	526.39
Dynamic MoE adapted*	0.519	0.094	6.00	541.96
MetaNet (comparison setting)	0.487	0.065	4.16	474.53

Table 7: Expanded comparison under the common inference harness. “Adapted” denotes an inference-compatible decision rule rather than an exact reproduction. The MetaNet comparison setting is reported separately from the profile sweep in Table 8. C-Eval-10 is separate from C-Eval-Full in Table 1. * The original method requires training or replacement of routing components.

C.1 Operating-point sweep

Table 8 reports four independently trained controller profiles. The bt0p3, bt0p4, bt0p5, and bt0p8 tags denote nominal base budget targets $u_0 = 0.3, 0.4, 0.5$, and 0.8 , respectively, while the other shared settings follow Table 6. These targets configure the training objective; they are not achieved active ratios. We therefore report the realized metrics and do not claim that every row is Pareto-optimal.

Scope	Profile	Acc.	Act. ratio	Mean k	Lat. (ms)
MMLU	bt0p3	0.460	0.027	1.75	527.07
	bt0p4	0.472	0.031	1.98	527.96
	bt0p5	0.477	0.040	2.55	516.02
	bt0p8	0.494	0.045	2.88	502.15
C-Eval-10	bt0p3	0.375	0.036	2.30	457.05
	bt0p4	0.487	0.102	6.50	457.41
	bt0p5	0.434	0.114	7.30	457.64
	bt0p8	0.490	0.145	9.30	443.84

Table 8: Additional operating-point sweep. Active ratio is the realized mean top- k divided by 64. Latency is measured in the same three-GPU harness.

C.2 Sensitivity to episode sampling

The subject split is fixed in this check; the seed changes support/query sampling only. Each seed

contains two episodes per subject and 384 query predictions.

Method	Seed	Acc.	Act. ratio	Mean k	Lat. (ms)
Fixed $k=6$	0	0.484	0.094	6.00	537.83
Fixed $k=6$	1	0.526	0.094	6.00	538.22
Fixed $k=12$	0	0.503	0.188	12.00	538.69
Fixed $k=12$	1	0.531	0.188	12.00	540.36
MetaNet (conservative)	0	0.490	0.080	5.13	514.58
MetaNet (conservative)	1	0.521	0.074	4.75	511.83
MetaNet (aggressive)	0	0.461	0.070	4.47	535.72
MetaNet (aggressive)	1	0.492	0.060	3.82	496.27

Table 9: Repeated MMLU evaluation. Active ratio is mean top- k divided by 64.

Across the two seeds, accuracy is 0.505 ± 0.030 for fixed $k=6$, 0.517 ± 0.020 for fixed $k=12$, 0.506 ± 0.022 for conservative MetaNet, and 0.477 ± 0.022 for aggressive MetaNet (mean $\pm$ sample standard deviation). The corresponding mean top- k values are 6.00 ± 0.00 , 12.00 ± 0.00 , 4.94 ± 0.27 , and 4.15 ± 0.46 . With only two seeds, these results measure sampling sensitivity rather than establish statistical significance.