ADAUPMOE: ADAPTIVE FINE-GRAINED COMPRESSION OF UPCYCLED MOE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) has become a promising architecture for scaling large models with sparse activation. Recently, upcycling dense models into MoE models has emerged as an effective approach for constructing powerful sparse models, motivating the need for effective compression techniques for upcycled MoE models. In this work, we present the first systematic study of upcycled MoE compression and reveal two key observations: (1) existing methods typically apply uniform compression strategies across the entire model, overlooking the heterogeneous compression requirements among different MoE components; (2) compression effectiveness is highly sensitive to fine-grained compression policies, including sparsity ratios and rescaling factors. Based on these findings, we propose **AdaUpMoE**, an adaptive fine-grained compression framework for upcycled MoE models. Specifically, AdaUpMoE formulates compression as a component-wise policy learning problem, leveraging a small amount of unlabeled data and a two-stage distillation-based learning paradigm to automatically learn compression policies for different layers, experts, and model modules. By adaptively allocating compression budgets according to component characteristics, AdaUpMoE effectively exploits the heterogeneous redundancy of upcycled MoE models while preserving model capability. Extensive experiments on 5 upcycled MoE approaches, 8 models with diverse architectures and scales, and 30 datasets covering various tasks and modalities demonstrate that, AdaUpMoE consistently outperforms existing compression approaches, achieving particularly significant improvements under extreme (up to 100$\times$ on expert-specific deltas) compression regimes.

1 INTRODUCTION

Mixture-of-Experts (MoE) (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Du et al., 2022; Zhou et al., 2022; Dai et al., 2024) has emerged as a promising architecture for scaling large models with sparse activation, enabling larger parameter capacity while maintaining manageable computational costs. Despite their scalability, training MoE models from scratch requires considerable computational resources and large-scale training data (Zoph et al., 2022; DeepSeek-AI, 2026).

Recently, upcycling, which transforms pretrained dense models into sparse MoE models, has attracted increasing attention (Komatsuzaki et al., 2023; Liew et al., 2025). By expanding dense Feed-Forward Network (FFN) into multiple experts while inheriting pretrained weights, upcycling enables the construction of powerful MoE models with significantly reduced costs (Lin et al., 2026; Li et al., 2025; Jiang et al., 2024; Ding et al., 2024; Wang et al., 2025). However, the increased number of experts also introduces substantial parameter redundancy (Lu et al., 2024; Huang et al., 2025b), causing considerable storage and memory overhead at deployment. Thus, effective compression techniques are essential for enabling the efficient deployment of upcycled MoE models.

Existing compression approaches for MoE models mainly follow two directions. The first direction directly applies general-purpose MoE compression techniques, including pruning (Chen et al., 2022; Xie et al., 2024), quantization (Frantar & Alistarh, 2024; Huang et al., 2025a), expert reduction (Lu et al., 2024; Yang et al., 2024; Lasby et al., 2025), and parameter sharing (Li et al., 2024c). However, these methods are primarily designed for MoE models trained from scratch and cannot reach very high compression ratios. The second direction specifically targets upcycled MoE compression by exploiting the redundancy inherited from dense-to-MoE transformation.

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.

As the pioneering work, DeRS (Huang et al., 2025b) first decomposes expert weights into expert-shared base parameters and expert-specific delta parameters and then compresses the delta parameters through sparsity (Yu et al., 2024a) or quantization. Although effective, these methods mainly focus on what parameters are suitable for compressing while overlooking how compression should be allocated. Specifically, they typically apply uniform compression policies across different model components, ignoring their varying compressibility and sensitivity to compression.

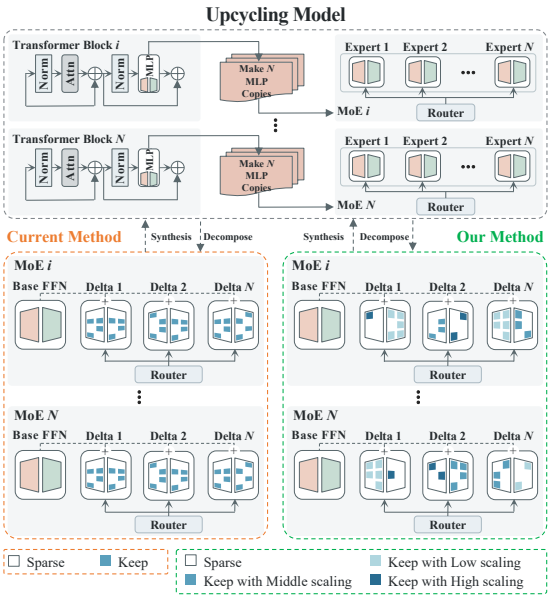


Figure 1: Overview of upcycled MoE construction and compression. Existing methods apply a uniform policy to all expert delta parameters, whereas AdaUpMoE learns component-wise sparsity ratios and rescaling factors across layers, experts, and model modules.

To investigate this issue, we conduct the first systematic analysis of fine-grained compression behaviors in upcycled MoE models and uncover two important characteristics. First, compression requirements are highly heterogeneous across layers, experts, and modules, echoing prior observations on dense and MoE models (He et al., 2018; Wang et al., 2019; Li et al., 2024b): some components tolerate aggressive compression while others degrade quickly at even modest compression ratios. Second, compression effectiveness is highly sensitive to the fine-grained policy, i.e., the sparsity ratios and rescaling factors. Even under the same overall compression budget, different compression allocations can lead to substantially different performance. These findings indicate that, effective upcycled MoE compression requires more than simply removing redundant parameters, it requires adaptive mechanisms that can automatically learn component-specific compression policies and allocate compression budgets accordingly.

Motivated by these observations, we propose AdaUpMoE, an adaptive fine-grained compression framework for upcycled MoE models. Unlike existing approaches that apply predefined compression policies uniformly, AdaUpMoE formulates compression as a component-wise policy learning problem and adaptively allocates compression budgets according to the characteristics of different components. Specifically, leveraging only a small amount of unlabeled data, AdaUpMoE introduces a two-stage distillation-based learning paradigm to automatically learn compression policies, including sparsity ratios and rescaling factors, for different layers, experts, and model modules, as shown in Figure 1. Extensive experiments across 5 upcycled MoE approaches, 8 models, and 30 datasets covering diverse tasks and modalities demonstrate the effectiveness and generality of AdaUpMoE, especially under extreme compression regimes.

Our contributions are summarized as follows:

- **Comprehensive analysis of upcycled MoE compression.** We present the first systematic study of upcycled MoE compression, showing that compression requirements are highly heterogeneous across MoE components and that performance is highly sensitive to fine-grained sparsity ratios and rescaling factors, which existing uniform policies overlook.
- **Adaptive and fine-grained compression framework for upcycled MoE models.** We propose **AdaUpMoE**, to our knowledge the first adaptive fine-grained framework for upcycled MoE delta compression. Specifically, leveraging a small amount of unlabeled data, AdaUpMoE employs a two-stage distillation-based learning paradigm to automatically learn component-wise compression policies for different layers, experts, and model modules, enabling adaptive fine-grained compression.
- **Extensive evaluation across diverse upcycled MoE approaches, models, and datasets.** Experiments on 5 upcycled MoE approaches, 8 models, and 30 datasets show that AdaUp-

MoE consistently outperforms existing compression methods, with the largest gains under extreme (up to 100×) compression.

2 Related Work

2.1 Mixture-of-Experts Models and Upcycling

MoE architectures scale capacity by routing each input to a small subset of experts (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Du et al., 2022; Zhou et al., 2022; Dai et al., 2024), but it is costly and unstable to train them from scratch (Zoph et al., 2022). Sparse upcycling (Komatsuzaki et al., 2023) avoids this cost by replicating the dense model's feed-forward networks into multiple experts and continuing training. The paradigm has since spread to code (Ding et al., 2024; Wang et al., 2025), vision-language, multimodal (Lin et al., 2026; Li et al., 2024a; 2025), medical (Jiang et al., 2024), and speech models (Fu et al., 2025), with construction and scaling variants (Zhu et al., 2024; Li et al., 2026; Liew et al., 2025). A shared consequence is substantial parameter redundancy (Lu et al., 2024; Li et al., 2024c; Huang et al., 2025b). Experts dominate the parameter budget of MoE models in general, accounting for 1.55T of the 1.6T parameters in DeepSeek-V4-Pro (DeepSeek-AI, 2026), making storage and memory overhead a first-order deployment concern.

2.2 Compression of MoE Models

Existing MoE compression methods mostly target models trained from scratch. They prune rarely-routed experts (Chen et al., 2022; Lu et al., 2024; Xie et al., 2024; Lasby et al., 2025), merge redundant experts or share parameters (Li et al., 2024c), use low-rank factorization (Yang et al., 2024), or quantize experts to extremely low bit-widths (Frantar & Alistarh, 2024; Huang et al., 2025a).

None of them accounts for the strong correlations that upcycled experts inherit from their shared dense initialization. To our knowledge, DeRS (Huang et al., 2025b) is the first method to exploit this structure. It decomposes each expert into a shared base weight and an expert-specific delta, compressing only the delta by DARE-style random sparsification with analytic rescaling (Yu et al., 2024a) or by quantization. Its compression policy, however, is uniform and predefined, with a single sparsity ratio and rescaling factor applied to every component. Our analysis in Figure 2 indicates that such uniformity fails under aggressive compression, because compression requirements are highly heterogeneous across layers, experts, and modules. In this paper, we therefore learn component-wise compression policies for upcycled MoE models.

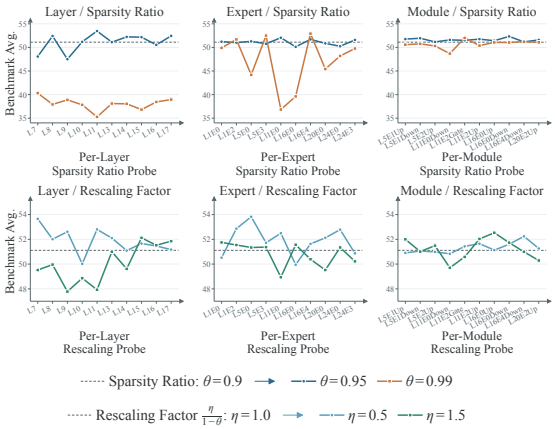


Figure 2: Fine-grained compression probes on $\mathcal{X}$FT under different sparsity ratios and rescaling factors. $LxEy$ denotes expert y in layer x. Up, Down, and Gate denote the up, down, and gate projections, respectively. Benchmark Avg. denotes the arithmetic mean of pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

3 Methodology

Guided by this heterogeneity, AdaUpMoE replaces the globally shared sparsity ratio and the analytic rescaling factor of prior delta compression methods (Yu et al., 2024a; Huang et al., 2025b) with learnable sparsity ratios and rescaling factors assigned separately to each layer, expert, and module. These component-wise parameters are then optimized through a two-stage distillation-based learning paradigm using only a small amount of unlabeled data. Figure 3 summarizes this two-stage pipeline and its teacher- and module-level distillation signals.

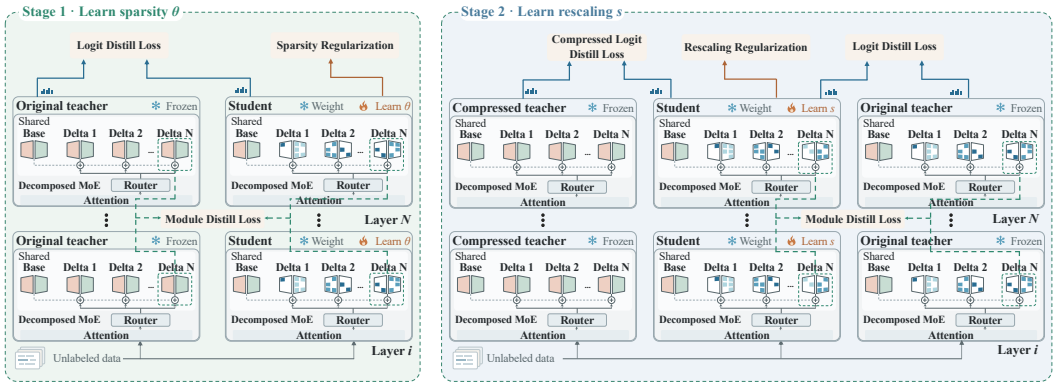


Figure 3: Overview of AdaUpMoE's two-stage policy-learning pipeline. With a small amount of unlabeled data, Stage 1 learns component-wise sparsity ratios θ under the global target sparsity θ^* using original-teacher, while Stage 2 learns component-wise rescaling factors s under the rescaling regularization using both original-teacher and compressed-teacher.

3.1 Preliminaries

Consider an upcycled MoE model (Komatsuzaki et al., 2023) with L MoE layers, E experts per layer, and P linear modules per expert. For layer $l \in \{1, \dots, L\}$, expert $e \in \{1, \dots, E\}$, and module type $p \in \{1, \dots, P\}$, such as the gate, up, and down projections of an FFN expert, we denote the trained expert weight by $\mathbf{W}^{\text{moe}}_{l,e,p} \in \mathbb{R}^{R_p \times C_p}$, where R_p and C_p are the output and input dimensions of the p-th module type. Since all experts of a layer are initialized from the corresponding FFN weight $\mathbf{W}^{\text{base}}_{l,p} \in \mathbb{R}^{R_p \times C_p}$ of the pretrained dense model, the trained weight can be decomposed as (Huang et al., 2025b)

$$\mathbf{W}^{\text{moe}}_{l,e,p} = \mathbf{W}^{\text{base}}_{l,p} + \mathbf{\Delta}_{l,e,p}, \tag{1}$$

where $\mathbf{W}^{\text{base}}_{l,p}$ serves as the expert-shared base weight and $\mathbf{\Delta}_{l,e,p} \in \mathbb{R}^{R_p \times C_p}$ is the expert-specific delta. Throughout this section, bold symbols denote matrices or tensors, and scalar functions applied to them act element-wise unless otherwise stated. Random sparsification (Yu et al., 2024a; Huang et al., 2025b) sets each element of $\mathbf{\Delta}_{l,e,p}$ to zero with probability θ^* and rescales the survivors by $1/(1-\theta^*)$ to keep the expectation unbiased. Both the sparsity ratio θ^* and the rescaling factor are shared by all layers, experts, and modules. In AdaUpMoE, θ^* instead serves as the global target sparsity.

3.2 Component-Wise Learnable Compression

Each component-wise compression policy consists of a sparsity ratio and a rescaling factor.

Learnable Sparsity Ratios. A globally shared sparsity ratio cannot capture heterogeneous compression needs across layers, experts, and modules. Therefore, AdaUpMoE introduces a learnable tensor $\boldsymbol{\alpha} \in \mathbb{R}^{L \times E \times P}$ to parameterize the sparsity ratios (Louizos et al., 2018), where each entry $\alpha_{l,e,p}$ is an unconstrained scalar assigned to the delta $\mathbf{\Delta}_{l,e,p}$ and mapped through the sigmoid function $\sigma(\cdot)$ as

$$\theta_{l,e,p} = \sigma\left(\alpha_{l,e,p}\right), \qquad k_{l,e,p} = 1 - \theta_{l,e,p}, \tag{2}$$

where $k_{l,e,p}$ denotes the corresponding keep rate. At each forward pass, a hard binary mask $\mathbf{M}^{\text{hard}}_{l,e,p} \sim \text{Bernoulli}\left(k_{l,e,p}\right)$ is sampled for the delta, where each of its $R_p \times C_p$ elements is drawn independently under the shared keep rate $k_{l,e,p}$. A run-level seed makes the sampling reproducible. The sampling process is not differentiable, so we adopt the straight-through estimator (Bengio et al., 2013), which applies $\mathbf{M}^{\text{hard}}_{l,e,p}$ in the forward pass while assigning a unit derivative to each mask entry with respect to the scalar keep rate $k_{l,e,p}$ in the backward pass, yielding the effective mask

$$\mathbf{M}_{l,e,p} = \text{sg}\left[\mathbf{M}^{\text{hard}}_{l,e,p} - k_{l,e,p}\right] + k_{l,e,p}, \tag{3}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operation (van den Oord et al., 2017). Thus, the gradient flows through $k_{l,e,p}$ to $\alpha_{l,e,p}$, allowing each component to learn its own sparsity level.

Learnable Rescaling Factors. The analytic factor $1/(1-\theta_{l,e,p}+\varepsilon)$ provides an approximately expectation-preserving starting point but is not necessarily optimal for downstream performance. We therefore learn a positive rescaling factor in log space, preserving the sign of the retained deltas. Specifically, AdaUpMoE introduces a second learnable tensor $\boldsymbol{\beta} \in \mathbb{R}^{L\times E\times P}$, whose entry $\beta_{l,e,p}$ recovers the factor of the corresponding component through the exponential function $s_{l,e,p} = \exp(\beta_{l,e,p})$. We initialize $\beta^{(0)}_{l,e,p} = \log(1/(1-\theta_{l,e,p}+\varepsilon))$ such that $s_{l,e,p}$ initially equals the analytic factor determined by the sparsity ratio learned in Stage 1, where the superscript (0) denotes initialization and ε is a small constant for numerical stability.

Compressed Expert Synthesis. Given the mask and rescaling factor, the compressed delta and the corresponding expert weight are obtained as

$$\widetilde{\boldsymbol{\Delta}}_{l,e,p} = s_{l,e,p}\cdot \mathbf{M}_{l,e,p}\odot \boldsymbol{\Delta}_{l,e,p}, \qquad \widetilde{\mathbf{W}}_{l,e,p} = \mathbf{W}^{\text{base}}_{l,p} + \widetilde{\boldsymbol{\Delta}}_{l,e,p}, \tag{4}$$

where $\odot$ denotes the element-wise product. When $s_{l,e,p} = 1/(1-\theta_{l,e,p}+\varepsilon)$, Eq. (4) reduces to standard random sparsification with analytic rescaling.

3.3 Two-Stage Distillation

AdaUpMoE optimizes the two parameter sets through a two-stage distillation-based learning paradigm. Stage 1 learns the sparsity tensor $\boldsymbol{\alpha}$ under the global target sparsity θ^*, with every component initialized as $\alpha^{(0)}_{l,e,p} = \sigma^{-1}(\theta^*)$. No independent rescaling is learned, and each component uses the analytic factor $1/(1-\theta_{l,e,p}+\varepsilon)$, which is recomputed from the current $\theta_{l,e,p}$ at every step. Stage 2 freezes $\boldsymbol{\alpha}$ and optimizes $\boldsymbol{\beta}$ with learnable rescaling enabled. Bernoulli masks are still applied to the student during training. Throughout both stages, all model weights are kept frozen. Only the policy parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, amounting to $2\cdot L\cdot E\cdot P$ scalars in total, are trained on a small unlabeled calibration set, following the calibration-based post-training compression setting (Li et al., 2021; Shao et al., 2024). The training signal combines three distillation and two regularization terms.

Logit-Level Distillation. Within each training step, we align the compressed student with teacher logits via temperature-scaled KL distillation (Hinton et al., 2015; Li et al., 2023f; Ko et al., 2025). Writing z^{s} for the student logits at a sequence position and $p_T(z) = \text{softmax}(z/T)$, the losses with respect to the original teacher logits z^{t} and the compressed teacher logits z^{ct} are

$$\mathcal{L}^{\text{t}}_{\text{KL}} = T^2\cdot \text{KL}\big(p_T(z^{\text{t}}) \,\big\|\, p_T(z^{\text{s}})\big), \qquad \mathcal{L}^{\text{ct}}_{\text{KL}} = T^2\cdot \text{KL}\big(p_T(z^{\text{ct}}) \,\big\|\, p_T(z^{\text{s}})\big), \tag{5}$$

Both losses are averaged over non-padding token positions. Stage 1 uses only $\mathcal{L}^{\text{t}}_{\text{KL}}$. If Stage 2 relies on the original teacher alone, the learnable rescaling tends to over-scale retained deltas, causing drift from the sparse solution learned in Stage 1. Stage 2 therefore additionally uses a compressed teacher snapshot (Mirzadeh et al., 2020) frozen with the learned sparsity ratios, instantiated masks, and analytic rescaling, yielding $\mathcal{L}^{\text{ct}}_{\text{KL}}$. The original teacher enforces fidelity to the uncompressed model, while the compressed teacher anchors calibration around the feasible solution from Stage 1.

Module-Level Distillation. The global logit signals provide only indirect supervision for an individual component (l,e,p). For each activated component m, i.e., one whose expert receives at least one token in the current batch, we evaluate its frozen original weight on the same input, obtaining $\mathbf{h}_m$ as a detached local target for the compressed output $\widehat{\mathbf{h}}_m$ (Romero et al., 2015). Both values are standardized and aligned with Smooth L1 loss:

$$\mathcal{L}_{\text{mod}} = \frac{1}{|\mathcal{A}|}\sum_{m\in\mathcal{A}} \text{SmoothL1}\Big(\text{N}_m\Big(\widehat{\mathbf{h}}_m\Big),\ \text{N}_m(\mathbf{h}_m)\Big), \tag{6}$$

where $\mathcal{A}$ is the set of activated components and N_m standardizes each output using the mean and standard deviation of the detached $\mathbf{h}_m$. Averaging over activated components rather than tokens prevents frequently routed experts from dominating the loss, mitigating routing imbalance.

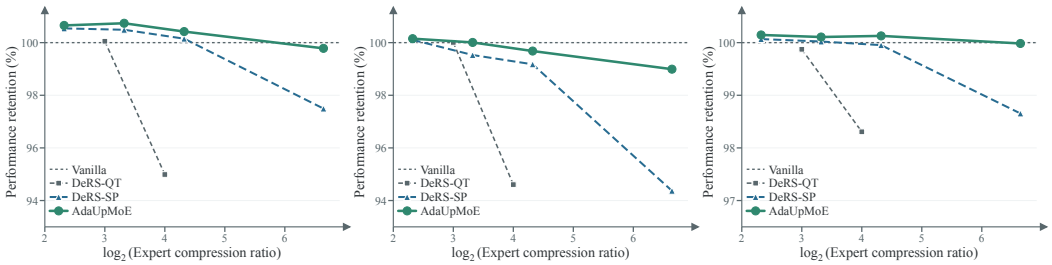


Figure 4: Performance retention on MoE-LLaVA (Lin et al., 2026) under expert-delta compression. Left to right: StableLM (3.08B), Qwen (3.48B), and Phi (4.50B). Scores average seven benchmarks (Lu et al., 2022; Singh et al., 2019; Hudson & Manning, 2019; Li et al., 2023e; Yu et al., 2024b; Liu et al., 2024b; Goyal et al., 2019), normalized by the uncompressed model, and the dashed line marks 100% retention.

Table 1: Performance comparison on Uni-MoE (upcycled from Vicuna) for the general multimodal task. Ratio denotes the nominal compression ratio applied to expert-specific deltas, excluding shared base and non-MoE weights. Avg. is the mean score. $^{\text{A}}$ denotes audio versions. Benchmark sources: Marino et al. (2019); Schwenk et al. (2022); Goyal et al. (2019); Liu et al. (2024b); Li et al. (2023e;b); Lai et al. (2017). EHSL denotes the English High School Listening tests from Uni-MoE. ↑ denotes the Avg improvement over DeRS-SP at the same ratio.

Method	Ratio	OKVQA	AOKVQA	VQA$^{\text{v2}}$	MMBench$^{\text{A}}$	POPE	SEED	RACE$^{\text{A}}$		EHSL		Avg
								Middle	High	Long	Short	
Uni-MoE *(Vicuna as Dense)* **37.06B** (Li et al., 2025)												
Vanilla	–	75.88	76.77	86.61	60.25	86.37	66.43	62.47	64.38	48.67	42.00	66.98
DeRS-QT	8×	74.97	74.41	84.82	60.56	85.51	63.21	58.36	57.78	47.33	40.00	64.70
DeRS-QT	16×	0.96	0.96	0.03	2.17	67.10	0.14	0.07	0.00	0.00	0.00	7.14
DeRS-SP	10×	70.89	72.49	84.62	47.83	84.64	62.39	37.05	36.16	36.00	44.00	57.61
DeRS-SP	20×	38.92	38.69	67.68	9.32	76.25	32.39	5.08	2.32	1.33	8.00	28.00
DeRS-SP	100×	0.79	1.48	0.00	1.24	50.49	0.45	0.00	0.00	0.00	0.00	5.45
AdaUpMoE	10×	72.97	72.14	83.06	56.52	85.23	59.23	54.60	53.03	36.00	48.00	62.08 (↑4.47)
AdaUpMoE	20×	71.09	69.17	80.61	55.28	83.95	55.00	52.09	49.71	38.00	48.00	60.29 (↑32.29)
AdaUpMoE	100×	66.39	61.75	75.99	49.07	80.10	49.60	46.17	41.28	40.00	47.30	55.77 (↑50.32)

Sparsity and Rescaling Regularization. Stage 1 uses $\mathcal{L}_{\text{sp}}$ to keep the average learned sparsity close to the global target sparsity θ^*. After the sparsity parameters are frozen, Stage 2 uses $\mathcal{L}_{\text{rs}}$ to regularize each rescaling factor around its analytic initialization $s^*_{l,e,p} = 1/(1-\theta_{l,e,p}+\varepsilon)$:

$$\mathcal{L}_{\text{sp}} = \left(\frac{1}{L \cdot E \cdot P} \sum_{l,e,p} \theta_{l,e,p} - \theta^* \right)^2, \qquad \mathcal{L}_{\text{rs}} = \frac{1}{L \cdot E \cdot P} \sum_{l,e,p} \left(\frac{s_{l,e,p}}{s^*_{l,e,p}} - 1 \right)^2. \tag{7}$$

Overall Objective. The total loss is a weighted combination of the five terms:

$$\mathcal{L} = \lambda_{\text{t}} \mathcal{L}^{\text{t}}_{\text{KL}} + \lambda_{\text{ct}} \mathcal{L}^{\text{ct}}_{\text{KL}} + \lambda_{\text{mod}} \mathcal{L}_{\text{mod}} + \lambda_{\text{sp}} \mathcal{L}_{\text{sp}} + \lambda_{\text{rs}} \mathcal{L}_{\text{rs}}, \tag{8}$$

where Stage 1 sets $\lambda_{\text{ct}} = \lambda_{\text{rs}} = 0$ and Stage 2 sets $\lambda_{\text{sp}} = 0$.

3.4 Storage and Deployment

For deployment, sparse positions are regenerated from the learned keep rates and run-level seed in a deterministic component order. Expert weights are synthesized on demand by adding the masked and rescaled deltas to the shared base weights. This design targets storage reduction rather than inference acceleration. Further details are provided in the appendix.

4 Experiments

We evaluate AdaUpMoE on general multimodal, medical multimodal, and code-generation tasks, covering five upcycled MoE families and eight model variants. Throughout, we follow the original

evaluation protocol of each model and compress its MoE expert parameters, which dominate the parameter budget of these models. A nominal ratio of $r\times$ corresponds to a global target sparsity $\theta^* = 1 - 1/r$ on expert-specific deltas, e.g., $\theta^* = 0.99$ for $100\times$. Vanilla denotes the original uncompressed upcycled MoE model. Model, metric, and calibration details are in the appendix.

Calibration Protocol. Following calibration-based post-training compression (Frantar & Alistarh, 2023; Sun et al., 2024), we calibrate compression policies on 100 unlabeled target-task inputs per model, which gives a unified protocol across image, audio, speech, medical-image, and code tasks that text-only corpora such as C4 (Raffel et al., 2020) or WikiText (Merity et al., 2017) do not cover. The appendix assesses sensitivity to calibration–evaluation overlap through held-out benchmarks, leave-one-benchmark-out calibration, and held-out calibration data (LLaVA-665K (Liu et al., 2024a)). In all three checks, AdaUpMoE performs comparably to the default setting and outperforms DeRS-SP.

4.1 General Multimodal Task

Experimental Setup. We evaluate two representative families: MoE-LLaVA (Lin et al., 2026), a partially sparse vision-language family upcycled from three language backbones, and Uni-MoE (Li et al., 2025), a fully sparse multimodal model with modality-specific experts. We report seven vision-language benchmarks for MoE-LLaVA and ten vision-, audio-, and speech-augmented benchmarks for Uni-MoE.

Results and Analysis. For MoE-LLaVA and Uni-MoE, we compare AdaUpMoE with the two upcycled MoE compression families introduced by DeRS (Huang et al., 2025b): both decompose each expert into shared base weights and an expert-specific delta, then compress the delta by quantization or sparsification. We refer to the two variants as DeRS-QT and DeRS-SP, respectively. The MoE-LLaVA results are visualized in Figure 4, and the Uni-MoE results are reported in Table 1. Their nominal ratios are not directly comparable. DeRS-QT is capped at $16\times$ by the BF16 bit width, whereas DeRS-SP and AdaUpMoE can target arbitrary ratios.

As shown in Figure 4, AdaUpMoE hardly affects the performance of the three MoE-LLaVA models, remaining close to or above the vanilla averages at $10\times$ and $20\times$ and outperforming DeRS-SP on every ratio–variant combination, with the largest margin at the extreme $100\times$ compression. Uni-MoE is harsher. Its eight modality-specific experts are first fine-tuned separately and then upcycled to initialize joint multimodal fine-tuning, which likely yields larger and less redundant expert deltas. As Table 1 shows, DeRS-QT collapses at its $16\times$ ceiling, while DeRS-SP already drops 9.37 points at $10\times$ and falls to 5.45 at $100\times$. AdaUpMoE instead degrades gracefully, retaining 83% of the vanilla average at $100\times$, where no baseline remains functional. It also approaches DeRS-SP at $10\times$ while storing only $\frac{1}{10}$ as many expert delta parameters.

4.2 Medical Multimodal Task

Experimental Setup. We evaluate on Med-MoE (Jiang et al., 2024), which upcycles StableLM and Phi medical vision-language backbones and retains a universal FFN branch alongside top-1-routed experts. The compressed models are tested on three medical VQA datasets (Lau et al., 2018; Liu et al., 2021; He et al., 2020), with accuracy for closed-ended and recall for open-ended questions.

Results and Analysis. We compare AdaUpMoE with DeRS-QT and DeRS-SP on Med-MoE, with results reported in Table 2. Compression is milder on this task. Both baselines stay within 1–2 points of the vanilla model across feasible ratios, likely because Med-MoE retains an uncompressed universal FFN branch alongside the routed experts. The relative ordering is still clear. DeRS-QT drops by about two points at $16\times$, DeRS-SP degrades visibly at $100\times$, whereas AdaUpMoE is the only method that exceeds the vanilla model at every ratio. At $100\times$, AdaUpMoE attains the best overall score on StableLM, remains above the vanilla model on Phi, and outperforms DeRS-SP on both variants. The gains mainly come from open-ended questions, with VQA-RAD (Lau et al., 2018) Open improving by nearly 5 points on StableLM at $100\times$.

Table 2: Performance comparison on Med-MoE (upcycled from StableLM) for the medical multimodal task. Phi results are in the appendix. Ratio denotes the nominal compression ratio applied to expert-specific deltas, excluding shared base and non-MoE weights. ↑ denotes the Avg improvement over DeRS-SP at the same ratio.

Method	Ratio	VQA-RAD		SLAKE		PathVQA		Avg
		Open	Closed	Open	Closed	Open	Closed	
Med-MoE *(StableLM as Dense)* **3.61B** (Jiang et al., 2024)								
Vanilla	–	47.64	81.62	83.54	85.10	33.38	90.80	70.35
DeRS-QT	8×	52.67	81.25	83.72	84.62	33.08	90.83	71.03
DeRS-QT	16×	47.67	79.78	79.36	83.17	30.91	89.71	68.43
DeRS-SP	10×	47.25	82.35	83.59	84.38	33.29	90.98	70.31
DeRS-SP	20×	46.57	82.72	82.88	84.86	33.12	90.95	70.18
DeRS-SP	100×	44.60	80.88	82.75	85.10	32.71	90.68	69.45
AdaUpMoE	10×	51.39	82.72	83.84	84.13	33.25	90.74	71.01 (↑0.70)
AdaUpMoE	20×	51.02	81.99	83.85	84.86	33.26	91.04	71.00 (↑0.82)
AdaUpMoE	100×	51.96	82.35	83.25	85.34	33.48	90.83	71.20 (↑1.75)

Table 3: Performance comparison on $\mathcal{X}$FT (upcycled from DeepSeek-Coder) for Python code generation.
Ratio denotes the nominal compression ratio applied to expert-specific deltas, excluding shared base and non-MoE weights. ↑ denotes the Avg improvement over DeRS-SP at the same ratio.

Method	Ratio	HEval	HEval+	MBPP	MBPP+	Avg
$\mathcal{X}$FT *(DeepSeek as Dense)* **5.14B** (Ding et al., 2024)						
Vanilla	–	65.2	61.0	62.2	52.1	60.12
DeRS-QT	8×	57.3	53.7	59.1	49.1	54.80
DeRS-QT	16×	0.0	0.0	0.0	0.0	0.00
DeRS-SP	10×	63.4	59.1	62.2	52.9	59.40
DeRS-SP	20×	61.0	57.3	62.7	53.1	58.52
DeRS-SP	100×	57.9	53.0	55.1	46.9	53.23
AdaUpMoE	10×	65.2	61.0	63.4	53.1	60.68 (↑1.28)
AdaUpMoE	20×	64.6	59.8	62.9	53.6	60.22 (↑1.70)
AdaUpMoE	100×	64.0	60.4	63.9	53.6	60.48 (↑7.25)

Table 4: Performance comparison on MultiPL-MoE (upcycled from Qwen) for multilingual code generation. Ratio denotes the nominal compression ratio applied to expert-specific deltas, excluding shared base and non-MoE weights. ↑ denotes the Avg improvement over DeRS-SP at the same ratio.

Method	Ratio	HumanEval							MBPP						
		Python	Java	C++	Rust	Go	Ruby	Avg.	Python	Java	C++	Rust	Go	Ruby	Avg.
MultiPL-MoE *(Qwen as Dense)* **10.8B** (Wang et al., 2025)															
Vanilla	–	16.50	10.76	9.32	8.33	6.49	6.83	9.71	25.40	18.39	18.89	12.99	15.78	17.88	18.22
DeRS-QT	8×	15.85	12.03	11.80	4.49	7.79	7.45	9.90	23.56	18.13	16.12	11.86	14.44	15.37	16.58
DeRS-QT	16×	13.41	8.86	9.32	3.21	4.55	4.35	7.28	19.30	11.92	17.38	5.37	9.63	9.07	12.11
DeRS-SP	10×	5.50	3.80	4.97	3.21	4.55	3.73	4.29	14.00	7.77	10.58	5.08	10.16	12.09	9.95
DeRS-SP	20×	3.00	1.27	2.48	1.28	1.30	1.24	1.76	0.80	4.66	3.02	1.98	3.48	5.54	3.25
DeRS-SP	100×	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AdaUpMoE	10×	15.90	10.76	9.94	5.77	5.19	6.83	9.07 (↑4.78)	26.20	13.99	17.93	12.43	15.78	15.87	17.03 (↑7.08)
AdaUpMoE	20×	16.50	10.13	11.18	5.13	5.84	4.35	8.85 (↑7.09)	24.90	15.03	15.87	12.71	13.90	10.83	15.54 (↑12.29)
AdaUpMoE	100×	17.70	8.86	10.56	4.49	4.55	4.35	8.42 (↑8.42)	23.50	16.06	17.88	8.76	11.23	4.79	13.70 (↑13.70)

4.3 Code Generation Task

Experimental Setup. We evaluate $\mathcal{X}$FT (Ding et al., 2024) and MultiPL-MoE (Wang et al., 2025), which cover token-level and hybrid token/segment-level routing. $\mathcal{X}$FT is evaluated on HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and their EvalPlus variants (Liu et al., 2023c), and MultiPL-MoE on MultiPL-E (Cassano et al., 2023) across six programming languages, all reported as pass@1.

Results and Analysis. We compare AdaUpMoE with DeRS-QT and DeRS-SP on $\mathcal{X}$FT and MultiPL-MoE. Tables 3 and 4 report the results, respectively. On $\mathcal{X}$FT, DeRS-QT drops by over 5 points at $8\times$ and collapses at $16\times$, whereas AdaUpMoE matches or exceeds the vanilla model at every ratio and remains above it at $100\times$.

On MultiPL-MoE, DeRS-SP collapses at $100\times$, whereas AdaUpMoE retains about 93% and 79% of vanilla performance at $10\times$ and $100\times$, respectively, with only 1.97B of 10.8B parameters.

4.4 Ablation and Analysis

Effect of Compression Policy Sharing Granularity. All learned variants share the same global target sparsity, calibration data, and optimization setup and are evaluated at $100\times$ on three MoE-LLaVA variants. As shown in Figure 5(a), the full $\boldsymbol{L} \times \boldsymbol{E} \times \boldsymbol{P}$ policy outperforms DeRS-SP and achieves the highest average score on all three variants. Per-benchmark results are provided in the appendix, where Figures 6 and 7 further visualize representative learned sparsity ratios.

Effect of Two-Stage Distillation. Table 5 compares four ways of learning θ and s at $100\times$. $\boldsymbol{\theta \to s}$ is best or tied-best on all five model variants, whereas $\boldsymbol{\theta}$**-only** falls below DeRS-SP on MoE-LLaVA.

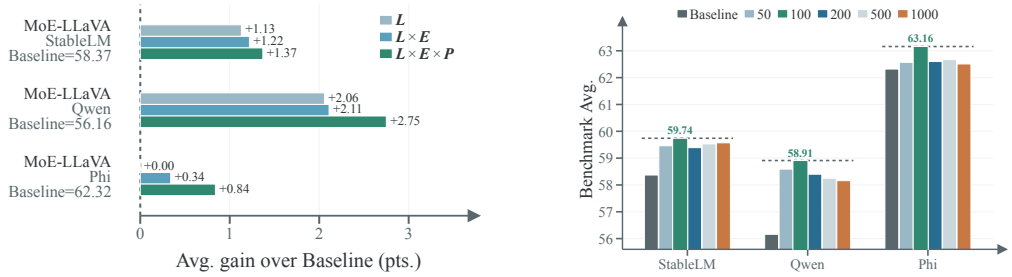


Figure 5: Ablations at 100× on MoE-LLaVA. **(a)** Compression-policy sharing granularity. Bars show average-score gains over DeRS-SP after replacing fixed global sparsification with learned policies shared at the layer (L), layer–expert ($L \times E$), or layer–expert–module ($L \times E \times P$) level. $L \times E \times P$ is the default setting. **(b)** Calibration-data size. Bars show the benchmark mean score for each model variant, and colors denote the number of unlabeled calibration examples. Dashed lines mark the default setting, which achieves the best mean score for all model variants.

With the global target sparsity θ^* fixed at 99%, lowering $\theta_{l,e,p}$ for some components pushes others toward 100%, where the analytic factor $1/(1 - \theta_{l,e,p} + \varepsilon)$ explodes, reaching about 829 at the highest learned sparsity of 99.88%. Learning s after θ removes this coupling, and Figures 8–10 in the appendix show that the learned factors stay below the analytic ones on all eight model variants.

Effect of Optimization and Inference Seeds. At 100×, we conduct five runs per seed setting. Med-MoE achieves 72.51–72.82 Avg when varying sparsity-optimization, global-generation, and inference seeds separately, above DeRS-SP's 71.71. On $\mathcal{X}$FT, our lowest Avg across separate optimization- and inference-seed sweeps exceeds DeRS-SP's highest across inference seeds by 5.80 points. Protocols and per-benchmark results are in the appendix.

Table 5: Ablation of the two-stage distillation procedure at 100×, reported by average score. θ-only learns sparsity ratios, s-only learns rescaling factors, $\theta{+}s$ optimizes θ and s at the same time, and the shaded $\theta{\to}s$ column is the default two-stage procedure that learns θ before s.

Baseline	Dense	θ-only	s-only	$\theta{+}s$	$\theta{\to}s$
MoE-LLaVA	StableLM	57.81	58.83	59.43	59.74
	Qwen	51.39	58.82	58.71	58.91
	Phi	60.55	63.08	63.06	63.16
Med-MoE	StableLM	70.88	70.89	70.90	71.20
	Phi	72.67	72.76	72.77	72.77

Effect of Training Objectives. We ablate the training objectives of the two-stage distillation procedure on Uni-MoE under 10×, 20×, and 100× target ratios, with details in the appendix. Original-teacher KL has the largest effect at 10×. Removing module distillation causes the largest performance drops at 20× and 100×, reducing Avg by 26.33 and 55.07 points, respectively. Compressed-teacher KL and rescaling regularization provide smaller additional gains at 100×.

Effect of Calibration-Data Size. We vary the number of unlabeled calibration examples on MoE-LLaVA, as summarized in Figure 5(b) and detailed in the appendix. Across 50–1000 examples at 100×, performance changes little, with 100 achieving the best mean across model variants.

5 Conclusion

We presented AdaUpMoE, an adaptive fine-grained framework for compressing upcycled MoE models by learning component-wise sparsity ratios and rescaling factors from a small unlabeled calibration set. Across general multimodal, medical multimodal, and code-generation tasks, AdaUpMoE consistently improves over upcycled MoE compression baselines, especially under extreme compression where fixed global sparsification often fails. The ablation studies further show that learned component-wise policies, two-stage distillation, and module-level alignment are important for stable compression. Overall, these results suggest that exploiting heterogeneous compression needs across layers, experts, and modules is a practical path toward more storage-efficient upcycled MoE deployment.

Reproducibility Statement

Implementation details, hyperparameters, calibration protocols, and seed settings are provided in the appendix. We will release code and learned compression policies upon publication.

AI Use Statement

We used generative AI tools for coding assistance and language polishing. All AI-assisted code and text were carefully reviewed by the authors. The authors take full responsibility for the accuracy and integrity of the final manuscript, code, and reported results.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pp. 4218–4222. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.520/.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL http://arxiv.org/abs/1308.3432.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Trans. Software Eng.*, 49(7):3675–3691, 2023. doi: 10.1109/TSE.2023.3267446. URL https://doi.org/10.1109/TSE.2023.3267446.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. Task-specific expert pruning for sparse mixture-of-experts, 2022. URL https://doi.org/10.48550/arXiv.2206.00277.

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. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 1280–1297.

Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.70. URL https://doi.org/10.18653/v1/2024.acl-long.70.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://doi.org/10.48550/arXiv.2606.19348.

Yifeng Ding, Jiawei Liu, Yuxiang Wei, and Lingming Zhang. $\mathcal{X}$FT: unlocking the power of code instruction tuning by simply merging upcycled mixture-of-experts. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12941–12955. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.699. URL https://doi.org/10.18653/v1/2024.acl-long.699.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pp. 736–740. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9052990. URL https://doi.org/10.1109/ICASSP40776.2020.9052990.

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, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5547–5569. PMLR, 2022. URL https://proceedings.mlr.press/v162/du22c.html.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar and Dan Alistarh. QMoE: Sub-1-bit compression of trillion parameter models. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/c74b624843218d9b6713fcf299d6d5e4-Abstract-Conference.html.

Li Fu, Shanyong Yu, Siqi Li, Fan Lu, Youzheng Wu, and Xiaodong He. UME: upcycling mixture-of-experts for scalable and efficient automatic speech recognition. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, pp. 1–5. IEEE, 2025. doi: 10.1109/ICASSP49660.2025.10890533. URL https://doi.org/10.1109/ICASSP49660.2025.10890533.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.*, 127(4):398–414, 2019. doi: 10.1007/S11263-018-1116-0. URL https://doi.org/10.1007/s11263-018-1116-0.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. Deepseek-coder: When the large language model meets programming - the rise of code intelligence, 2024. URL https://doi.org/10.48550/arXiv.2401.14196.

Xuehai He, Yichen Zhang, Luntian Mou, Eric P. Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering, 2020. URL https://arxiv.org/abs/2003.10286.

Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: automl for model compression and acceleration on mobile devices. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pp. 815–832. Springer, 2018. doi: 10.1007/978-3-030-01234-2_48. URL https://doi.org/10.1007/978-3-030-01234-2_48.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network, 2015. URL http://arxiv.org/abs/1503.02531.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts llms gains more. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=hheFYjOsWO.

Yongqi Huang, Peng Ye, Chenyu Huang, Jianjian Cao, Lin Zhang, Baopu Li, Gang Yu, and Tao Chen. DeRS: Towards extremely efficient upcycled mixture-of-experts models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 10056–10066. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.00940. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_DeRS_Towards_Extremely_Efficient_Upcycled_Mixture-of-Experts_Models_CVPR_2025_paper.html.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6700–6709. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00686. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

Songtao Jiang, Tuo Zheng, Yan Zhang, Yeying Jin, Li Yuan, and Zuozhu Liu. Med-MoE: Mixture of domain-specific experts for lightweight medical vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 3843–3860. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.221. URL https://doi.org/10.18653/v1/2024.findings-emnlp.221.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 119–132. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1011. URL https://doi.org/10.18653/v1/n19-1011.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine*

Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The stack: 3 TB of permissively licensed source code. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=pxpbTdUEpD.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=T5nUQDrM4u.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 785–794. Association for Computational Linguistics, 2017. doi: 10.18653/V1/D17-1082. URL https://doi.org/10.18653/v1/d17-1082.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot moe compression. *CoRR*, abs/2510.13999, 2025. doi: 10.48550/ARXIV.2510.13999. URL https://doi.org/10.48550/arXiv.2510.13999.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):180251, 2018. doi: 10.1038/sdata.2018.251. URL https://doi.org/10.1038/sdata.2018.251.

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 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. MIMIC-IT: multi-modal in-context instruction tuning, 2023a. URL https://doi.org/10.48550/arXiv.2306.05425.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023b. URL https://doi.org/10.48550/arXiv.2307.16125.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c. URL http://papers.nips.cc/paper_files/paper/2023/hash/5abcdf8ecdcacba028c6662789194572-Abstract-Datasets_and_Benchmarks.html.

Fengfa Li, Hongjin Ji, Yifeng Ding, Lei Ren, and Chen Wei. Dense2moe: Pushing the pareto frontier of on-device llms via unified pruning and upcycling, 2026. URL https://doi.org/10.48550/arXiv.2605.26496.

Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal LLM with co-upcycled mixture-of-experts. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M.

Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/ed1d3d4c64dc1b95332a8cde3f2a0bdf-Abstract-Conference.html.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023d. URL https://proceedings.mlr.press/v202/li23q.html.

Pingzhi Li, Xiaolong Jin, Yu Cheng, and Tianlong Chen. Examining post-training quantization for mixture-of-experts: A benchmark. *CoRR*, abs/2406.08155, 2024b. doi: 10.48550/ARXIV.2406.08155. URL https://doi.org/10.48550/arXiv.2406.08155.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024c. URL https://openreview.net/forum?id=eFWG9Cy3WK.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 292–305. Association for Computational Linguistics, 2023e. doi: 10.18653/V1/2023.EMNLP-MAIN.20. URL https://doi.org/10.18653/v1/2023.emnlp-main.20.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: pushing the limit of post-training quantization by block reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=POWv6hDd9XH.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-MoE: Scaling unified multimodal llms with mixture of experts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(5):3424–3439, 2025. doi: 10.1109/TPAMI.2025.3532688. URL https://doi.org/10.1109/TPAMI.2025.3532688.

Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 1504–1512. AAAI Press, 2023f. doi: 10.1609/AAAI.V37I2.25236. URL https://doi.org/10.1609/aaai.v37i2.25236.

Seng Pei Liew, Takuya Kato, and Sho Takase. Scaling laws for upcycling mixture-of-experts language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/liew25a.html.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. MoE-LLaVA: Mixture of experts for large vision-language models. *IEEE Trans. Multim.*, 28:4408–4419, 2026. doi: 10.1109/TMM.2026.3654458. URL https://doi.org/10.1109/TMM.2026.3654458.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *18th IEEE International Symposium on Biomedical Imaging, ISBI 2021, Nice, France, April 13-16, 2021*, pp. 1650–1654. IEEE, 2021. doi: 10.1109/ISBI48211.2021.9434010. URL https://doi.org/10.1109/ISBI48211.2021.9434010.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning, 2023a. URL https://doi.org/10.48550/arXiv.2306.14565.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 26286–26296. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.02484. URL https://doi.org/10.1109/CVPR52733.2024.02484.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c. URL http://papers.nips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in Computer Science*, pp. 216–233. Springer, 2024b. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 6159–6172. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.334. URL https://doi.org/10.18653/v1/2024.acl-long.334.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In Lun-Wei Ku, Andre

Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12585–12602. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.679. URL https://doi.org/10.18653/v1/2024.acl-long.679.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 3195–3204. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00331. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Marino_OK-VQA_A_Visual_Question_Answering_Benchmark_Requiring_External_Knowledge_CVPR_2019_paper.html.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE ACM Trans. Audio Speech Lang. Process.*, 32:3339–3354, 2024. doi: 10.1109/TASLP.2024.3419446. URL https://doi.org/10.1109/TASLP.2024.3419446.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5191–5198. AAAI Press, 2020. doi: 10.1609/AAAI.V34I04.5963. URL https://doi.org/10.1609/aaai.v34i04.5963.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pp. 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 527–536. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1050. URL https://doi.org/10.18653/v1/p19-1050.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6550.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VIII*, volume 13668 of *Lecture Notes in Computer Science*, pp. 146–162. Springer, 2022. doi: 10.1007/978-3-031-20074-8_9. URL https://doi.org/10.1007/978-3-031-20074-8_9.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

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 *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting GPT-4V for better visual instruction tuning, 2023. URL https://doi.org/10.48550/arXiv.2311.07574.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: hardware-aware automated quantization with mixed precision. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8612–8620. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00881. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_HAQ_Hardware-Aware_Automated_Quantization_With_Mixed_Precision_CVPR_2019_paper.html.

Qing Wang, Xue Han, Jiahui Wang, Lehao Xing, Qian Hu, Lianlian Zhang, Chao Deng, and Junlan Feng. MultiPL-MoE: Multi-programming-lingual extension of large language models through hybrid mixture-of-experts. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 12817–12828. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.686. URL https://doi.org/10.18653/v1/2025.findings-emnlp.686.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router, 2024. URL https://doi.org/10.48550/arXiv.2410.12013.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. Moe-i^2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 10456–10466. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.612. URL https://doi.org/10.18653/v1/2024.findings-emnlp.612.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/yu24p.html.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754. PMLR / OpenReview.net, 2024b. URL https://proceedings.mlr.press/v235/yu24o.html.

Bo Zhao, Boya Wu, and Tiejun Huang. SVIT: scaling up visual instruction tuning, 2023. URL https://doi.org/10.48550/arXiv.2307.04087.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from llama with continual pre-training. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 15913–15923. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.890. URL https://doi.org/10.18653/v1/2024.emnlp-main.890.

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, 2022. URL https://arxiv.org/abs/2202.08906.

A Models, Datasets, and Evaluation Protocols

This section details the model architectures, datasets and evaluation metrics for the five upcycled MoE models evaluated in the main paper.

A.1 General Multi-Modal Models

Model Architectures and Training Recipes

The general multi-modal experiments are conducted on two representative MoE frameworks, MoE-LLaVA (Lin et al., 2026) and Uni-MoE (Li et al., 2025). MoE-LLaVA pairs a CLIP-Large (Radford et al., 2021) visual encoder with three language backbones of different scales, obtained by upcycling StableLM-1.6B, Qwen-1.8B, and Phi-2.7B, each of which is first fine-tuned as a dense model on a mixture of MIMIC-IT-256k (Li et al., 2023a), LRV-331k (Liu et al., 2023a), SViT-157k (Zhao et al., 2023), and LVIS-256k (Wang et al., 2023). During upcycling, the FFN in every other transformer block is expanded into an MoE layer containing 4 experts, and each token is routed to the top-2 of them at inference. The upcycled model is then fine-tuned on LLaVA-mix-665k (Liu et al., 2024a), with only the MoE layers trainable. Uni-MoE instead builds a fully sparse backbone on top of Vicuna-7B through a progressive three-stage recipe. The first stage performs cross-modality alignment, training only the modality connectors on audio captions from WavCaps (Mei et al., 2024), AudioCaps (Kim et al., 2019), MELD (Poria et al., 2019), and ClothoV1 (Drossos et al., 2020), together with Common Voice speech data (Ardila et al., 2020), while the LLM remains frozen. The second stage trains eight modality-specific experts separately, each fine-tuned as a dense model on its own combination of LLaVA-Instruct-150K (Liu et al., 2023b), RACE (Lai et al., 2017), LibriSpeech (Panayotov et al., 2015), and the above audio corpora. The final stage assembles these experts into an 8-way MoE layer that replaces the FFN in all 32 transformer blocks and jointly refines them with LoRA (Hu et al., 2022) under top-2 routing, on a mixture of LLaVA-v1.5-665K (Liu et al., 2024a), the Video-Instruct dataset (Maaz et al., 2024), and speech-augmented RACE and LibriSpeech data. On the perception side, it encodes images with CLIP-Large (Radford et al., 2021) and speech with the Whisper-small (Radford et al., 2023) encoder, projecting the former into the language space through an MLP connector and the latter through a Q-Former (Li et al., 2023d) with a linear projection.

Datasets and Evaluation Metrics

For MoE-LLaVA, we follow its original evaluation protocol and assess the compressed models on seven vision–language benchmarks: VQAv2 (Goyal et al., 2019), GQA (Hudson & Manning, 2019), ScienceQA-IMG (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al., 2023e), MMBench (Liu et al., 2024b), and MM-Vet (Yu et al., 2024b). Because MM-Vet requires an LLM-as-a-judge evaluation and we observed some scoring stochasticity with GPT-5.4, we run the judge five times on each fixed model output and report the mean score. For Uni-MoE, evaluation spans ten scores over image and speech understanding. The image benchmarks are OKVQA (Marino et al., 2019), A-OKVQA (Schwenk et al., 2022), VQAv2 (Goyal et al., 2019), POPE (Li et al., 2023e), and SEED-Bench (Li et al., 2023b). The speech benchmarks are MMBench-Audio (Liu et al., 2024b), RACE-Audio (Lai et al., 2017), and the English High School Listening tests (EHSL) (Li et al., 2025). MMBench-Audio and RACE-Audio convert the corresponding textual questions and passages into speech to assess joint image–speech understanding. RACE-Audio contributes Middle and High splits, and EHSL contributes Long and Short splits, so the reported Avg is the mean over these ten scores. Each general multi-modal benchmark is reported with its official metric.

A.2 Medical Multi-Modal Models

Model Architectures and Training Recipes

The medical experiments follow the Med-MoE setting (Jiang et al., 2024). Both variants retain CLIP-Large (Radford et al., 2021) as the vision encoder and use language models specialized for medical image instruction following. StableLM-1.6B and Phi-2.7B are first fine-tuned on the medical language–image instruction-following data of LLaVA-Med (Li et al., 2023c) and then upcycled into MoE backbones. During upcycling, the FFN of every other transformer block is replaced by a

parallel structure containing a universal FFN and a 4-expert MoE layer. The universal FFN remains active for every token, whereas the MoE branch selects the top-1 expert for each input. Their outputs are summed to form the final representation.

Datasets and Evaluation Metrics

For Med-MoE, the upcycled models are fine-tuned and evaluated on three medical VQA datasets: VQA-RAD (Lau et al., 2018), SLAKE (Liu et al., 2021), and PathVQA (He et al., 2020). Each dataset contains both open- and closed-ended question–answer pairs. Accuracy is reported for closed-ended questions, and recall is reported for open-ended questions.

A.3 Code Generation Models

Model Architectures and Training Recipes

For code generation, we compress two MoE models that sparsify their backbones at rather different granularities, $\mathcal{X}$FT (Ding et al., 2024) and MultiPL-MoE (Wang et al., 2025). $\mathcal{X}$FT upcycles DeepSeek-Coder-Base-1.3B (Guo et al., 2024) into a Coder-MoE architecture for code instruction tuning, replacing the FFN in every transformer block with an MoE layer working alongside a universal FFN layer that is shared across all tokens. Each MoE layer contains 4 experts, of which only the top-1 is activated per token. MultiPL-MoE pursues multi-programming-lingual extension instead and augments Qwen1.5-1.8B with a hybrid MoE that routes at two granularities within each layer. The token-level branch is a standard upcycled MoE with 6 experts, one of which is a shared expert isolated in the DeepSeekMoE style (Dai et al., 2024), activating the top-2 per token under a normalized gating scheme. The segment-level branch slices the input sequence into fixed-length windows and adopts expert-choice routing (Zhou et al., 2022), letting each of its 6 experts pick the segments it prefers under a fixed capacity, after which the outputs of the two branches are fused through a weighted sum. The hybrid model is post-pretrained on 3.3B code tokens sampled from The Stack (Kocetkov et al., 2023), with a 9-to-1 ratio between low- and high-resource programming languages, updating only the newly added experts and routers.

Datasets and Evaluation Metrics

For code generation, $\mathcal{X}$FT is evaluated on HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and their EvalPlus variants (Liu et al., 2023c). MultiPL-MoE is evaluated with MultiPL-E (Cassano et al., 2023), which translates HumanEval and MBPP into Rust, Go, Ruby, Python, Java, and C++. All code-generation results are reported as pass@1.

B Compression and Implementation Details

This section specifies the compression scope, calibration data, optimization settings, hardware and software environments, and storage and deployment procedures used in our experiments.

B.1 Compression Scope

Following prior work on MoE expert pruning and upcycled MoE compression, including REAP (Lasby et al., 2025), MCMoE (Huang et al., 2025a), and DeRS (Huang et al., 2025b), we report the nominal expert-delta compression ratio r_Δ when comparing with existing methods. This ratio measures the reduction applied to the expert-specific delta tensors alone. As in those works, untouched components such as attention and layer-normalization layers are left out of the ratio, because counting them would pad the budget with parameters that are never modified and make the numbers incomparable across methods. The retained dense FFN base weights and all non-MoE parameters are excluded for the same reason. r_Δ should therefore not be read as the end-to-end compression ratio of the expert block or of the whole model. All compared methods are reported under this same convention. AdaUpMoE compresses only the MoE expert parameters. Following Eq. (1) of the main paper, the FFN weights of the pretrained dense model are kept uncompressed as the expert-shared base, and the deltas of all experts relative to this base are compressed at the specified ratio. The whole-model storage reduction therefore also counts the retained dense FFN base weights and all non-MoE parameters.

For example, Uni-MoE contains 37.06B parameters, of which 2.41B are non-MoE parameters and 34.65B belong to eight experts. At the nominal 100× setting, the 4.33B dense FFN weights are retained as the base, while the compressed deltas of all eight experts occupy 0.35B. Together with the non-MoE parameters, the stored model occupies 7.09B. This corresponds to an effective whole-model compression ratio of 5.23× and an 80.9% storage reduction. MultiPL-MoE has 10.8B parameters, including 9.74B in the twelve experts of its two six-expert branches and 1.06B in non-MoE parameters. At 100×, the 0.81B dense FFN weights are retained as the base and the compressed deltas of all twelve experts occupy 0.10B. The resulting stored model occupies 1.97B, corresponding to an effective whole-model compression ratio of 5.48× and an 81.8% storage reduction.

B.2 Calibration Data

General Multi-Modal Models

For MoE-LLaVA, we construct a 100-example calibration set by randomly sampling 20 unlabeled image–question pairs from each of ScienceQA-IMG (Lu et al., 2022), GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), POPE (Li et al., 2023e), and MM-Vet (Yu et al., 2024b). Only the input images and questions are used. Answer labels and ground-truth annotations are excluded. This calibration set is small relative to the source benchmarks. For example, the 20 GQA examples account for approximately 0.16% of its 12,578 test-dev examples. Under the adopted protocol, POPE and MM-Vet come only as evaluation splits, so there is no separate training or validation split from which calibration examples could be drawn.

For Uni-MoE, we construct a 100-example calibration set by randomly sampling 25 examples from each of OKVQA (Marino et al., 2019), A-OKVQA (Schwenk et al., 2022), VQAv2 (Goyal et al., 2019), and MMBench-Audio. MMBench-Audio is an audio-converted version of MMBench (Liu et al., 2024b) constructed by the Uni-MoE authors (Li et al., 2025). POPE, SEED-Bench, RACE-Audio, and EHSL contribute no calibration examples.

Medical Multi-Modal Models

For Med-MoE, we construct a 100-example unlabeled calibration set from the three medical evaluation benchmarks, sampling 33 image–question pairs from VQA-RAD, 34 from PathVQA, and 33 from SLAKE. Only the images and questions are used, without answer labels. The 34 PathVQA examples amount to about 0.50% of its 6,761 examples, so calibration touches only a small fraction of the data.

Code Generation Models

For $\mathcal{X}$FT, the calibration set contains 100 tasks, with 50 drawn from HumanEvalPlus and 50 from MBPPPlus.

For MultiPL-MoE, calibration tasks are drawn directly from Python HumanEvalPlus and from the MultiPL-E HumanEval test sets for Rust, Go, Ruby, Java, and C++, roughly 20–30 tasks per language. No tasks are taken from a general code corpus or from MBPP.

Section F.1 reports the calibration leakage checks.

B.3 Two-Stage Optimization

General Multi-Modal Models

The two stages run continuously in the same training process rather than reloading a Stage-1 checkpoint. Let N_1 and N_2 denote the numbers of Stage-1 and Stage-2 steps, respectively. Both stages use a base learning rate of 0.1. This value is the peak of each stage-specific learning-rate schedule rather than a constant instantaneous learning rate throughout training. Both stages use the Adam optimizer. Stage 1 optimizes the learnable sparsity tensor $\boldsymbol{\alpha}$, which parameterizes the learnable component-wise sparsity ratios $\boldsymbol{\theta}$. It uses analytical rescaling and begins with $\min(50, \lfloor \frac{N_1+N_2}{10} \rfloor)$ linear warm-up steps followed by cosine decay parameterized over the total $N_1 + N_2$ steps, with a minimum multiplier of 0.01. At the transition, the analytical factors are recomputed from the learned sparsity tensor $\boldsymbol{\alpha}$, after which optimization of $\boldsymbol{\beta}$ begins.

Stage 2 optimizes this learnable rescaling tensor, which parameterizes the learnable component-wise rescaling factors $\boldsymbol{s}$. The $\boldsymbol{\beta}$ parameters are appended to the existing Adam optimizer, and the Stage-2 base learning rate is set to 0.1. Consequently, Adam's first- and second-moment states for $\boldsymbol{\alpha}$ are preserved, while the newly enabled $\boldsymbol{\beta}$ parameters acquire their optimizer states from their first Stage-2 update. The Stage-1 scheduler is replaced by a fresh Stage-2 scheduler with $\min(50, \lfloor \frac{N_2}{10} \rfloor)$ warm-up steps followed by cosine decay over the N_2 Stage-2 steps, with a minimum multiplier of 0.01. During Stage 2, $\boldsymbol{\alpha}$ remains frozen, and only the learnable rescaling tensor $\boldsymbol{\beta}$ receives gradient updates. Student masks are not carried over from Stage 1. Instead, a new Bernoulli mask is sampled from the current keep rates on each forward pass. When $\lambda_{\text{ct}} > 0$, a frozen compressed-teacher snapshot is constructed once at the stage transition from the Stage-1 sparsity state, analytical rescaling, and one instantiated mask, and the same snapshot is reused throughout Stage 2 rather than rebuilt at every step.

MEDICAL MULTI-MODAL MODELS

Med-MoE follows the same two-stage procedure and schedule as the general multi-modal models. Stage 1 learns component-wise sparsity ratios with analytical rescaling, and Stage 2 learns the rescaling factors. Both stages run continuously in one process with Adam and a base learning rate of 0.1, each under linear warm-up and cosine decay. At the transition the analytical factors are recomputed. During Stage 2 the learned sparsity ratios stay frozen, only the rescaling factors are optimized, and student masks are resampled on every forward pass.

CODE GENERATION MODELS

The code-generation models follow the same two-stage procedure and schedule as above, learning sparsity ratios in Stage 1 and rescaling factors in Stage 2. Both stages use Adam with no model-specific learning-rate adjustments.

B.4 HYPERPARAMETER SETTINGS

Following the total objective in the main paper, λ_{t}, λ_{ct}, λ_{mod}, λ_{sp}, and λ_{rs} weight the original-teacher KL loss, compressed-teacher KL loss, module-level distillation loss, sparsity regularizer, and rescaling regularizer, respectively. These coefficients were chosen as follows. We set the coefficient of the loss term with the largest unweighted value to 1 and pick the others so that the weighted terms sit at fixed proportions of that reference. From there, only a handful of per-model, per-ratio adjustments were needed to account for how strongly each regularizer has to act, which yields the ranges reported below. λ_{ct} is zero in Stage 1 because the compressed teacher only exists after the stage transition, λ_{rs} is absent from Stage 1 because rescaling is analytical there, and λ_{sp} is absent from Stage 2 because the sparsity regularizer is switched off once sparsity is frozen. Unless stated otherwise, Stage 2 keeps the Stage-1 values of λ_{t} and λ_{mod}. All experiments use a batch size of 1, with 30 Stage-1 steps and 70 Stage-2 steps.

GENERAL MULTI-MODAL MODELS

For the three MoE-LLaVA variants (Phi, Qwen, and StableLM) at $5\times$–$100\times$, Stage 1 uses $\lambda_{\text{t}} \in [7, 12]$, $\lambda_{\text{mod}} \in [8.5 \times 10^{-4}, 1.1 \times 10^{-3}]$, $\lambda_{\text{ct}} = 0$, and $\lambda_{\text{sp}} = 1$, and Stage 2 uses $\lambda_{\text{ct}} \in [3, 5]$ and $\lambda_{\text{rs}} \in [5 \times 10^{-5}, 4 \times 10^{-4}]$. For Uni-MoE built on Vicuna at $10\times$–$100\times$, Stage 1 uses $\lambda_{\text{t}} = 10^3$, $\lambda_{\text{ct}} = 0$, $\lambda_{\text{mod}} \in [0.7, 1.5]$, and $\lambda_{\text{sp}} \in [2\times10^3, 3\times10^3]$, and Stage 2 uses $\lambda_{\text{ct}} = 25$ and $\lambda_{\text{rs}} = 2\times10^{-4}$.

MEDICAL MULTI-MODAL MODELS

For the two Med-MoE variants (StableLM and Phi) at $10\times$–$100\times$, Stage 1 uses $\lambda_{\text{t}} \in [4, 10]$, $\lambda_{\text{mod}} \in [8\times10^{-4}, 6\times10^{-3}]$, $\lambda_{\text{ct}} = 0$, and $\lambda_{\text{sp}} = 1$, and Stage 2 uses $\lambda_{\text{ct}} \in [0.5, 5]$ and $\lambda_{\text{rs}} \in [5\times10^{-6}, 2.5\times10^{-4}]$.

CODE GENERATION MODELS

$\mathcal{X}$FT and MultiPL-MoE use the same two-stage objective: Stage 1 combines original-teacher KL, module-level distillation, and sparsity regularization with the compressed-teacher KL switched off,

and Stage 2 turns on the compressed-teacher KL, drops the sparsity regularizer, and calibrates the rescaling factors. For $\mathcal{X}$FT (DeepSeek-Coder) and MultiPL-MoE (Qwen) at $10\times$–$100\times$, Stage 1 uses $\lambda_{\text{t}} = 1$, $\lambda_{\text{ct}} = 0$, $\lambda_{\text{mod}} \in [10, 200]$, and $\lambda_{\text{sp}} \in [10^4, 5 \times 10^4]$, and Stage 2 uses $\lambda_{\text{t}} = 0.7$, $\lambda_{\text{ct}} = 0.3$, and $\lambda_{\text{rs}} = 0.15$ while keeping the Stage-1 λ_{mod}. **Table S6** compares alternative splits of the 100-step budget and supports the 30/70 choice. Giving Stage 2 the larger share matters most at high compression ratios, where the rescaling factors need room to drift away from their analytical initializations. The learned mean rescaling factors in the final configurations do show this drift.

B.5 Experimental Environment

We conduct experiments on NVIDIA RTX 4090, NVIDIA H100, and NVIDIA H200 GPUs. To assess reproducibility across hardware and reduce the chance that observed gains arise from hardware differences, we rerun most setups on multiple GPU types. Due to space limitations, **Table S15** reports only the MultiPL-MoE HumanEval and MBPP results across the three GPU types.

MoE-LLaVA uses a single-GPU CUDA 11.7 environment. Med-MoE and $\mathcal{X}$FT use single-GPU CUDA 12.1 environments, whereas Uni-MoE uses two GPUs under CUDA 12.1. MultiPL-MoE uses CUDA 12.1 with one H100 or H200 GPU, or two RTX 4090 GPUs. All experiments use Conda environments that exactly match those released with the respective original papers, but we cannot guarantee identical GPU hardware. For this reason we re-ran the uncompressed baselines ourselves rather than copying published numbers. Most of the resulting mean scores match or slightly exceed the reported values.

B.6 Storage and Deployment

For deployment, Bernoulli masks are re-sampled from the learned keep rates and the run-level seed used during training, following a deterministic module order to regenerate the sparse positions. During expert synthesis, delta entries outside the regenerated mask are zeroed out, and the retained entries are scaled by the learned rescaling factor. Expert weights are synthesized on demand at inference. This design targets storage reduction rather than inference acceleration. The sparsity regularizer constrains the unweighted mean of the learned component-wise sparsity ratios toward θ^*. Because each mask is instantiated through Bernoulli sampling, its realized sparsity may deviate slightly from its learned rate, with the deviation decreasing as the tensor size grows.

C Detailed Experimental Results

This section provides the complete per-benchmark results omitted or summarized in the main paper, covering three MoE-LLaVA models for general multi-modal tasks and two Med-MoE models for medical multi-modal tasks. The full results for Uni-MoE and the code-generation models are reported in the main paper.

C.1 General Multi-Modal Tasks

Due to space limitations, the MoE-LLaVA results are summarized in Figure 4 of the main paper. The complete results are reported in **Table S1** of this appendix.

Table S1: Performance comparisons on MoE-LLaVA upcycled from StableLM, Qwen, and Phi for general multi-modal understanding. Ratio denotes the nominal compression ratio applied to expert-specific delta tensors, following the DeRS reporting convention. It excludes the retained dense FFN base weights and non-MoE parameters. Avg is the arithmetic mean of SQA-IMG, TextVQA, GQA, POPE, MM-Vet, MMBench, and VQAv2.

Method	Ratio	SQA^{I}	VQA^{T}	GQA	POPE	MM-Vet	$MMBench^{EN}$	VQA^{v2}	Avg
MoE-LLaVA *(StableLM as Dense)* **3.08B** (Lin et al., 2026)									
Vanilla	0	59.37	50.14	60.49	86.75	27.40	58.29	76.63	59.87
DeRS-QT	2x	59.80	50.25	60.38	86.66	27.60	58.23	76.62	59.93
DeRS-QT	4x	59.84	49.94	60.56	86.66	28.50	57.97	76.66	60.02
DeRS-QT	8x	59.44	50.44	60.09	86.91	28.80	57.45	76.41	59.94
DeRS-QT	16x	56.80	47.72	54.36	85.44	28.10	52.05	73.59	56.87
DeRS-SP	5x	62.62	50.11	60.52	86.57	29.70	58.03	76.59	60.59
DeRS-SP	10x	62.02	50.34	60.34	86.46	29.80	58.13	76.56	60.52
DeRS-SP	20x	61.97	49.96	59.90	87.04	28.00	57.25	76.39	60.07
DeRS-SP	100x	58.88	47.84	59.06	86.82	28.20	52.73	75.09	58.37
AdaUpMoE	5x	62.96	50.23	60.24	86.91	30.30	57.92	76.60	60.74
AdaUpMoE	10x	62.72	50.14	60.14	86.92	31.50	58.13	76.43	60.85
AdaUpMoE	20x	62.57	50.39	60.33	87.10	28.50	57.71	76.40	60.43
AdaUpMoE	100x	63.21	49.82	59.21	87.19	27.60	55.22	75.96	59.74
MoE-LLaVA *(Qwen as Dense)* **3.48B** (Lin et al., 2026)									
Vanilla	0	63.16	47.74	61.50	87.81	25.50	54.70	76.18	59.51
DeRS-QT	2x	64.07	47.91	61.19	87.73	26.10	54.60	76.10	59.67
DeRS-QT	4x	63.95	47.74	61.22	87.69	26.60	54.96	76.12	59.75
DeRS-QT	8x	63.78	47.75	61.07	87.80	25.60	54.75	75.91	59.52
DeRS-QT	16x	56.64	44.06	56.81	87.24	25.20	51.43	72.75	56.30
DeRS-SP	5x	62.57	47.71	61.02	87.74	27.90	54.49	76.10	59.65
DeRS-SP	10x	61.87	47.29	61.25	87.74	26.10	54.34	75.99	59.23
DeRS-SP	20x	60.39	46.97	61.13	88.09	26.70	54.13	75.73	59.02
DeRS-SP	100x	58.35	43.32	58.48	86.96	24.40	48.42	73.19	56.16
AdaUpMoE	5x	62.77	47.76	61.38	87.64	27.70	54.70	76.02	59.71
AdaUpMoE	10x	62.57	47.51	61.17	87.67	27.40	54.39	75.96	59.52
AdaUpMoE	20x	62.67	47.41	61.00	87.64	26.70	54.13	75.70	59.32
AdaUpMoE	100x	61.63	47.18	60.22	87.63	27.40	53.40	74.89	58.91
MoE-LLaVA *(Phi as Dense)* **4.50B** (Lin et al., 2026)									
Vanilla	0	68.96	51.16	61.42	87.23	32.10	63.74	77.58	63.17
DeRS-QT	2x	68.67	51.39	61.27	87.43	34.60	63.74	77.57	63.52
DeRS-QT	4x	69.36	51.59	61.40	87.43	35.00	63.84	77.62	63.75
DeRS-QT	8x	68.82	51.34	60.92	87.62	32.20	63.27	77.49	63.09
DeRS-QT	16x	69.56	50.35	58.94	87.06	32.00	60.99	75.83	62.10
DeRS-SP	5x	68.67	51.49	61.34	87.15	34.40	62.91	77.59	63.36
DeRS-SP	10x	67.97	51.19	61.33	87.35	33.90	63.32	77.52	63.23
DeRS-SP	20x	68.91	51.22	61.42	87.18	32.90	62.81	77.52	63.14
DeRS-SP	100x	66.93	50.43	60.69	87.64	31.30	62.44	76.81	62.32
AdaUpMoE	5x	68.77	51.36	61.42	87.21	35.00	63.69	77.60	63.58
AdaUpMoE	10x	69.01	51.48	61.37	87.23	34.20	63.43	77.58	63.47
AdaUpMoE	20x	68.67	51.47	61.52	87.31	35.50	62.70	77.51	63.53
AdaUpMoE	100x	68.22	50.91	61.27	87.27	33.70	63.38	77.40	63.16

C.2 MEDICAL MULTI-MODAL TASK

Table 2 of the main paper reports only the StableLM-as-Dense model for lack of space. **Table S2** adds the Phi-as-Dense model.

Table S2: Performance comparisons on Med-MoE upcycled from StableLM and Phi for medical multi-modal understanding. Ratio denotes the nominal compression ratio applied to expert-specific delta tensors, following the DeRS reporting convention. It excludes the retained dense FFN base weights and non-MoE parameters. Open-ended and closed-ended questions are scored by recall and accuracy, respectively. Avg is the arithmetic mean of the six scores on VQA-RAD, SLAKE, and PathVQA.

Method	Ratio	VQA-RAD		SLAKE		PathVQA		Avg
		Open	Closed	Open	Closed	Open	Closed	
Med-MoE *(StableLM as Dense)* **3.61B**								
Vanilla	0	47.64	81.62	83.54	85.10	33.38	90.80	70.35
DeRS-QT	2x	48.09	81.99	83.51	84.38	33.30	90.92	70.36
DeRS-QT	4x	47.11	82.35	83.59	84.13	33.34	90.95	70.25
DeRS-QT	8x	52.67	81.25	83.72	84.62	33.08	90.83	71.03
DeRS-QT	16x	47.67	79.78	79.36	83.17	30.91	89.71	68.43
DeRS-SP	10x	47.25	82.35	83.59	84.38	33.29	90.98	70.31
DeRS-SP	20x	46.57	82.72	82.88	84.86	33.12	90.95	70.18
DeRS-SP	100x	44.60	80.88	82.75	85.10	32.71	90.68	69.45
AdaUpMoE	10x	51.39	82.72	83.84	84.13	33.25	90.74	71.01
AdaUpMoE	20x	51.02	81.99	83.85	84.86	33.26	91.04	71.00
AdaUpMoE	100x	51.96	82.35	83.25	85.34	33.48	90.83	71.20
Med-MoE *(Phi as Dense)* **6.45B**								
Vanilla	0	56.45	83.46	83.69	86.30	34.43	91.33	72.61
DeRS-QT	2x	56.45	83.09	83.69	86.30	34.39	91.36	72.55
DeRS-QT	4x	56.45	83.09	83.72	86.30	34.34	91.33	72.54
DeRS-QT	8x	59.90	84.56	84.40	85.82	34.27	91.36	73.39
DeRS-QT	16x	54.66	84.19	80.22	82.93	32.03	90.47	70.75
DeRS-SP	10x	55.61	83.46	83.82	86.30	34.22	91.27	72.45
DeRS-SP	20x	55.52	83.46	83.50	86.30	34.34	91.27	72.40
DeRS-SP	100x	54.03	82.35	83.02	85.82	34.13	90.89	71.71
AdaUpMoE	10x	56.83	83.82	83.75	86.30	34.53	91.24	72.74
AdaUpMoE	20x	57.76	83.32	83.50	86.30	34.57	91.06	72.75
AdaUpMoE	100x	56.73	83.82	84.08	86.06	34.59	91.36	72.77

D ABLATION STUDIES

D.1 COMPRESSION GRANULARITY

Table S3 reports the full per-benchmark scores of the three compression granularities on MoE-LLaVA at $100\times$, expanding the averages reported in the main paper. All settings learn sparsity–rescaling pairs, but differ in the extent to which these parameters are shared across layers, experts, and linear modules. The layer-wise setting (L) assigns one pair to each layer and shares it across all experts and modules in that layer. The layer–expert-wise setting ($L \times E$) assigns one pair to each expert, allowing experts within a layer to differ while tying the gate, up, and down projections of the same expert. The component-wise setting ($L \times E \times P$) learns an independent pair for every linear module of every expert, i.e., each gate, up, and down projection. This is the default configuration of AdaUpMoE.

Table S3: Compression-granularity ablation on MoE-LLaVA upcycled from StableLM, Qwen, and Phi at $100\times$. The L, $L \times E$, and $L \times E \times P$ settings share one learned policy per layer, per expert, and per expert projection, respectively. The shaded $L \times E \times P$ rows show the default setting. Avg is the arithmetic mean of SQA-IMG, TextVQA, GQA, POPE, MM-Vet, MMBench, and VQAv2.

Dense Model	Granularity	SQA$^{\mathrm{I}}$	VQA$^{\mathrm{T}}$	GQA	POPE	MM-Vet	MMBench$^{\mathrm{EN}}$	VQA$^{\mathrm{v2}}$	Avg
StableLM	L	62.17	49.98	59.00	87.19	26.80	55.43	75.92	59.50
	$L \times E$	62.67	50.01	58.77	87.24	26.80	55.74	75.90	59.59
	$L \times E \times P$	63.21	49.82	59.21	87.19	27.60	55.22	75.96	59.74
Qwen	L	60.78	46.50	59.92	87.59	25.00	53.09	74.67	58.22
	$L \times E$	60.14	46.45	59.98	87.71	26.10	52.73	74.76	58.27
	$L \times E \times P$	61.63	47.18	60.22	87.63	27.40	53.40	74.89	58.91
Phi	L	68.37	50.39	60.67	87.36	29.10	62.91	77.35	62.31
	$L \times E$	69.06	50.96	61.23	87.31	29.40	63.22	77.41	62.66
	$L \times E \times P$	68.22	50.91	61.27	87.27	33.70	63.38	77.40	63.16

D.2 Two-Stage Distillation

Tables S4 and S5 report the full per-benchmark scores of the four two-stage distillation variants on MoE-LLaVA and Med-MoE at $100\times$, corresponding to the averages in the two-stage distillation ablation table of the main paper. All four variants share the distillation losses and optimization procedure of AdaUpMoE and differ only in which policy variables are learned. θ-only learns module-wise sparsity ratios and sets each module's rescaling factor analytically from its learned ratio. s-only learns module-wise rescaling factors under fixed uniform sparsity. $\theta{+}s$ updates both jointly, and $\theta{\to}s$ is the two-stage procedure used by AdaUpMoE in the main paper.

Table S4: Two-stage distillation ablation on MoE-LLaVA upcycled from StableLM, Qwen, and Phi at $100\times$. The θ-only and s-only settings learn one policy variable, while $\theta{+}s$ learns both jointly. The shaded $\theta{\to}s$ rows show the default procedure that learns sparsity before rescaling. Avg is the arithmetic mean of SQA-IMG, TextVQA, GQA, POPE, MM-Vet, MMBench, and VQAv2.

Dense Model	Setting	SQA$^{\mathrm{I}}$	VQA$^{\mathrm{T}}$	GQA	POPE	MM-Vet	MMBench$^{\mathrm{EN}}$	VQA$^{\mathrm{v2}}$	Avg
StableLM	θ-only	60.63	46.27	58.29	86.70	28.10	50.60	74.08	57.81
	s-only	62.12	50.13	58.30	87.12	28.90	55.17	70.07	58.83
	$\theta{+}s$	62.02	49.98	58.96	87.14	27.10	55.01	75.83	59.43
	$\theta{\to}s$	63.21	49.82	59.21	87.19	27.60	55.22	75.96	59.74
Qwen	θ-only	52.70	32.49	54.54	83.03	28.10	40.36	68.54	51.39
	s-only	61.68	46.63	59.77	87.46	28.40	53.14	74.65	58.82
	$\theta{+}s$	62.92	46.09	59.47	87.63	28.20	52.21	74.45	58.71
	$\theta{\to}s$	61.63	47.18	60.22	87.63	27.40	53.40	74.89	58.91
Phi	θ-only	66.14	45.02	58.47	85.69	33.40	60.94	74.19	60.55
	s-only	68.02	51.79	60.69	87.37	33.70	62.60	77.38	63.08
	$\theta{+}s$	68.96	50.61	60.96	87.47	33.40	62.55	77.44	63.06
	$\theta{\to}s$	68.22	50.91	61.27	87.27	33.70	63.38	77.40	63.16

Table S5: Two-stage distillation ablation on Med-MoE upcycled from StableLM and Phi at $100\times$. The θ-only and s-only settings learn one policy variable, while $\theta{+}s$ learns both jointly. The shaded $\theta{\to}s$ rows show the default procedure that learns sparsity before rescaling. Avg is the arithmetic mean of the six open-ended recall and closed-ended accuracy scores on VQA-RAD, SLAKE, and PathVQA.

Dense Model	Setting	VQA-RAD		SLAKE		PathVQA		Avg
		Open	Closed	Open	Closed	Open	Closed	
StableLM	θ-only	50.50	83.09	83.30	84.13	33.33	90.95	70.88
	s-only	52.16	81.99	82.82	84.38	33.17	90.86	70.89
	$\theta{+}s$	51.37	81.99	82.77	85.34	33.15	90.77	70.90
	$\theta{\to}s$	51.96	82.35	83.25	85.34	33.48	90.83	71.20
Phi	θ-only	56.17	84.19	84.03	86.30	34.11	91.24	72.67
	s-only	56.17	84.56	83.74	86.30	34.54	91.27	72.76
	$\theta{+}s$	57.01	83.82	83.80	86.06	34.54	91.36	72.77
	$\theta{\to}s$	56.73	83.82	84.08	86.06	34.59	91.36	72.77

D.3 Two-Stage Step Allocation

To see how the training budget should be split between the two stages, we vary the numbers of Stage-1 and Stage-2 steps while holding their sum at 100. **Table S6** reports the results on $\mathcal{X}$FT at $100\times$. The split moves all four benchmarks by a few points, and $30 + 70$ gives the best *Avg* of the configurations tried. It is the default used for the main-text results and is shaded in the table.

Table S6: Two-stage step-allocation study on $\mathcal{X}$FT at $100\times$ under a fixed 100-step training budget. The shaded 30-step plus 70-step row is the default allocation for sparsity learning and rescaling calibration. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Stage 1	Stage 2	HumanEval	HumanEval+	MBPP	MBPP+	Avg
10	90	60.37	57.32	61.65	51.88	57.80
20	80	60.98	58.54	63.41	53.13	59.01
30	70	64.00	60.40	63.90	53.60	60.48
40	60	60.98	57.32	62.91	52.38	58.40
50	50	64.02	60.37	62.16	51.63	59.54
60	40	61.59	58.54	61.90	51.13	58.29
70	30	62.80	59.76	63.91	52.88	59.84
80	20	65.24	60.37	62.66	51.88	60.04
90	10	62.80	57.32	61.65	52.13	58.48

D.4 Training Objectives

Table S7 reports the full per-benchmark scores of the training-objective ablations on Uni-MoE under $10\times$, $20\times$, and $100\times$ target ratios. The *w/o original-teacher KL* row removes the KL loss from the original uncompressed MoE teacher while keeping the Stage-2 compressed-teacher snapshot. The *w/o sparsity reg* runs never reach the sparsity implied by the target ratio, so they violate the compression budget and their higher scores are not comparable to the other rows. Zero scores indicate collapsed runs.

Table S7: Training-objective ablation on Uni-MoE at $10\times$, $20\times$, and $100\times$. Each non-default row removes one loss term from the complete method shown by the shaded row. Runs without the sparsity regularizer do not satisfy the target compression budget. Avg is the arithmetic mean of OKVQA, AOKVQA, VQAv2, MMBench-Audio, POPE, SEED, the two RACE-Audio subsets, and the two EHSL subsets.

Method	Ratio	OKVQA	AOKVQA	VQA^{v2}	MMBench^{A}	POPE	SEED	RACE^{A}		EHSL		Avg
								Middle	High	Long	Short	
AdaUpMoE	$10\times$	72.97	72.14	83.06	56.52	85.23	59.23	54.60	53.03	36.00	48.00	62.08
w/o original-teacher KL		69.01	65.50	78.07	54.35	82.05	52.00	50.14	48.34	47.33	32.00	57.88
w/o compress-teacher KL		72.89	72.23	83.20	56.21	85.38	59.38	55.01	52.94	46.67	36.00	61.99
w/o module distill		68.81	72.02	79.07	54.07	83.57	60.85	53.64	49.72	45.33	37.00	60.41
w/o sparsity reg		75.96	77.29	86.48	61.80	86.25	66.04	62.40	62.66	54.00	42.00	67.49
w/o rescale reg		72.87	72.05	82.95	56.21	85.34	59.05	54.81	52.80	48.67	36.00	62.07
AdaUpMoE	$20\times$	71.09	69.17	80.61	55.28	83.95	55.00	52.09	49.71	38.00	48.00	60.29
w/o original-teacher KL		68.27	63.67	77.40	54.35	82.63	50.92	49.51	46.37	47.33	32.00	57.25
w/o compress-teacher KL		70.96	69.52	80.65	55.28	84.26	55.03	51.16	49.89	48.00	38.00	60.28
w/o module distill		43.01	35.66	51.08	28.26	50.40	27.50	32.18	26.89	24.14	20.48	33.96
w/o sparsity reg		75.47	76.33	85.71	59.94	85.77	64.92	59.05	59.18	48.67	38.00	65.30
w/o rescale reg		70.91	68.91	80.63	55.90	84.12	54.89	52.37	49.83	48.67	32.00	59.82
AdaUpMoE	$100\times$	66.39	61.75	75.99	49.07	80.10	49.60	46.17	41.28	40.00	47.30	55.77
w/o original-teacher KL		66.29	62.10	75.98	50.31	80.08	49.42	46.52	41.94	44.67	40.00	55.73
w/o compress-teacher KL		66.17	61.92	76.09	47.83	79.72	49.51	45.19	40.82	46.67	34.00	54.79
w/o module distill		0.00	1.48	0.00	2.17	0.00	0.00	0.00	0.00	1.33	2.00	0.70
w/o sparsity reg		69.92	66.72	79.19	54.35	83.55	53.76	51.60	49.03	47.33	44.00	59.94
w/o rescale reg		66.37	61.75	76.19	48.14	79.52	49.42	44.78	40.31	47.33	38.00	55.18

D.5 Calibration Data Size

Table S8 reports the per-benchmark results obtained with different numbers of unlabeled calibration examples on MoE-LLaVA at $100\times$. The shaded 100-example rows use the default calibration set and reproduce the corresponding results in **Table S1**.

Table S8: Calibration-data-size ablation on MoE-LLaVA upcycled from StableLM, Phi, and Qwen at $100\times$ using 50 to 1000 unlabeled examples. The shaded 100-example rows show the default setting. Avg is the arithmetic mean of SQA-IMG, TextVQA, GQA, POPE, MM-Vet, MMBench, and VQAv2.

Dense Model	Calib.	SQA^{I}	VQA^{T}	GQA	POPE	MM-Vet	MMBench^{EN}	VQA^{v2}	Avg
StableLM	50	63.01	49.94	59.25	87.22	25.64	55.27	75.89	59.46
	100	63.21	49.82	59.21	87.19	27.60	55.22	75.96	59.74
	200	62.12	50.10	59.35	87.05	25.64	55.43	76.04	59.39
	500	62.42	49.85	59.29	87.18	26.88	55.06	75.99	59.53
	1000	62.27	49.77	59.26	87.13	27.02	55.53	76.00	59.57
Phi	50	68.86	50.44	61.33	87.31	30.09	62.55	77.41	62.57
	100	68.22	50.91	61.27	87.27	33.70	63.38	77.40	63.16
	200	69.06	50.06	61.13	87.37	30.21	62.91	77.43	62.60
	500	68.67	50.82	61.29	87.38	30.06	63.01	77.44	62.67
	1000	67.77	50.24	61.12	87.36	31.12	62.55	77.44	62.51
Qwen	50	61.68	46.90	59.87	87.76	25.64	53.40	74.90	58.59
	100	61.63	47.18	60.22	87.63	27.40	53.40	74.89	58.91
	200	61.77	46.72	60.18	87.68	24.50	53.04	74.89	58.40
	500	61.23	46.71	59.80	87.65	24.04	53.40	74.89	58.25
	1000	61.03	46.58	60.03	87.62	23.53	53.40	74.92	58.16

E Learned Compression Policies

This section visualizes the compression policies learned by AdaUpMoE. **Figures 6 and 7** show representative learned sparsity ratios $\theta_{l,e,p}$, **Figures 8 and 9** show representative learned rescaling factors $s_{l,e,p}$, and **Figure 10** summarizes the mean rescaling factors on all eight model variants. Each heatmap covers one MoE layer at the $10\times$, $20\times$, and $100\times$ target ratios, with one cell per expert and linear module.

Sparsity ratios. The learned sparsity ratios differ markedly across experts and modules within the same layer, even though every component is initialized at the global target sparsity θ^*. On MoE-LLaVA (Phi), the fc1 projection of expert E3 keeps 73.34% sparsity at $10\times$, whereas the other cells of the layer lie between 86.08% and 91.94%. On $\mathcal{X}$FT, the down projection of expert E4 is similarly kept much denser than its neighbors at $10\times$ and $20\times$. As the target ratio increases, all cells move toward higher sparsity and the spread narrows, while the component kept densest at $10\times$ remains the densest at $20\times$ in both layers.

Rescaling factors. The learned rescaling factors are also heterogeneous. On MultiPL-MoE (Qwen1.5), experts E1–E5 of layer 20 learn much smaller factors than experts E6–E12 at every ratio. On Uni-MoE (Vicuna), the factors of layer 31 vary across both experts and projections. At $10\times$, most cells lie between 1.6 and 4.2, well below the uniform-target factor of 10, whereas the down projections of experts E4 and E8 reach 11.0 and 7.0. Figure 10 compares, for each model variant and target ratio, the mean analytic factor $1/(1-\theta_{l,e,p}+\varepsilon)$ implied by the learned sparsity ratios with the mean learned factor $s_{l,e,p}$. The learned mean is lower than the analytic mean in all 24 settings, and the gap is largest on Uni-MoE, $\mathcal{X}$FT, and MultiPL-MoE. This indicates that the expectation-preserving factor tends to over-scale the retained deltas, which the second stage of AdaUpMoE corrects.

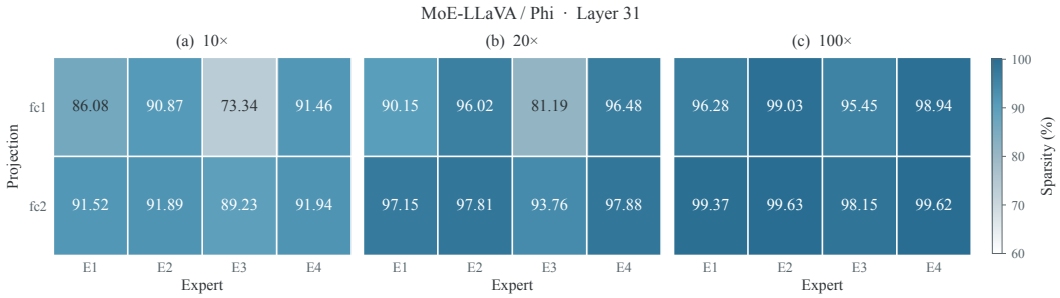


Figure 6: Learned sparsity ratios (%) of layer 31 of MoE-LLaVA upcycled from Phi at **(a)** $10\times$, **(b)** $20\times$, and **(c)** $100\times$. Each cell corresponds to one expert and one linear module.

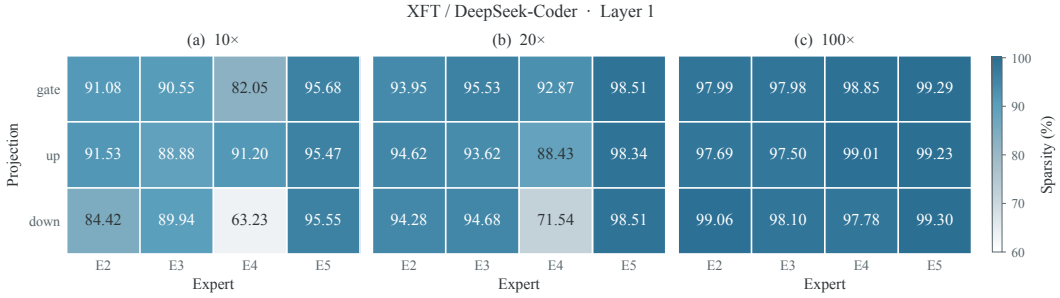


Figure 7: Learned sparsity ratios (%) of layer 1 of $\mathcal{X}$FT upcycled from DeepSeek-Coder at **(a)** $10\times$, **(b)** $20\times$, and **(c)** $100\times$. Each cell corresponds to one expert and one linear module.

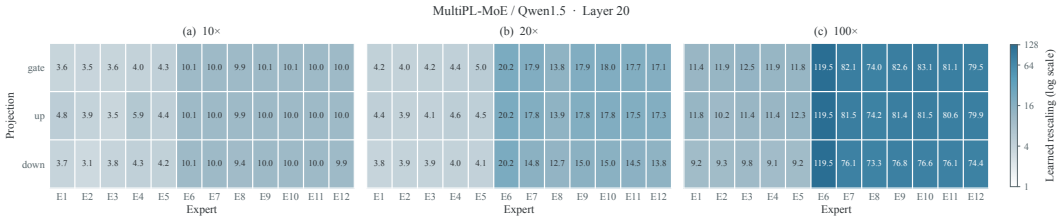


Figure 8: Learned rescaling factors of layer 20 of MultiPL-MoE upcycled from Qwen1.5 at **(a)** $10\times$, **(b)** $20\times$, and **(c)** $100\times$, shown on a logarithmic color scale. Each cell corresponds to one expert and one linear module.

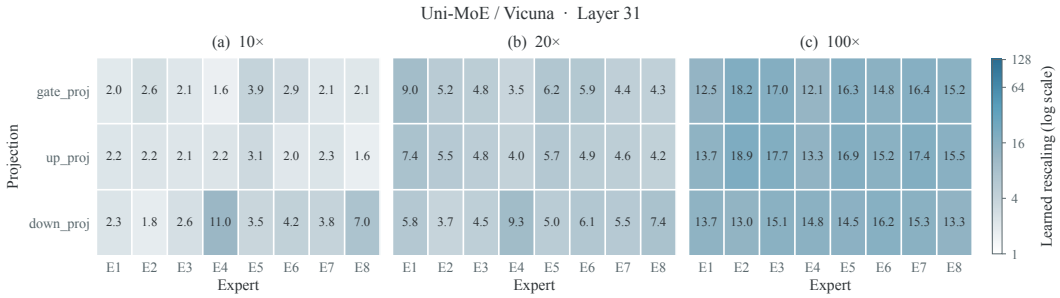


Figure 9: Learned rescaling factors of layer 31 of Uni-MoE upcycled from Vicuna at **(a)** $10\times$, **(b)** $20\times$, and **(c)** $100\times$, shown on a logarithmic color scale. Each cell corresponds to one expert and one linear module.

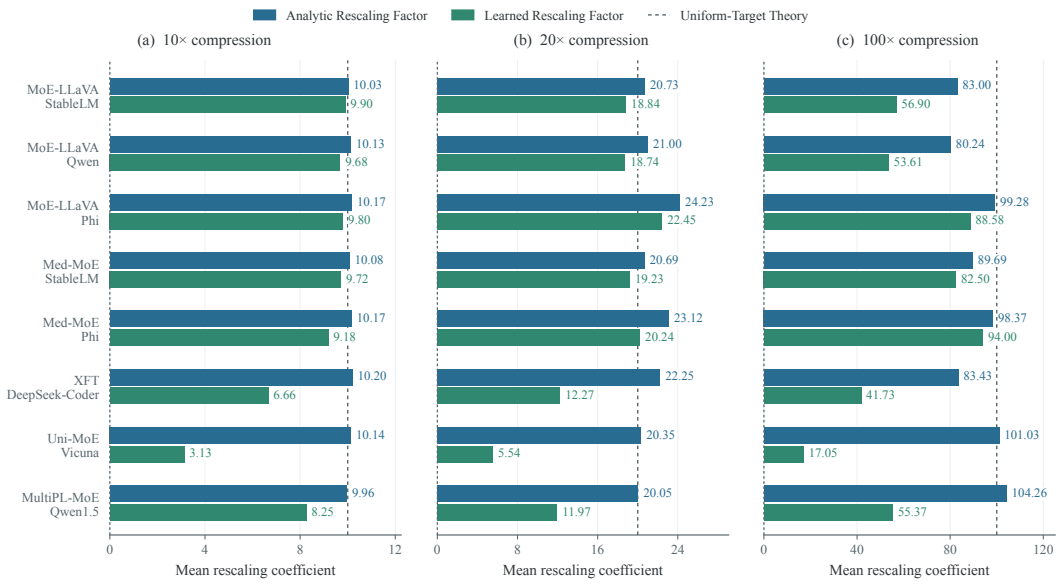


Figure 10: Mean rescaling factors on all eight model variants at **(a)** $10\times$, **(b)** $20\times$, and **(c)** $100\times$. Blue bars show the mean analytic factor $1/(1-\theta_{l,e,p}+\varepsilon)$ under the learned sparsity ratios, green bars show the mean learned factor $s_{l,e,p}$, and dashed lines mark the uniform-target factor $1/(1-\theta^*)$.

F Calibration and Robustness Analyses

F.1 Calibration Leakage Checks

Our calibration sets are drawn from the evaluation benchmarks, which raises a natural worry: the gains might come from overlap between calibration and evaluation inputs rather than from the method itself. We address this with three complementary checks that together cover all five model families. The first check looks at benchmarks that contribute no calibration examples at all. This covers MMBench and VQAv2 for MoE-LLaVA, POPE, SEED-Bench, RACE-Audio, and EHSL for Uni-MoE, and every MBPP result for MultiPL-MoE, whose calibration tasks come only from HumanEval. The second check is leave-one-benchmark-out calibration, in which the evaluated benchmark is removed from the calibration set. It is run on Med-MoE upcycled from Phi and on $\mathcal{X}$FT. The third check replaces the benchmark-derived calibration set altogether with examples from LLaVA-665K, a source unrelated to any evaluation benchmark, and is run on Med-MoE upcycled from StableLM and on Uni-MoE. In every case the advantage over DeRS-SP survives, and in the held-out and LLaVA-665K settings it is essentially unchanged. We take this as evidence that the unlabeled calibration inputs do not leak benchmark-specific information. They contain no answers, and the policy they help learn consists of one sparsity ratio and one rescaling factor per projection, which is far too coarse to memorize individual examples.

A related observation is that AdaUpMoE sometimes scores slightly above the uncompressed MoE, for example on MoE-LLaVA at $5\times$ and $10\times$, on both Med-MoE variants, and on $\mathcal{X}$FT at $10\times$. This is not a leakage artifact either. DeRS-QT is training-free and uses no calibration data at all, yet at $2\times$ to $8\times$ it shows the same effect on MoE-LLaVA and Med-MoE. Our reading is that the expert deltas of an upcycled MoE are fine-tuned from a shared dense base on comparatively small instruction data and therefore carry sample-specific noise. Sparsifying and rescaling the deltas acts as a regularizer on that noise, in the same spirit as delta dropping in DARE (Yu et al., 2024a), and the learned per-component policy simply decides where this regularization can be applied most aggressively. The small gains over the vanilla model appear on held-out benchmarks as well, which is consistent with a regularization effect rather than with fitting the calibration data.

Held-Out Benchmark Results

The general multi-modal results include benchmarks that contribute no calibration examples: MMBench (Liu et al., 2024b) and VQAv2 (Goyal et al., 2019) for MoE-LLaVA. As **Table S1** shows, at $100\times$ on the Qwen backbone AdaUpMoE still improves over DeRS-SP (Huang et al., 2025b) from 48.42 to 53.40 on MMBench and from 73.19 to 74.89 on VQAv2, so overlap alone cannot account for the improvement.

The same holds for Uni-MoE, where POPE, SEED-Bench, RACE-Audio, and EHSL are entirely held out. At $100\times$, AdaUpMoE improves over DeRS-SP from 50.49 to 80.10 on POPE and from 0.45 to 49.60 on SEED-Bench, lifts the Middle/High splits of RACE-Audio from 0.00/0.00 to 46.17/41.28, and lifts the Long/Short splits of EHSL from 0.00/0.00 to 40.00/47.30. None of these visual, audio, or speech benchmarks contributed a single calibration example, so leakage cannot explain the gains. They come from the learned per-component sparsity ratios and rescaling factors themselves.

MultiPL-MoE offers a further held-out case. Its calibration tasks are drawn only from HumanEval, yet the main-paper results on MBPP, which contributed nothing to calibration, show the same picture: at $100\times$ the six-language MBPP average is 13.70 for AdaUpMoE against 0.00 for DeRS-SP, and **Table S15** confirms this on three GPU types.

Leave-One-Benchmark-Out Evaluation

We conduct a leave-one-benchmark-out study for Med-MoE upcycled from Phi at $100\times$. Each AdaUpMoE run calibrates on 50 examples from each of the two retained benchmarks, using the same evaluation setting and metrics as the Med-MoE results in the main paper. Open is the mean recall on open-ended questions, Closed is the yes/no accuracy on closed-ended questions, and Avg is their arithmetic mean. **Table S9** shows that dropping the evaluated benchmark from calibration

costs at most a few tenths of a point relative to calibrating on all three, and the margin over DeRS-SP is preserved in every setting.

Table S9: Leave-one-benchmark-out calibration on Med-MoE upcycled from Phi at $100\times$. The shaded row in each block excludes the named evaluation dataset and calibrates on the other two datasets. Open reports recall, Closed reports accuracy, and Avg is their arithmetic mean.

Benchmark	Calibration Sources	Method	Open	Closed	Avg
VQA-RAD	–	DeRS-SP	54.03	82.35	68.19
VQA-RAD	VQA-RAD + SLAKE + PathVQA	AdaUpMoE	56.73	83.82	70.28
VQA-RAD	SLAKE + PathVQA	AdaUpMoE	56.45	83.82	70.14
SLAKE	–	DeRS-SP	83.02	85.82	84.42
SLAKE	VQA-RAD + SLAKE + PathVQA	AdaUpMoE	84.08	86.06	85.07
SLAKE	VQA-RAD + PathVQA	AdaUpMoE	84.05	85.82	84.93
PathVQA	–	DeRS-SP	34.13	90.89	62.51
PathVQA	VQA-RAD + SLAKE + PathVQA	AdaUpMoE	34.59	91.36	62.98
PathVQA	VQA-RAD + SLAKE	AdaUpMoE	34.43	91.30	62.86

We repeat the study for $\mathcal{X}$FT at $100\times$, whose default calibration set draws on both HumanEval and MBPP. Calibrating on MBPP alone, AdaUpMoE reaches 65.20 on HumanEval and 61.60 on HumanEval+. Calibrating on HumanEval alone, it reaches 63.16 on MBPP and 51.88 on MBPP+. **Table S10** lists these numbers alongside the DeRS-SP baseline and the default $100\times$ setting. In both directions the held-out result stays well above DeRS-SP.

Table S10: Leave-one-benchmark-out calibration on $\mathcal{X}$FT at $100\times$. The unshaded AdaUpMoE rows use HumanEval and MBPP for calibration, while each shaded row excludes the named evaluation dataset. All results are pass@1 percentages and Avg is the arithmetic mean of the two available scores in each block.

Benchmark	Calibration Sources	Method	HumanEval	HumanEval+	MBPP	MBPP+	Avg
HumanEval	–	DeRS-SP	57.90	53.00	–	–	55.45
HumanEval	HumanEval + MBPP	AdaUpMoE	64.00	60.40	–	–	62.20
HumanEval	MBPP	AdaUpMoE	65.20	61.60	–	–	63.40
MBPP	–	DeRS-SP	–	–	55.10	46.90	51.00
MBPP	HumanEval + MBPP	AdaUpMoE	–	–	63.90	53.60	58.75
MBPP	HumanEval	AdaUpMoE	–	–	63.16	51.88	57.52

Held-Out Calibration Data

As a stricter test, we replace the benchmark-derived calibration set entirely with 100 examples sampled from LLaVA-665K (Liu et al., 2024a), a source unrelated to any evaluation benchmark, for Med-MoE upcycled from StableLM and for Uni-MoE. Both models are run at target sparsities of 0.90, 0.95, and 0.99, i.e., nominal $10\times$, $20\times$, and $100\times$ compression of the expert deltas. **Tables S11 and S12** compare the two calibration sources. Med-MoE reaches 70.41, 70.69, and 70.74 Avg at the three ratios against 71.01, 71.00, and 71.20 with default calibration. Uni-MoE reaches 62.12, 60.43, and 55.47 against 62.08, 60.29, and 55.77. The individual EHSL scores move around more than the rest, but the averages stay within 0.6 points of the default setting and remain above DeRS-SP at every ratio.

Table S11: Held-out calibration data for AdaUpMoE on Med-MoE upcycled from StableLM at $10\times$, $20\times$, and $100\times$. Each shaded row uses 100 examples independently sampled from LLaVA-665K, while the default row uses the original benchmark-derived calibration data. Open reports recall, Closed reports accuracy, and Avg is the arithmetic mean of the six scores on VQA-RAD, SLAKE, and PathVQA.

Ratio	Calibration Data	VQA-RAD		SLAKE		PathVQA		Avg
		Open	Closed	Open	Closed	Open	Closed	
$10\times$	Default	51.39	82.72	83.84	84.13	33.25	90.74	71.01
$10\times$	LLaVA-665K	49.03	82.35	83.19	84.13	33.05	90.68	70.41
$20\times$	Default	51.02	81.99	83.85	84.86	33.26	91.04	71.00
$20\times$	LLaVA-665K	50.13	81.99	83.85	83.89	33.29	91.01	70.69
$100\times$	Default	51.96	82.35	83.25	85.34	33.48	90.83	71.20
$100\times$	LLaVA-665K	50.97	81.99	82.86	84.38	33.45	90.80	70.74

Table S12: Held-out calibration data for AdaUpMoE on Uni-MoE at $10\times$, $20\times$, and $100\times$. Each shaded row uses 100 examples independently sampled from LLaVA-665K, while the default row uses the original benchmark-derived calibration data. $^{\mathrm{A}}$ denotes audio versions, and SEED denotes the image subset. Avg follows the main-paper convention and averages ten scores using POPE accuracy (Acc.). POPE F1 is listed for reference only and is not part of Avg.

Ratio	Calibration	OKVQA	AOKVQA	VQA$^{\mathrm{v2}}$	MMBench$^{\mathrm{A}}$	POPE		SEED	RACE$^{\mathrm{A}}$		EHSL		Avg
						F1	Acc.		Middle	High	Long	Short	
$10\times$	Default	72.97	72.14	83.06	56.52	84.93	85.23	59.23	54.60	53.03	36.00	48.00	62.08
$10\times$	LLaVA-665K	73.17	72.58	83.33	55.90	85.01	85.30	59.85	54.53	53.23	47.33	36.00	62.12
$20\times$	Default	71.09	69.17	80.61	55.28	83.74	83.95	55.00	52.09	49.71	38.00	48.00	60.29
$20\times$	LLaVA-665K	71.42	69.52	81.01	54.66	84.00	84.26	55.51	51.95	50.00	50.00	36.00	60.43
$100\times$	Default	66.39	61.75	75.99	49.07	81.77	80.10	49.60	46.17	41.28	40.00	47.30	55.77
$100\times$	LLaVA-665K	66.37	62.27	76.19	48.14	81.84	80.22	49.66	45.47	40.34	48.00	38.00	55.47

F.2 SEED SENSITIVITY

The main-paper experiments keep whatever seeds the original baselines ship with. To check that the conclusions do not hinge on a particular seed, we additionally vary the seeds as follows.

Table S13 reports three settings on Med-MoE upcycled from Phi at $100\times$. Each setting varies one seed, for sparsity optimization, global generation, or inference, and fixes the other two as stated in the table. The inference seed governs the Bernoulli masks sampled at inference. Every configuration is run five times with different seeds. Avg spans 72.57 to 72.68 when the sparsity-optimization seed varies and 72.51 to 72.82 when the inference seed varies, while the global-generation seed leaves every score unchanged. All AdaUpMoE runs exceed the DeRS-SP baseline of 71.71, by 0.80 to 1.11 Avg points.

Table S13: Seed sensitivity on Med-MoE upcycled from Phi at $100\times$. The varied and fixed randomness sources are stated in each block. Avg is the arithmetic mean of the open-ended recall and closed-ended accuracy scores on VQA-RAD, PathVQA, and SLAKE.

Setting	Seed	VQA-RAD		PathVQA		SLAKE		Avg
		Open	Closed	Open	Closed	Open	Closed	
DeRS-SP baseline at $100\times$								
DeRS-SP	–	54.03	82.35	34.13	90.89	83.02	85.82	71.71
AdaUpMoE with varying sparsity-optimization seed, global-generation seed 0*, and inference seed* 42								
Sparsity Opt.	Seed 1	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Sparsity Opt.	Seed 2	56.73	83.82	34.32	91.24	83.97	85.82	72.65
Sparsity Opt.	Seed 3	55.99	84.19	34.51	91.21	84.07	86.06	72.67
Sparsity Opt.	Seed 4	56.45	83.82	34.36	91.30	84.02	86.06	72.67
Sparsity Opt.	Seed 5	56.36	83.82	34.59	91.27	83.99	86.06	72.68
AdaUpMoE with varying global-generation seed and sparsity-optimization and inference seeds fixed to 42								
Global Gen.	Seed 1	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Global Gen.	Seed 2	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Global Gen.	Seed 3	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Global Gen.	Seed 4	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Global Gen.	Seed 5	56.17	83.82	34.55	91.27	83.80	85.82	72.57
AdaUpMoE with varying inference seed, sparsity-optimization seed 42*, and global-generation seed* 0								
Inference	Seed 1	56.17	83.82	34.55	91.27	83.80	85.82	72.57
Inference	Seed 2	55.80	83.46	34.55	91.33	83.90	86.06	72.52
Inference	Seed 3	56.27	83.09	34.59	91.30	84.00	85.82	72.51
Inference	Seed 4	56.17	83.82	34.59	91.18	83.21	86.06	72.51
Inference	Seed 5	57.62	84.93	34.44	91.27	83.08	85.58	72.82

For $\mathcal{X}$FT, **Table S14** reports five runs, Seed 1 through Seed 5, for each setting. DeRS-SP varies its inference seed, while AdaUpMoE varies the optimization seed and the inference seed separately, holding the other fixed as stated in the table. DeRS-SP spans 52.88 to 53.33 Avg, whereas AdaUpMoE spans 59.31 to 60.68 under the optimization-seed sweep and 59.13 to 60.48 under the inference-seed sweep, so the worst AdaUpMoE seed still clears the best DeRS-SP seed by about six points.

Table S14: Seed sensitivity of DeRS-SP and AdaUpMoE on $\mathcal{X}$FT at $100\times$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Setting	Seed	HumanEval	HumanEval+	MBPP	MBPP+	Avg
DeRS-SP with varying inference seed at $100\times$						
DeRS-SP	Seed 1	57.9	53.0	55.1	46.9	53.23
DeRS-SP	Seed 2	57.9	53.0	55.1	47.1	53.31
DeRS-SP	Seed 3	56.7	52.4	55.9	47.1	53.04
DeRS-SP	Seed 4	57.3	53.7	55.9	46.4	53.33
DeRS-SP	Seed 5	57.3	51.2	55.1	47.9	52.88
AdaUpMoE with varying optimization seed and inference seed fixed to 0						
Optimization	Seed 1	64.0	60.4	63.9	53.6	60.48
Optimization	Seed 2	64.6	61.0	62.9	52.9	60.35
Optimization	Seed 3	64.0	61.6	61.9	52.1	59.90
Optimization	Seed 4	63.4	59.8	64.9	54.6	60.68
Optimization	Seed 5	62.2	59.8	62.7	52.6	59.31
AdaUpMoE with varying inference seed and optimization seed fixed to 42						
Inference	Seed 1	64.0	60.4	63.9	53.6	60.48
Inference	Seed 2	65.2	61.6	61.7	51.1	59.90
Inference	Seed 3	63.4	59.8	62.2	51.6	59.24
Inference	Seed 4	61.0	57.9	64.2	53.4	59.13
Inference	Seed 5	62.8	61.0	63.1	50.6	59.39

F.3 Hardware Sensitivity

We evaluate MultiPL-MoE at $100\times$ on multilingual HumanEval and MBPP on three GPU types: NVIDIA H200, H100, and GeForce RTX 4090. **Table S15** reports pass@1 for the six programming languages and their average. The H200 rows are the DeRS-SP and AdaUpMoE results from the main paper, and the H100 and RTX 4090 rows repeat the same evaluation with nothing else changed. The averages differ by less than 0.6 points across the three GPUs.

Table S15: MultiPL-MoE performance at $100\times$ on NVIDIA H200, NVIDIA H100, and NVIDIA GeForce RTX 4090 hardware. HumanEval and MBPP are evaluated by pass@1 in Python, Java, C++, Rust, Go, and Ruby, with Avg denoting the six-language arithmetic mean. The NVIDIA H200 entries reproduce the main-paper results.

Hardware	Benchmark	Method	Ratio	Python	Java	C++	Rust	Go	Ruby	Avg
NVIDIA H200	HumanEval	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	HumanEval	AdaUpMoE	$100\times$	17.70	8.86	10.56	4.49	4.55	4.35	8.42
	MBPP	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MBPP	AdaUpMoE	$100\times$	23.50	16.06	17.88	8.76	11.23	4.79	13.70
NVIDIA H100	HumanEval	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	HumanEval	AdaUpMoE	$100\times$	18.30	8.86	9.94	3.85	5.19	5.59	8.62
	MBPP	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MBPP	AdaUpMoE	$100\times$	22.50	13.21	17.13	6.50	12.57	7.05	13.16
NVIDIA GeForce RTX 4090	HumanEval	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	HumanEval	AdaUpMoE	$100\times$	17.70	9.49	9.32	3.85	5.19	4.35	8.32
	MBPP	DeRS-SP	$100\times$	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MBPP	AdaUpMoE	$100\times$	21.70	15.54	15.87	7.06	12.03	7.05	13.21

G Fine-Grained Compression Probes

G.1 Probe Design

A uniform compression policy would be sufficient if the sensitivity to sparsity and rescaling followed a stable pattern across layers, experts, or projections. We test this hypothesis on $\mathcal{X}$FT by probing these three granularities separately. Each probe changes the compression policy of one target while keeping the remaining targets at the default configuration, then evaluates pass@1 on HumanEval, HumanEval+, MBPP, and MBPP+. Their arithmetic mean is denoted by *Avg*. All reported results use training seed 42 and inference seed 0.

In the probes we write the rescaling factor as $s = \eta/(1-\theta)$. The default is $\theta = 0.90$ and $\eta = 1$, i.e., $s = 10$. Sparsity probes set $\theta \in \{0.95, 0.99\}$ with $\eta = 1$, so s follows θ analytically. Rescaling probes keep $\theta = 0.90$ and set $\eta \in \{0.5, 1.5\}$, which gives $s \in \{5, 15\}$. In other words, a rescaling probe perturbs the multiplier η on top of the analytical factor rather than picking 5 or 15 as arbitrary constants.

Figure 2 in the main paper plots ten randomly chosen targets at each granularity. The responses vary widely, yet neither layer depth nor the identity of an expert or projection gives a usable rule for predicting which targets are sensitive. That is what motivates learning θ and s per component instead of prescribing a policy at some fixed level of the hierarchy.

G.2 Layer Probes

Here a target is an entire MoE layer: the gate, up, and down projections of every expert in that layer are modified together. **Tables S16 and S17** report the sparsity probes with $\theta = 0.95$ and $\theta = 0.99$, and **Tables S18 and S19** the rescaling probes with $\eta = 0.5$ and $\eta = 1.5$.

Table S16: Layer-level sparsity probe on $\mathcal{X}$FT with target $\theta = 0.95$ and $\eta = 1$. Each row jointly changes all expert projections in one MoE layer while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	All	56.10	50.61	54.89	45.36	51.74
2	All	57.32	53.05	55.14	45.36	52.72
3	All	52.44	49.39	54.64	47.62	51.02
4	All	55.49	51.22	54.64	46.62	51.99
5	All	54.88	50.61	53.63	44.86	51.00
6	All	54.27	50.61	55.14	46.37	51.60
7	All	48.78	44.51	53.38	45.61	48.07
8	All	54.88	50.00	56.89	47.87	52.41
9	All	49.39	45.12	51.88	43.61	47.50
10	All	55.49	50.00	54.14	45.11	51.18
11	All	56.71	53.66	56.64	46.87	53.47
12	All	49.39	46.95	54.14	46.37	49.21
13	All	55.49	51.22	53.63	44.11	51.11
14	All	56.10	51.83	55.14	45.86	52.23
15	All	56.10	51.83	54.89	45.86	52.17
16	All	54.27	49.39	54.14	44.36	50.54
17	All	55.49	50.00	56.89	47.37	52.44
18	All	55.49	51.83	54.89	45.86	52.02
19	All	51.83	46.95	53.88	44.86	49.38
20	All	55.49	50.61	54.39	46.12	51.65
21	All	55.49	51.83	54.39	44.86	51.64
22	All	53.66	48.78	54.64	46.87	50.99
23	All	56.10	51.22	54.39	45.61	51.83
24	All	51.83	46.34	55.14	46.62	49.98

Table S17: Layer-level sparsity probe on $\mathcal{X}$FT with target $\theta = 0.99$ and $\eta = 1$. Each row jointly changes all expert projections in one MoE layer while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	All	50.61	46.95	51.13	41.60	47.57
2	All	42.68	40.24	51.38	43.86	44.54
3	All	43.29	39.63	51.38	41.85	44.04
4	All	46.34	41.46	48.12	40.85	44.19
5	All	42.07	39.02	53.13	44.36	44.65
6	All	42.68	41.46	49.37	39.85	43.34
7	All	39.02	34.76	47.12	40.35	40.31
8	All	37.20	33.54	45.11	35.84	37.92
9	All	34.15	34.15	47.12	40.10	38.88
10	All	34.15	31.71	46.62	38.85	37.83
11	All	34.76	31.10	41.10	34.09	35.26
12	All	39.63	35.37	44.86	38.10	39.49
13	All	40.85	35.37	42.11	34.09	38.10
14	All	40.24	35.98	41.10	34.84	38.04
15	All	36.59	30.49	43.36	36.84	36.82
16	All	38.41	34.15	43.11	38.10	38.44
17	All	36.59	34.76	45.86	38.60	38.95
18	All	45.73	42.07	48.37	39.85	44.01
19	All	44.51	40.85	48.37	40.85	43.65
20	All	50.61	46.95	49.62	41.85	47.26
21	All	47.56	43.90	48.37	40.35	45.05
22	All	48.78	43.90	50.13	42.36	46.29
23	All	50.00	45.12	50.63	41.35	46.78
24	All	51.22	45.73	50.63	42.61	47.55

Table S18: Layer-level rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 0.5$. Each row jointly changes all expert projections in one MoE layer while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	All	53.66	50.61	54.89	46.62	51.43
2	All	54.27	50.00	56.64	48.37	52.32
3	All	57.93	52.44	55.89	46.12	53.09
4	All	53.66	50.00	56.14	46.37	51.54
5	All	54.88	49.39	57.14	46.87	52.07
6	All	55.49	51.22	56.39	46.62	52.43
7	All	58.54	53.05	55.64	47.37	53.65
8	All	54.27	50.00	56.64	47.12	52.01
9	All	54.27	49.39	57.64	49.12	52.61
10	All	51.22	46.34	55.14	47.37	50.02
11	All	54.88	51.83	57.14	47.37	52.80
12	All	53.66	49.39	55.64	47.62	51.58
13	All	54.27	50.61	56.89	46.62	52.10
14	All	54.88	50.00	54.89	44.61	51.09
15	All	54.27	50.61	55.64	46.12	51.66
16	All	53.05	48.78	57.14	46.87	51.46
17	All	54.27	48.17	55.64	46.62	51.17
18	All	53.05	47.56	57.14	47.12	51.22
19	All	54.27	48.78	54.89	47.37	51.33
20	All	56.71	50.00	55.89	46.87	52.37
21	All	56.71	49.39	55.14	46.37	51.90
22	All	54.27	48.78	55.64	46.37	51.26
23	All	57.32	52.44	54.39	46.37	52.63
24	All	57.32	51.83	55.39	46.12	52.66

Table S19: Layer-level rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 1.5$. Each row jointly changes all expert projections in one MoE layer while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	All	53.66	50.61	54.89	46.62	51.43
2	All	53.66	50.61	53.13	45.36	50.69
3	All	56.10	51.83	54.14	45.86	51.98
4	All	53.66	47.56	56.14	45.86	50.81
5	All	52.44	47.56	55.39	45.61	50.25
6	All	51.83	46.95	54.14	46.12	49.76
7	All	49.39	46.95	55.64	46.12	49.52
8	All	53.66	48.17	53.38	44.61	49.96
9	All	50.00	45.12	52.63	43.36	47.78
10	All	50.61	46.34	53.88	44.61	48.86
11	All	50.61	45.12	52.13	43.86	47.93
12	All	56.71	51.83	54.64	45.11	52.07
13	All	54.88	50.61	52.88	45.61	51.00
14	All	51.83	46.34	54.39	45.86	49.61
15	All	56.10	50.61	55.14	46.62	52.12
16	All	55.49	50.61	53.88	46.12	51.52
17	All	54.27	49.39	56.14	47.62	51.85
18	All	53.05	48.17	55.14	46.62	50.74
19	All	55.49	50.61	54.89	46.12	51.78
20	All	52.44	46.95	54.39	45.61	49.85
21	All	53.05	50.00	55.39	46.87	51.33
22	All	53.05	47.56	54.14	45.11	49.96
23	All	53.66	48.17	55.39	45.86	50.77
24	All	53.05	48.78	54.64	45.86	50.58

G.3 Expert Probes

Here a target is one expert in one layer, and its gate, up, and down projections are modified together. **Tables S20 and S21** report the sparsity probes with $\theta = 0.95$ and $\theta = 0.99$, and **Tables S22 and S23** the rescaling probes with $\eta = 0.5$ and $\eta = 1.5$.

Table S20: Layer–expert sparsity probe on $\mathcal{X}$FT with target $\theta = 0.95$ and $\eta = 1$. Each row jointly changes the gate, up, and down projections of one expert while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	54.27	50.61	54.39	45.61	51.22
1	1	54.88	49.39	55.39	45.86	51.38
1	2	53.66	48.17	55.64	46.62	51.02
1	3	53.66	48.78	56.39	46.62	51.36
1	4	53.66	47.56	55.14	46.12	50.62
5	0	54.27	50.61	54.39	45.86	51.28
5	1	56.10	50.61	55.64	46.12	52.12
5	2	54.88	48.78	55.89	46.12	51.42
5	3	54.27	47.56	55.14	46.12	50.77
11	0	54.88	51.83	55.64	45.86	52.05
11	2	55.49	51.22	55.39	46.37	52.12
11	4	54.27	49.39	55.89	46.37	51.48
16	0	53.66	47.56	54.39	44.86	50.12
16	1	54.27	48.17	54.89	46.12	50.86
16	2	55.49	48.17	54.89	46.37	51.23
16	3	55.49	48.78	54.89	45.61	51.19
16	4	56.71	50.61	54.14	45.36	51.70
20	0	53.66	48.78	54.89	46.12	50.86
20	1	54.88	47.56	55.64	46.62	51.17
20	2	56.10	48.78	55.89	46.87	51.91
20	3	54.27	47.56	56.39	46.62	51.21
20	4	55.49	48.17	55.89	46.37	51.48
24	0	53.66	47.56	54.39	45.36	50.24
24	3	54.88	48.17	56.64	46.62	51.58

Table S21: Layer–expert sparsity probe on $\mathcal{X}$FT with target $\theta = 0.99$ and $\eta = 1$. Each row jointly changes the gate, up, and down projections of one expert while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	54.88	50.00	51.88	42.86	49.90
1	1	54.88	49.39	54.89	46.12	51.32
1	2	55.49	48.78	55.89	46.62	51.69
1	3	53.05	49.39	54.89	46.37	50.92
1	4	54.27	48.17	55.64	46.37	51.11
5	0	41.46	37.80	52.63	44.86	44.19
5	1	54.88	48.17	55.64	46.37	51.26
5	2	53.05	48.78	55.39	46.12	50.83
5	3	54.88	50.61	57.14	47.37	52.50
11	0	34.15	30.49	44.86	37.84	36.84
11	2	56.10	51.22	56.39	46.87	52.64
11	4	53.05	48.78	55.39	46.37	50.90
16	0	42.07	37.20	42.36	36.84	39.62
16	1	54.88	49.39	55.14	46.37	51.44
16	2	53.66	46.95	54.89	45.86	50.34
16	3	54.88	47.56	55.14	45.86	50.86
16	4	56.71	51.22	57.14	46.62	52.92
20	0	47.56	43.90	48.87	41.35	45.42
20	1	54.88	48.17	56.64	47.87	51.89
20	2	54.27	48.78	54.14	45.61	50.70
20	3	53.66	46.95	53.88	45.36	49.96
20	4	54.27	48.78	55.64	46.37	51.26
24	0	52.44	46.34	50.88	43.11	48.19
24	3	51.83	46.95	54.64	45.61	49.76

Table S22: Layer–expert rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 0.5$. Each row jointly changes the gate, up, and down projections of one expert while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	53.66	48.17	54.39	45.86	50.52
1	1	55.49	50.61	55.64	45.86	51.90
1	2	56.71	51.22	56.39	47.12	52.86
1	3	54.27	47.56	56.39	46.62	51.21
1	4	53.66	47.56	55.39	46.12	50.68
5	0	57.32	52.44	57.89	47.62	53.82
5	1	53.66	48.78	55.64	46.62	51.17
5	2	53.05	47.56	54.14	45.86	50.15
5	3	54.88	50.00	55.64	46.37	51.72
11	0	55.49	51.22	56.14	47.12	52.49
11	2	56.10	49.39	55.64	46.87	52.00
11	4	54.27	48.78	55.39	46.37	51.20
16	0	50.00	47.56	56.14	46.12	49.95
16	1	55.49	48.78	54.89	45.86	51.26
16	2	53.05	46.34	54.64	45.86	49.97
16	3	54.27	49.39	54.39	45.36	50.85
16	4	54.27	50.00	55.64	46.62	51.63
20	0	56.10	49.39	56.14	46.87	52.12
20	1	54.27	48.17	55.14	45.86	50.86
20	2	54.27	47.56	55.39	46.37	50.90
20	3	56.10	48.17	54.89	46.12	51.32
20	4	55.49	49.39	55.39	46.62	51.72
24	0	58.54	51.83	55.14	45.61	52.78
24	3	53.66	47.56	55.89	46.37	50.87

Table S23: Layer–expert rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 1.5$. Each row jointly changes the gate, up, and down projections of one expert while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	56.10	49.39	55.14	46.37	51.75
1	1	53.66	48.17	54.64	45.86	50.58
1	2	55.49	48.17	56.14	46.37	51.54
1	3	54.88	50.00	55.89	46.37	51.78
1	4	53.66	47.56	55.64	46.37	50.81
5	0	54.27	50.61	54.64	45.86	51.34
5	1	54.88	48.78	56.14	46.62	51.60
5	2	53.66	48.17	56.89	47.62	51.59
5	3	55.49	50.00	54.64	45.36	51.37
11	0	51.83	46.95	53.13	43.86	48.94
11	2	54.27	50.00	53.88	46.12	51.07
11	4	53.05	46.95	54.89	46.12	50.25
16	0	55.49	50.00	54.39	46.37	51.56
16	1	53.05	46.95	55.14	46.12	50.31
16	2	54.88	47.56	55.39	46.37	51.05
16	3	54.88	49.39	55.39	45.86	51.38
16	4	53.66	48.17	54.14	45.61	50.39
20	0	51.22	46.34	54.89	45.61	49.52
20	1	55.49	48.17	55.14	46.12	51.23
20	2	54.27	48.78	56.14	46.87	51.51
20	3	55.49	48.78	55.14	45.86	51.32
20	4	55.49	48.78	55.14	45.86	51.32
24	0	54.88	50.00	54.89	45.61	51.34
24	3	53.05	47.56	54.64	45.61	50.22

G.4 PROJECTION PROBES

Here a target is a single gate, up, or down projection of one expert. **Tables S24 and S25** report the sparsity probes with $\theta = 0.95$ and $\theta = 0.99$, and **Tables S26 and S27** the rescaling probes with $\eta = 0.5$ and $\eta = 1.5$.

Table S24: Module-cell sparsity probe on $\mathcal{X}$FT with target $\theta = 0.95$ and $\eta = 1$. Each row changes one gate, up, or down projection in a layer–expert pair while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	Module	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	Gate	53.05	47.56	56.14	46.62	50.84
1	0	Up	56.10	51.22	54.64	45.61	51.89
1	1	Gate	53.66	48.17	55.64	46.62	51.02
1	3	Down	53.05	48.78	56.64	46.62	51.27
5	1	Gate	54.88	49.39	55.64	45.86	51.44
5	1	Up	54.88	49.39	56.14	46.62	51.76
5	1	Down	56.10	49.39	55.89	46.37	51.94
5	2	Gate	54.27	50.00	56.14	46.37	51.69
5	2	Up	53.05	49.39	55.39	46.87	51.17
11	0	Down	53.05	47.56	56.89	48.87	51.59
11	1	Down	54.27	47.56	55.39	46.12	50.83
11	2	Gate	54.88	50.61	54.64	45.86	51.50
11	2	Up	55.49	50.00	55.39	46.12	51.75
16	0	Up	53.66	50.00	55.39	46.62	51.42
16	0	Down	56.10	51.22	55.64	46.37	52.33
16	2	Gate	54.27	46.95	55.39	46.87	50.87
16	2	Down	54.27	47.56	54.89	46.12	50.71
16	3	Down	54.27	48.78	55.14	45.86	51.01
16	4	Down	54.27	49.39	54.89	46.12	51.17
20	0	Up	55.49	50.00	53.88	46.37	51.43
20	1	Down	54.88	48.78	55.64	46.37	51.42
20	2	Up	56.71	49.39	54.64	45.61	51.59
24	3	Gate	55.49	48.78	55.89	46.62	51.69
24	3	Down	53.66	47.56	55.39	46.12	50.68

Table S25: Module-cell sparsity probe on $\mathcal{X}$FT with target $\theta = 0.99$ and $\eta = 1$. Each row changes one gate, up, or down projection in a layer–expert pair while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	Module	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	Gate	53.66	49.39	54.14	45.11	50.57
1	0	Up	54.88	50.61	54.39	45.36	51.31
1	1	Gate	54.27	49.39	54.89	45.86	51.10
1	3	Down	53.66	48.78	56.14	46.87	51.36
5	1	Gate	52.44	46.95	56.89	47.12	50.85
5	1	Up	53.66	48.17	54.89	45.61	50.58
5	1	Down	54.88	48.17	54.89	45.11	50.76
5	2	Gate	54.27	50.00	54.64	45.36	51.07
5	2	Up	53.05	48.17	54.64	45.36	50.30
11	0	Down	52.44	46.34	51.63	44.36	48.69
11	1	Down	54.27	48.17	55.39	46.12	50.99
11	2	Gate	56.71	51.83	54.14	45.36	52.01
11	2	Up	52.44	47.56	55.14	46.37	50.38
16	0	Up	53.05	48.17	56.39	46.62	51.06
16	0	Down	55.49	52.44	52.13	44.11	51.04
16	2	Gate	54.27	47.56	54.64	45.86	50.58
16	2	Down	54.27	48.78	54.89	45.86	50.95
16	3	Down	54.88	48.17	55.14	46.12	51.08
16	4	Down	53.66	48.17	56.14	46.87	51.21
20	0	Up	54.88	50.61	54.64	45.11	51.31
20	1	Down	53.66	47.56	56.14	47.12	51.12
20	2	Up	54.27	48.78	55.39	45.86	51.08
24	3	Gate	54.27	48.78	56.39	46.12	51.39
24	3	Down	53.05	46.34	54.64	45.86	49.97

Table S26: Module-cell rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 0.5$. Each row changes one gate, up, or down projection in a layer–expert pair while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	Module	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	Gate	53.66	48.17	55.39	45.86	50.77
1	0	Up	53.66	48.78	55.64	46.37	51.11
1	1	Gate	55.49	48.78	55.14	46.12	51.38
1	3	Down	53.66	47.56	54.89	46.12	50.56
5	1	Gate	54.27	49.39	55.39	46.62	51.42
5	1	Up	54.27	47.56	55.64	46.12	50.90
5	1	Down	53.66	47.56	56.39	46.62	51.06
5	2	Gate	54.27	48.17	55.89	46.87	51.30
5	2	Up	54.27	49.39	54.64	45.61	50.98
11	0	Down	53.05	48.78	55.14	46.37	50.83
11	1	Down	54.27	48.17	55.39	46.12	50.99
11	2	Gate	55.49	50.00	54.64	45.61	51.43
11	2	Up	55.49	50.61	54.64	45.86	51.65
16	0	Up	54.27	50.00	54.64	45.61	51.13
16	0	Down	54.88	48.78	56.14	46.62	51.60
16	2	Gate	54.27	46.95	55.14	46.37	50.68
16	2	Down	54.27	46.95	55.14	46.12	50.62
16	3	Down	55.49	50.00	54.64	45.86	51.50
16	4	Down	56.10	50.61	55.64	46.62	52.24
20	0	Up	55.49	49.39	55.14	45.61	51.41
20	1	Down	56.10	49.39	55.39	46.37	51.81
20	2	Up	54.88	48.17	55.64	46.37	51.26
24	3	Gate	54.88	47.56	56.14	46.62	51.30
24	3	Down	53.66	46.95	55.64	46.37	50.65

Table S27: Module-cell rescaling probe on $\mathcal{X}$FT with target $\theta = 0.90$ and $\eta = 1.5$. Each row changes one gate, up, or down projection in a layer–expert pair while every other target uses the default $\theta = 0.90$ and $\eta = 1$. Avg is the arithmetic mean of the pass@1 scores on HumanEval, HumanEval+, MBPP, and MBPP+.

Layer	Expert	Module	HumanEval	HumanEval+	MBPP	MBPP+	Avg
1	0	Gate	52.44	47.56	55.14	46.12	50.31
1	0	Up	53.66	47.56	56.14	46.62	50.99
1	1	Gate	54.88	48.78	56.39	47.37	51.85
1	3	Down	52.44	47.56	55.64	46.37	50.50
5	1	Gate	53.05	48.17	56.14	46.37	50.93
5	1	Up	54.88	50.61	55.89	46.62	52.00
5	1	Down	54.27	48.78	54.89	46.12	51.01
5	2	Gate	53.66	48.17	55.89	46.37	51.02
5	2	Up	54.88	47.56	56.64	46.87	51.49
11	0	Down	51.83	45.73	55.14	46.12	49.70
11	1	Down	54.88	48.78	55.39	46.12	51.29
11	2	Gate	54.27	47.56	54.64	45.86	50.58
11	2	Up	56.10	50.00	55.39	46.62	52.03
16	0	Up	56.10	50.00	56.64	47.37	52.53
16	0	Down	54.27	49.39	56.14	47.12	51.73
16	2	Gate	54.88	48.17	55.39	46.62	51.26
16	2	Down	54.88	47.56	55.39	46.62	51.11
16	3	Down	56.10	49.39	55.64	46.37	51.87
16	4	Down	53.05	48.17	55.89	46.87	50.99
20	0	Up	53.05	48.17	55.39	46.62	50.81
20	1	Down	54.88	46.95	55.14	45.61	50.65
20	2	Up	53.66	45.73	55.64	46.12	50.29
24	3	Gate	53.66	47.56	55.64	46.37	50.81
24	3	Down	54.27	48.17	55.14	45.86	50.86