When Does Sparse MoE Help in Vision? The Role of Backbone Compute Leverage in Sparse Routing

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) networks promise favorable accuracy–compute trade-offs, yet practical vision deployments are hindered by expert collapse and limited end-to-end efficiency gains. We study when sparse top-k routing with hard capacity constraints helps in vision classification, evaluated under multi-seed protocols on four benchmarks (CIFAR-10/100, Tiny-ImageNet, ImageNet-1K). We observe a *compute-leverage pattern*: positive accuracy gaps require a substantial fraction ρ of total FLOPs to be routed; at ImageNet scale this is necessary but not sufficient, as multi-expert routing ($k \geq 2$) is additionally required. Two controlled experiments isolate these factors. A hidden-size sweep on CIFAR-10 yields both predicted point-estimate sign reversals across standard and depthwise backbones (one statistically supported, one directional), ruling out backbone family alone as the explanation. An ImageNet-1K ablation that varies only top-k—holding architecture, initialization, and ρ fixed—reverses the gap from positive to negative across all five seeds. A per-sample variant of Soft MoE that softmaxes over experts rather than the batch rescues CIFAR-100 above the dense baseline, identifying batch-axis dispatch as the dominant failure mode in per-sample CNN settings. Code is included in the supplementary material.

1 Introduction

Mixture-of-Experts (MoE) architectures route inputs through specialized subnetworks (Jacobs et al., 1991; Jordan & Jacobs, 1994). Sparsely-gated variants—activating only a few experts per input—have scaled to thousands of experts in language (Shazeer et al., 2017; Fedus et al., 2022) and vision (Riquelme et al., 2021; Han et al., 2024) while keeping per-token compute tractable. Despite these advances, two persistent obstacles hinder practical deployments: the shared backbone often dominates total cost, leaving little structural opportunity for sparse routing to improve end-to-end efficiency (Riquelme et al., 2021; Du et al., 2022); and routers frequently collapse onto a subset of experts, undermining specialization (Nie et al., 2022; Chi et al., 2022).

These obstacles are coupled. When the classification head is a negligible fraction of total FLOPs—less than 0.1% on ResNet-18—even perfect routing cannot meaningfully reduce end-to-end cost, and the absence of a clear compute payoff blunts incentives to diagnose collapse or refine routing. *Depthwise separable convolutions* (Chollet, 2017; Howard et al., 2017) shift the head-to-backbone FLOPs ratio toward parity ($\sim$49% on CIFAR), opening a regime in which sparse routing has structural room to help. Combined with hard capacity constraints to enforce per-batch expert usage, this design admits a controlled study of when sparse MoE delivers accuracy gains in vision classification, and what conditions are necessary for those gains to appear.

Our central observation is a compute-leverage pattern for vision MoE: positive accuracy gaps require a substantial fraction ρ of total FLOPs to be routed, and at ImageNet scale this is necessary but not sufficient, with multi-expert routing ($k \geq 2$) additionally required. Depthwise separable backbones tend to raise ρ toward parity, situating the gain mechanism within the joint backbone–routing recipe rather than within either component alone. We support this with two controlled experiments separating the roles of routed compute and routing breadth: a six-configuration hidden-size sweep on CIFAR-10 yields a sign boundary

that tracks ρ across both backbone families, with both predicted point-estimate reversals—a high-ρ *standard* backbone turns positive (statistically supported) and a low-ρ *depthwise* backbone turns negative (directional, not significant), ruling out backbone family alone as the explanation—and a complementary k-ablation on ImageNet-1K (backbone MoE at fixed routed fraction and initialization) varies only the top-k selection and reverses the gap by roughly 3.25% across all five seeds. We evaluate on CIFAR-10/100, Tiny-ImageNet, and ImageNet-1K (Fig. 3); under depthwise co-optimization, gaps are statistically significant on the three smaller-scale benchmarks and at ImageNet only when MoE is placed within backbone convolutions, with the caveat that cross-dataset comparisons span different backbone widths and evaluation protocols. Two analyses support the headline result. Under Switch-style hard-capacity routing (Fedus et al., 2022), temperature scheduling prevents late-stage routing collapse (Section 5.1). A cross-method dispatch diagnostic identifies batch-axis dispatch as the dominant failure mode of Soft MoE in per-sample CNN settings: a per-sample variant softmaxing over experts rather than the batch rescues CIFAR-100 above dense but leaves Tiny-ImageNet and ImageNet-1K negative, indicating a residual, scale-dependent factor we do not isolate.

2 Related Work

Mixture-of-Experts formulations partition the predictive task across specialists blended through a learned router (Jacobs et al., 1991; Jordan & Jacobs, 1994), with sparsely-activated variants scaling to thousands of experts in language (Shazeer et al., 2017; Fedus et al., 2022) and vision (Riquelme et al., 2021; Han et al., 2024) and recent large-scale deployments such as Mixtral (Jiang et al., 2024) demonstrating the approach in practice. Routing collapse—a few experts attracting most of the load—remains an active challenge (Nie et al., 2022; Chi et al., 2022), and Clark et al. (2022) derive unified scaling laws for routed models that formalize the capacity–compute trade-off motivating our compute-leverage observation. Several alternative routing strategies populate this design space: Soft MoE (Puigcerver et al., 2024) replaces discrete assignments with differentiable slot combinations, Expert Choice (Zhou et al., 2022) inverts routing to let experts select tokens, and ST-MoE (Zoph et al., 2022) introduces a router z-loss for training stability. We do not propose a new routing mechanism but compare against these alternatives under matched conditions to characterize when sparse routing helps.

In vision specifically, MoE deployments face a compute imbalance in which backbone layers dominate total cost and leave routing little structural opportunity to help (Riquelme et al., 2021); Liu et al. (2024) report that router design matters more at smaller scales, and Videau et al. (2024) find sparse routing most effective for small- to mid-sized vision models, both consistent with the FLOP-leverage pattern we identify. We leverage depthwise separable convolutions (Chollet, 2017; Howard et al., 2017; Sandler et al., 2018)—which substantially reduce convolutional cost—not as a stand-alone efficiency contribution but as a way to raise the routed FLOP fraction ρ on which the compute-leverage condition depends. Adjacent lines of conditional computation include dynamic networks that adapt per input via early exit, block skipping, or token pruning (Bengio et al., 2013; Han et al., 2022; Wang et al., 2018; Rao et al., 2021) and Mixture-of-Depths (Raposo et al., 2024), which extends sparse routing along the depth axis; fused MoE kernels (Gale et al., 2023; Hwang et al., 2023) address dispatch overhead at deployment, an issue we return to in our latency analysis (Section 5.2). For hyperparameter tuning, we use a narrow four-dimensional evolutionary search (Rechenberg, 1973; Real et al., 2019)—a lightweight alternative to grid or Bayesian methods (Li et al., 2017)—as a development-set protocol rather than a NAS contribution.

3 Methodology

3.1 Architecture Overview

Our architecture comprises a shared convolutional backbone feeding either a dense classifier or an MoE layer (Fig. 1). We study three backbone variants. The *standard backbone* uses two 3×3 conv blocks (24/48 channels) for CIFAR and three blocks (32/64/128) for Tiny-ImageNet, with BatchNorm on Tiny-ImageNet. The *depthwise backbone* replaces each 3×3 convolution with a depthwise 3×3 followed by a pointwise 1×1, scaled by width factor w (Chollet, 2017; Howard et al., 2017). For ImageNet-1K, we use two pretrained

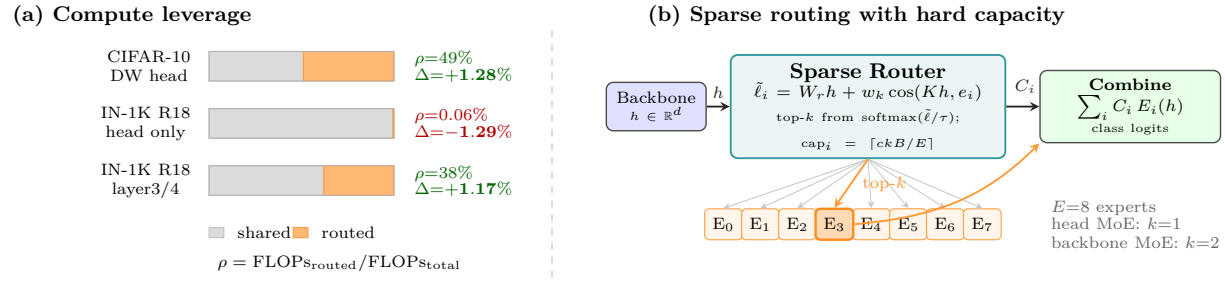

Figure 1: (a) Compute leverage. Bars show shared versus routed FLOPs for the three main operating points: substantial ρ yields positive MoE–dense gaps, while head-only ImageNet routing at ρ<1% is negative. (b) Routing mechanism. The router scores E=8 experts using learned logits plus key similarity, selects top-k under hard per-batch capacity (Eq. 3), and combines selected expert outputs. Head-only MoE uses k=1; backbone MoE (Section 4.6) uses k=2.

backbones: *ResNet-18* (He et al., 2016) (512-dim features) and *MobileNet-V2* (Sandler et al., 2018) (1280-dim features), both after global average pooling.

Experts are MLPs with two hidden layers ($d_{\text{in}} \to h \to h \to d_{\text{out}}$), ReLU activations, dropout 0.3, and hidden widths h set per dataset (304 for CIFAR, 1024 for Tiny-ImageNet, 512 for ImageNet; h=256 sensitivity in Section 4.5).

Throughout the paper, we denote the routed share of total inference FLOPs by $\rho = \text{FLOPs}_{\text{routed}} / \text{FLOPs}_{\text{total}}$. For head-only MoE experiments, $\text{FLOPs}_{\text{routed}}$ is the classifier head; for backbone MoE, it is the replaced convolutional block compute.

3.2 Sparse Routing with Hard Capacity Constraints

The router produces logits $\ell_i = (W_r h)_i$ from input features $h \in \mathbb{R}^{d_{\text{in}}}$ using a learned routing matrix $W_r \in \mathbb{R}^{E \times d_{\text{in}}}$. A learned key-similarity term provides content-based routing: each expert maintains a key embedding $e_i \in \mathbb{R}^{d_k}$, and a projection $K \in \mathbb{R}^{d_k \times d_{\text{in}}}$ maps input features to the same space (d_k=64). The combined routing scores are:

$$\tilde{\ell}_i = \ell_i + w_k \cos(Kh,\, e_i), \quad i = 1, \ldots, E, \tag{1}$$

where w_k=0.5 is the key-similarity weight and $\cos(\cdot,\cdot)$ denotes cosine similarity. The top-k experts are selected from $\text{softmax}(\tilde{\ell}/\tau)$, where τ is the temperature. The routing schedule anneals temperature from $\tau_{\max}$ to $\tau_{\min}$ using either a linear schedule $\tau(t) = \tau_{\max} - (\tau_{\max} - \tau_{\min})\, t/T$ or a sigmoid schedule:

$$\tau(t) = \tau_{\min} + (\tau_{\max} - \tau_{\min}) \cdot \sigma(-\kappa(t/T - 0.5)), \tag{2}$$

where σ is the logistic function and κ=7.0 the sharpness. Per-dataset schedule choices are specified in Section 4.1.

While load-balancing (λ_{lb}) and entropy (λ_{ent}) regularizers encourage uniform expert usage, they do not guarantee balanced per-batch dispatch. Following Switch Transformer (Fedus et al., 2022), we therefore enforce hard capacity limits:

$$\text{cap}_i = \left\lceil c \cdot \frac{k \cdot B}{E} \right\rceil, \quad \forall i \in \{1, \ldots, E\}, \tag{3}$$

where c is the capacity factor, B the batch size, k the top experts selected, and E the total experts. Samples are assigned to their highest-scoring expert with available capacity; if all k candidates are full, the sample is force-assigned to its top-1 expert (with c>1, this fallback is rare). The dispatch mask $M \in \{0,1\}^{B \times E}$ records assignments. Let $p_{b,:} = \text{softmax}(\tilde{\ell}_b/\tau)$; the combine weights retain only the probabilities of assigned experts, renormalized: $C_{b,i} = M_{b,i}\, p_{b,i} / \sum_j M_{b,j}\, p_{b,j}$, so that $o_b = \sum_i C_{b,i}\, \text{Expert}_i(h_b)$.

Development note. During development, we also explored an additive utility bias $\lambda_u u_i$ added to the routing scores of Eq. 1, where u_i is an EMA of expert gradient magnitudes intended to reward actively learning

Algorithm 1 Sparse MoE training with hard capacity (one iteration)

1: $(\tau, k) \leftarrow \textsc{Schedule}(\text{epoch})$
2: Sample mini-batch (x, y)
3: $h \leftarrow \textsc{Backbone}(x)$
4: $\tilde{\ell}_i \leftarrow (W_r h)_i + w_k \cos(Kh,\, e_i)$ for $i = 1, \ldots, E$ {Routing scores}
5: $(M, C) \leftarrow \textsc{CapacityAwareTopK}(\text{softmax}(\tilde{\ell}/\tau), k, c)$ {Mask, combine weights}
6: $o \leftarrow \textsc{DispatchAndCombine}(h, \{\text{Exp}_i\}, M, C)$
7: $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}(o, y) + \lambda_{\text{lb}}\mathcal{L}_{\text{lb}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}$
8: Backpropagate and update all parameters

experts. Code inspection revealed that this bias was operationally negligible in our CIFAR experiments: the additive term was too small relative to learned router margins to change any top-k selections. We retain the utility machinery in the released code for reproducibility but do not claim it as a contribution.

3.3 Training Objective

The total loss combines cross-entropy with two regularizers:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{lb}}\mathcal{L}_{\text{lb}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}, \tag{4}$$

where $\mathcal{L}_{\text{lb}}$ is the Switch Transformer load-balance loss (Fedus et al., 2022):

$$\mathcal{L}_{\text{lb}} = E \cdot \sum_{i=1}^{E} f_i \cdot p_i, \tag{5}$$

with f_i the fraction of samples dispatched to expert i and p_i the mean routing probability for expert i. The entropy regularizer $\mathcal{L}_{\text{ent}} = -H(\bar{p})$, where $\bar{p}$ is the batch-averaged routing probability vector, penalizes concentrated routing distributions.

Algorithm 1 summarizes one training iteration.

3.4 Evolutionary Search and Baselines

Four hyperparameters govern the accuracy–efficiency trade-off: width scale w, capacity factor c, load-balance weight λ_{lb}, and entropy weight λ_{ent} (search ranges and optimized values in Appendix D). We designate CIFAR-10 as the *development dataset* for hyperparameter search; consequently, CIFAR-10 results may be optimistic and our strongest generalization evidence rests on the three transfer datasets (CIFAR-100, Tiny-ImageNet, ImageNet-1K). The three routing hyperparameters (c, λ_{lb}, λ_{ent}) are held fixed across datasets. Dataset-specific training parameters (warmup epochs, $\tau_{\min}$, expert hidden dimension h) are adapted per dataset (Section 4.1). We use population size 6 with top-2 elites; each candidate trains for 50 epochs with fitness $F = \text{Acc}_{\text{test}} - \lambda_{\text{red}} \max(0, r^* - r) - \lambda_{\text{gap}} \max(0, |g| - g^*)$, where r is the observed FLOPs reduction, g the MoE–dense gap, r^*=0.2, g^*=0.02, λ_{red}=5, λ_{gap}=2. The search converged in fourteen evaluations ($\sim$12 GPU-hours); the narrow ranges constrain the optimizable volume, and the three routing hyperparameters transfer to all three evaluation datasets without retuning.

For cross-method comparison, we reimplement Soft MoE (Puigcerver et al., 2024) and Expert Choice (Zhou et al., 2022), adapting each to our per-sample CNN setting. All MoE variants share the same expert architecture (8 experts, 2-hidden-layer MLP) and base training configuration to ensure fair comparison.

3.5 Per-Sample Soft Gating

Soft MoE (Puigcerver et al., 2024) dispatches via softmax over the batch dimension, which averages unrelated images in per-sample CNN classification. To isolate this as the collapse cause, we implement *per-sample soft*

gating: each expert has a slot embedding $s_i \in \mathbb{R}^d$, and gating weights are:

$$w_i = \frac{\exp(h^\top s_i/\tau)}{\sum_{j=1}^{E} \exp(h^\top s_j/\tau)}, \quad i = 1, \ldots, E, \tag{6}$$

where the softmax is over experts (dim=E), not over the batch. The combined output is:

$$o = \sum_{i=1}^{E} w_i \cdot \text{Expert}_i(h). \tag{7}$$

This preserves per-sample discriminative information while retaining Soft MoE's differentiable, token-drop-free dispatch. No auxiliary load-balancing loss is needed since all experts process every sample.

4 Experiments

We validate the sparse MoE architecture on four vision benchmarks of increasing complexity, first using CIFAR-10 as a development set for architecture search, then transferring the optimized routing configuration to three progressively harder evaluation datasets.

4.1 Experimental Setup

We evaluate on four datasets of increasing complexity: CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) (32×32, 50k/10k train/test), Tiny-ImageNet (200 classes, 64×64, 100k train), and ImageNet-1K (Deng et al., 2009) (1000 classes, 224×224, 1.28M/50k train/val). CIFAR models train for 50 epochs, Tiny-ImageNet for 80, and ImageNet for 30, using Adam (Kingma & Ba, 2015) (lr = 10^{-3}, CIFAR/Tiny-IN) or SGD (lr = 10^{-2}, momentum 0.9, cosine schedule, ImageNet); pretrained ImageNet backbones are fine-tuned at 0.1× head lr with AMP BF16 and batch 256. We use standard augmentations (flips, random crops) rather than modern recipes (AdamW, CutMix) so that the dense and MoE models share identical training and the routing effect is isolated (see Limitations); FLOPs are computed analytically. The MoE temperature anneals from τ=1.0 to $\tau_{\min}$ (sigmoid with κ=7.0 for CIFAR w=2.0, linear otherwise; $\tau_{\min}$=0.13 on CIFAR and 0.3 on Tiny-IN/ImageNet), with a 5-epoch warmup (k=8→1) on CIFAR and k=1 throughout for the other datasets.

We compare four routing methods at matched expert capacity (8 experts, 2-hidden-layer MLP): dense (FC head), sparse MoE k=1 with hard capacity constraints (ours), Soft MoE (Puigcerver et al., 2024), and Expert Choice (Zhou et al., 2022). Headline results report mean±s.d. over 10 seeds on CIFAR and 5 seeds on Tiny-ImageNet/ImageNet with paired t-tests, Cohen's d, and 95% CIs; CIFAR reports final-epoch test accuracy while Tiny-ImageNet and ImageNet report peak validation accuracy (more reliable under higher training variance).

CIFAR-10 serves as the hyperparameter development set, with evolutionary search fitness evaluated on its *test split*; CIFAR-10 results are therefore development evidence rather than generalization evidence, and our strongest generalization claims rest on the three transfer datasets where no test-set information informed the search. Cross-method tables adopt a two-tier design—a *standard backbone* section isolating routing effects on identical architectures and a *depthwise* section reporting the full co-optimization recipe; Soft MoE and Expert Choice are evaluated only on standard backbones, so cross-method numbers speak to routing-method quality on matched architectures rather than the full recipe. The controlled ρ-sweep in Table 1 shows that neither backbone family alone nor sparse routing at low ρ explains the gains; single-seed rows are exploratory and all claims rest on multi-seed results. Appendix F contextualizes absolute accuracy against published baselines.

4.2 CIFAR-10: Architecture Search

We employ a lightweight CNN backbone to isolate sparse MoE routing effects, focusing on *relative improvement* over matched dense baselines. Development sweeps over backbone width, expert hidden dimension, temperature schedule, and routing variants (full table in Appendix I) yield two headline depthwise-MoE configurations: an efficiency-oriented variant (w=0.72) achieving $+1.28 \pm 1.26$% over dense at a 22.7% FLOPs

Config	ρ	Dense	MoE	Gap	sd	t
Std. h=128	5.5%	84.34	82.16	−2.18	0.47	−10.34
DW h=128	**18.0%**	**76.75**	**75.48**	**−1.27**	**2.27**	**−1.25**
Std. h=512	19.7%	83.51	83.24	−0.27	0.62	−0.99
DW h=512	48.8%	76.25	78.16	+1.90	2.03	+2.10
Std. h=2048	**54.7%**	**83.04**	**84.11**	**+1.07**	**1.10**	**+2.17**
DW h=2048	83.3%	75.90	80.69	+4.79	1.79	+6.00

Table 1: CIFAR-10 ρ-sweep (5-seed mean±s.d.). Bold rows are the two predicted reversals.

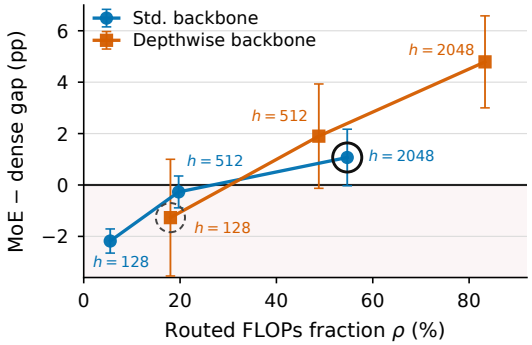

Figure 2: Visualizing Table 1: rings mark predicted reversals (solid = confirmed at t=+2.17; dashed = directional, t=−1.25, n.s.).

reduction (p=.011, d=1.01), and a higher-accuracy variant (w=2.0, three conv blocks with BatchNorm) reaching 88.41% test accuracy with $+0.54 \pm 0.16$% over dense ($p<10^{-5}$, d=3.31, all 10 seeds positive). The two sit at different accuracy–efficiency operating points on the same recipe, both with significant positive gaps.

Cross-Method Comparison. On CIFAR-10, both Soft MoE and Expert Choice underperform the dense baseline on the standard backbone; only our depthwise co-optimized configuration achieves a positive gap. Cross-method numbers across datasets are consolidated in Table 6 (Section 5).

4.3 Controlled ρ-sweep on CIFAR-10

To test whether ρ (the routed share of total FLOPs), rather than backbone family, is the active variable behind the compute-leverage observation, we sweep expert hidden size $h \in \{128, 512, 2048\}$ across both standard and depthwise backbones (six configurations × five seeds = 30 runs). The dense head is feature_dim → 1024 → h → 10 and each expert is feature_dim → h → h → 10, sharing h to remove the fixed-256 mismatch in earlier experiments. Two predictions follow: a high-ρ *standard* backbone should turn positive, and a low-ρ *depthwise* backbone should turn negative.

Table 1 and Fig. 2 report the result. The sign boundary tracks ρ: configurations at or below 19.7% routed FLOPs are negative, while the three higher-ρ configurations are positive. Both predicted point-estimate reversals occur: `Std.` h=2048 flips positive (t=+2.17, statistically supported) and `DW` h=128 flips negative (t=−1.25, directional only). This refutes both "depthwise is required" and "depthwise alone is sufficient," and is the cleanest evidence we have that ρ, not backbone family, is the active variable.

We note one caveat: the sweep varies head capacity and ρ jointly, since the expert hidden size determines both the routed FLOP fraction and the per-expert capacity. The *direction* of effect cleanly tracks ρ; the *magnitude* mixes capacity. A strict ρ-isolation experiment with FLOP-matched dense-head controls is left to future work.

4.4 Multi-Dataset Scaling

Having established routing effectiveness on the development benchmark, we transfer the four GA-optimized routing hyperparameters to three progressively more complex datasets to test generalization.

On CIFAR-100 (Table 2, top half) the MoE–dense gap widens: the GA-optimized routing hyperparameters transfer without retuning, producing a positive zero-shot gap and ten positive seeds at p=0.0013 in the multi-seed setting; the wider DW+MoE model exceeds the standard dense baseline in absolute accuracy, demonstrating that the depthwise efficiency penalty need not compromise performance. Token-mixing Soft MoE collapses catastrophically on this benchmark; per-sample dispatch (Section 3.5) rescues it above dense, and Expert Choice trails dense—see Section 5 and Table 6.

Configuration	Seeds	Dense	MoE	Gap	p	d	ΔFLOPs
CIFAR-100 (final-epoch test accuracy):							
Std backbone (24/48)	1	57.93	53.90	−4.03	—	—	−8.8%
Opt. DW Slim (ZS)†	1	43.72	44.02	+0.30	—	—	−22.7%
Opt. DW Wide	1	44.11	46.98	+2.87	—	—	−18.7%
Opt. DW Slim	10	41.38±2.10	44.37±1.61	**+2.99 ± 2.07**	.0013	1.45	−22.7%
3-blk+BN DW w=2.0	10	59.85±0.56	61.29±0.88	+1.44 ± 1.06	.0019	1.37	−4.3%
Tiny-ImageNet (peak validation accuracy):							
Std (h=1024)	1	43.73	46.33	+2.60	—	—	+5.5%
DW w=0.72 (h=1024)	5	39.23±.46	42.61±.30	+3.38 ± .48	$<10^{-4}$	7.00	+18.2%
DW w=1.2 (h=1024)	5	42.36±.20	**46.35±.54**	**+3.99 ± .42**	$<10^{-4}$	9.52	+12.0%
DW w=2.0 (h=1024)	5	44.05±.40	45.86±1.77	+1.81 ± 1.89	.10	0.95	+7.8%

Table 2: CIFAR-100 and Tiny-ImageNet results (%). Multi-seed rows give mean±s.d. with paired t-test p and Cohen's d. †Zero-shot: CIFAR-10 optimized config without retuning.

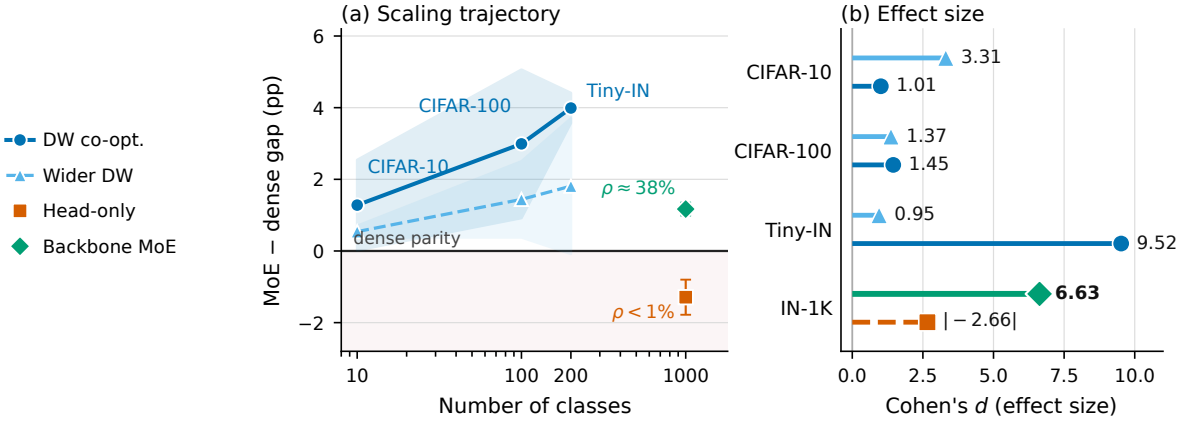

Figure 3: Task-complexity scaling. (a) MoE–dense gap vs. number of classes (±one s.d. bands); IN-1K head-only at ρ<1% is negative while backbone MoE at ρ≈38% recovers (Section 4.6). (b) Cohen's d by dataset; backbone MoE on IN-1K (d=6.63, bold) is the headline result.

Extending to 200 classes at 64×64 on Tiny-ImageNet (Table 2, bottom half), the MoE–dense gap increases further: the best configuration (w=1.2) achieves the largest gap across all datasets ($p<10^{-4}$, d=9.52) with remarkably low seed-to-seed variance, and DW+MoE *exceeds* the standard dense baseline in absolute accuracy at roughly 61% of standard-backbone FLOPs (positive ΔFLOPs values reflect that large experts exceed the dense head cost; Section 5.2). The standard-backbone cross-method picture mirrors CIFAR-100—both Expert Choice and our own sparse MoE show small negative gaps—underscoring that the positive multi-seed results arise from the joint DW co-optimization recipe rather than from routing alone.

The multi-seed gaps in Tables 2 and 3 are summarized visually in Fig. 3. Under depthwise co-optimization, the gap trends upward with class count across the three smaller-scale datasets, though these comparisons span different backbone widths and evaluation protocols (final-epoch test accuracy for CIFAR vs. peak validation for Tiny-ImageNet and ImageNet); the controlled w=2.0 series shows the same monotonic pattern on matched architectures, with Tiny-ImageNet falling short of significance (p=0.10). ImageNet-1K appears as two distinct configurations: head-only MoE yields negative gaps on both backbones, while backbone MoE recovers a positive gain (Section 4.6); the joint roles of ρ and routing regime in this recovery are disentangled by the k-ablation in that section. A utility-bias ablation (Appendix J) confirms that the additive utility term explored during development does not explain the observed gains.

Table 3: ImageNet-1K results (peak validation accuracy, 5-seed mean±s.d.). ⋆Per-sample soft gating. †Token-mixing Soft MoE (3-seed).

Backbone	Method	Top-1	Gap	h	ρ (rho)
ResNet-18	Dense (FC)	70.29 ± 0.04	—	—	0.06%
	MoE k=1	68.99 ± 0.51	-1.29	512	
	Soft (token)†	41.03 ± 0.24	-29.25	512	
	Soft (per-samp.)⋆	$\mathbf{69.67 \pm 0.08}$	$\mathbf{-0.66}$	512	
	Expert Choice	41.25 ± 1.72	-29.03	512	
MobileNet-V2	Dense (FC)	72.26 ± 0.06	—	—	0.85%
	MoE k=1 (h=512)	70.35 ± 0.45	-1.90	512	
	MoE k=1 (h=256)	68.07 ± 0.55	-4.19	256	
ResNet-18 (backbone)	Dense (layer3/4)	70.04 ± 0.14	—	—	≈38%
	MoE k=2‡	$\mathbf{71.21 \pm 0.17}$	$\mathbf{+1.17}$	—	

‡MoEConv2d in layer3/4 (E=8); details in Section 4.6.

4.5 ImageNet-1K Validation

To test whether MoE advantages extend to large-scale classification, we evaluate on ImageNet-1K using two pretrained backbones: ResNet-18 (ρ=0.06%) and MobileNet-V2 (ρ=0.85%). Both have $\rho < 1\%$, so these experiments test the compute-leverage hypothesis.

On ResNet-18, our sparse MoE trails the dense baseline significantly (p=0.004), consistent with negative margins on standard backbones at smaller scales. MobileNet-V2, which uses depthwise separable convolutions throughout and raises ρ by 14× over ResNet-18, would yield positive gaps if backbone type were the determining factor; instead, the gap remains significantly negative at both expert sizes (p<0.001, Table 3), and the worsening at smaller expert size confirms that undersized experts amplify the deficit (cf. Section 5.1). Depthwise convolutions alone are therefore insufficient when the head constitutes <1% of total FLOPs—further evidence that ρ, not backbone family, is the active variable.

Token-mixing Soft MoE and Expert Choice both exhibit catastrophic collapse on ImageNet despite a strong dense baseline; per-sample dispatch (Section 3.5) eliminates the collapse but only approaches dense parity rather than surpassing it. Section 5 analyzes the underlying mechanisms across all four datasets.

4.6 MoE-in-Backbone Validation

The head-only experiments above confirm that routing gains vanish in our ImageNet head-only settings when $\rho < 1\%$. We now test the converse by placing MoE directly within backbone convolutions where the majority of computation resides.

We replace ResNet-18's layer3 and layer4 3×3 convolutions with MoEConv2d layers (E=8 expert filter banks, k=2 routing); the router applies global average pooling followed by a linear layer to produce per-sample expert logits. Early layers (conv1, layer1, layer2) remain frozen pretrained, and MoE experts are initialized from pretrained layer3/4 weights with 10% noise perturbation to break symmetry. This places approximately 38% of total inference FLOPs under routing control, well above the $\rho < 1\%$ regime where head-only routing fails. The backbone-MoE variant achieves a statistically significant positive gap over the matched dense baseline (p=1.2×10^{-4}, all five seeds positive; full per-seed accuracies in Appendix H, reported alongside the k=1 ablation in Table 4), with balanced yet non-uniform expert utilization across all eight MoE layers—in contrast to the k=1 head-only configuration, which converged to exactly uniform routing with no specialization.

The contrast with head-only MoE on the *same backbone*—a significant negative gap at ρ=0.06% versus a significant positive gap once MoE moves into the backbone—is consistent with ρ being an important factor, but the two configurations also differ in routing regime (k=2 vs. k=1), trainable scope, and initialization. We isolate the routing-regime effect by repeating backbone MoE with k=1, holding architecture, routed fraction, pretrained initialization, and all other hyperparameters fixed: k=1 flips the gap negative across

Table 4: Routing-regime ablation on backbone MoE (ImageNet-1K, 5-seed mean±s.d.; same backbone, same routed fraction, same init). k alone reverses the gap by 3.25%.

Routing	Dense	MoE	Gap	t
k=2 (paper)	70.04 ± 0.14	71.21 ± 0.17	$\mathbf{+1.17 \pm 0.18}$	+14.82
k=1 (ablation)	70.03 ± 0.05	67.95 ± 0.12	$\mathbf{-2.08 \pm 0.15}$	−30.76

all five seeds (Table 4), a 3.25% reversal at identical ρ that demonstrates multi-expert routing ($k \geq 2$) is necessary in addition to a substantial ρ.

5 Mechanistic Analysis

The experiments in Section 4 establish that sparse MoE routing with depthwise backbones achieves consistent gains when the routed FLOP fraction is sufficiently high, and that depthwise backbones help primarily by raising this fraction (Tables 1 and 5). We now analyze the mechanisms underlying these gains and identify practical design guidelines.

5.1 Routing Stability and Specialization

Temperature scheduling significantly affects routing stability. On the CIFAR-10 development set, aggressive linear annealing to low $\tau_{\min}$ causes late-stage routing collapse where accuracy drops well below the training peak; sigmoid schedules avoid this collapse, and raising $\tau_{\min}$ does the same with linear schedules (full ablation in Appendix C, Fig. 10). Prior work addresses routing collapse through auxiliary losses (Fedus et al., 2022; Zoph et al., 2022); we find that temperature scheduling is a complementary and important stability lever. In development runs, soft auxiliary losses alone did not reliably prevent concentrated dispatch, so we treat hard capacity (Eq. 3) as a design constraint that enforces per-batch expert usage rather than as an isolated contribution. Expert hidden dimension further affects the capacity–diversity trade-off: moderate hidden sizes maintain full diversity, while smaller experts collapse at k=1 on harder tasks, suggesting the minimum viable expert dimension scales with output complexity.

Specialization emerges along output structure. The CIFAR-100 routing heatmap (Fig. 4, Appendix A) shows broad superclass-aligned patterns across multiple active experts; the CIFAR-10 heatmap produces sharper class specialization consistent with the lower output complexity. On the narrower backbone (w=0.72), test-time routing concentrates on 1–3 experts despite training-time usage showing 8/8 active, yet accuracy gaps *increase* with this concentration—plausibly a training-time diversity regularization effect, in which hard capacity constraints force all experts to develop competence during training and build latent capacity that benefits the dominant test-time experts. Training dynamics across the three smaller-scale datasets and a t-SNE of expert assignments versus class labels appear in Appendices B and A; we treat the alignment as qualitative rather than a quantitative claim of semantic specialization.

5.2 Efficiency Analysis

Table 5 decomposes FLOPs into backbone and head for efficiency-oriented expert sizes. On CIFAR, the head accounts for nearly half of total FLOPs, enabling substantial MoE reduction. On Tiny-ImageNet, convolutions dominate; headline configurations use larger experts for accuracy rather than efficiency. On ImageNet, the head is negligible. This decomposition explains the multi-dataset gap pattern: depthwise backbones shift ρ from negligible to near-parity, which is where positive accuracy gaps appear.

When the head constitutes fraction ρ of total FLOPs, a sparse router that reduces head computation by factor s saves only $\rho \cdot s$ overall, so for very small ρ even perfect routing yields negligible savings. Our experiments reveal a graded *compute-leverage relationship* across six ρ values (Table 5): at high ρ (CIFAR), routing yields both accuracy gains and meaningful FLOPs savings; at moderate ρ (Tiny-ImageNet), accuracy gains but negligible FLOPs reduction; at $\rho \leq 1\%$ (ImageNet head-only), routing consistently hurts. MobileNet-V2 is complementary evidence: depthwise convolutions throughout, yet still negative gaps—ρ is more predictive

Table 5: FLOPs decomposition (ρ = routed share of total FLOPs). Efficiency-oriented h: 304 CIFAR, 440 Tiny-IN.

Dataset	Conv	Head	ρ	Head Sav.	Total Red.
CIFAR-10	51.3%	48.7%	48.7%	43.0%	20.9%
CIFAR-100	51.1%	48.9%	48.9%	42.4%	20.8%
Tiny-ImageNet	81.9%	18.1%	18.1%	17.9%	3.2%
IN-1K (ResNet-18)	99.9%	0.1%	0.06%	—	∼0%
IN-1K (MobNetV2)	99.2%	0.8%	0.85%	—	∼0%
IN-1K (Backbone)†	62%	38%	38%	—	—

†MoE in layer3/4; "Head" denotes MoE-routed FLOPs.

than backbone type. This is a CNN analogue of the capacity–compute trade-offs formalized by Clark et al. (2022) for language MoE. Backbone MoE (Section 4.6) recovers a positive gain on the same backbone, though confounds with routing regime preclude a purely causal interpretation (see Limitations). See Appendix F for comparison against published efficient architectures.

At low ρ, the router learns to concentrate test-time routing on one expert (ResNet-18 on ImageNet); this collapse is adaptive—on MobileNet-V2, lower-entropy seeds achieve *less* negative gaps—suggesting that distributing samples at low ρ adds routing overhead without meaningful compute reorganization.

We emphasize that the FLOPs analysis is *analytical*: it quantifies the structural opportunity for routing, not inference speed. On wall-clock latency (Table 11, Appendix G) our sparse MoE is the slowest method despite activating only one of E=8 experts—an artifact of *how operations are scheduled, not how many are performed*: our Python-loop implementation incurs $\sim$10 μs host-side dispatch overhead per expert. The slowdown ratio *collapses with model scale* (145× over dense on CIFAR-100 down to 4.1× on ImageNet-1K) within the measured head-only configurations, suggesting that the unfused penalty would matter less in large-scale, high-ρ settings, though we did not measure latency for the backbone-MoE configuration. Fused MoE kernels (Gale et al., 2023; Hwang et al., 2023) close this dispatch-overhead gap.

5.3 Cross-Method Synthesis

Table 6 consolidates the cross-method comparison across all four datasets. Three patterns emerge: (1) token-mixing Soft MoE collapses across all datasets, with severe degradation on the harder benchmarks; (2) per-sample soft gating (Section 3.5) eliminates this collapse on every dataset, recovering to near-dense levels but never consistently surpassing them; and (3) sparse MoE with depthwise backbones achieves positive gaps when ρ is sufficiently high.

Diagnosing the Soft MoE failure. A per-sample dispatch variant (softmax over experts instead of the batch; Appendix E) rescues CIFAR-100 ($-36\% \rightarrow +1.69\%$), establishing batch-axis dispatch as the dominant failure mode in our per-sample CNN setting. On Tiny-ImageNet (-1.13%) and ImageNet (-0.66%) per-sample dispatch is no longer catastrophic but does not reach parity, indicating a residual scale-dependent factor.

Only our method shows an upward trend across the three smaller-scale datasets; head-only ImageNet confirms the negative-ρ boundary, while backbone MoE recovers a positive gap on the same dataset. On standard backbones, all methods show flat or declining gaps—consistent with the contribution being the joint recipe rather than routing alone.

6 Conclusion

We presented an empirical study of when sparse MoE routing helps in vision classification, identifying a *compute-leverage pattern*: positive accuracy gaps require a substantial fraction ρ of total FLOPs to be routed. A controlled hidden-size sweep on CIFAR-10 (Section 4.3, Table 1) yields a sign boundary that tracks ρ across both backbone families and both predicted point-estimate reversals (high-ρ standard statistically supported; low-ρ depthwise directional), ruling out backbone family alone as the explanation. At ImageNet scale, this

Table 6: Cross-method accuracy gaps (%) across all datasets. Best non-dense result **bolded**. ⋆Per-sample soft gating (5-seed mean for C-100/Tiny-IN/IN-1K).

Method	C-10	C-100	Tiny-IN	IN-1K	Trend
Soft MoE (token-mix)	−13.3	−36.4	−42.9	−29.3	↓
Soft MoE (per-sample)⋆	—	+1.7	−1.1	−0.7	≈
Expert Choice	−1.2	−6.8	−4.2	−29.0	↓
Sparse (std.)	−1.4	−4.0	−0.3	−1.3	≈
Ours (DW w=0.72)	**+1.3**	**+3.0**	+3.4	—	↑
Ours (wider DW)	+0.5	+1.4	**+4.0**	—	↑
ImageNet-1K (pretrained, head-only, $\rho < 1\%$):					
Ours (ResNet-18)	—	—	—	−1.3	—
Ours (MobNetV2)	—	—	—	−1.9	—
ImageNet-1K (backbone MoE, $\rho \approx 38\%$):					
Ours (layer3/4)	—	—	—	**+1.2**	↑

fraction-condition becomes necessary but not sufficient: a complementary k-ablation (Section 4.6) holds backbone, routed fraction, and pretrained initialization fixed and varies only the top-k selection, reversing the gap from +1.17% to −2.08% across all five seeds. The MobileNet-V2 control reinforces this picture: depthwise convolutions alone are insufficient when the routed fraction is small. Across our experiments, then, a substantial ρ is necessary at every scale, and at ImageNet scale multi-expert routing ($k \geq 2$) is additionally required; neither alone suffices.

Within the favorable ρ regime, gaps trend upward with task complexity and wider depthwise backbones reach competitive absolute accuracy; token-mixing Soft MoE and Expert Choice underperform on standard backbones.

Limitations. The additive utility bias explored during development was operationally negligible (p=0.87 on CIFAR-100; Appendix J) and is not claimed as a contribution. The evolutionary search uses CIFAR-10's test split for fitness; a held-out validation split would improve protocol cleanliness. Training recipes use standard optimizers and augmentations rather than modern techniques (AdamW, CutMix), so absolute accuracy lags stronger recipes though relative comparisons remain valid. Our backbone-MoE protocol uses pretrained initialization with k=2 while head-only uses random initialization with k=1; the k-ablation bridges these on one architecture, but a unified four-dataset protocol remains open. The CIFAR-10 ρ-sweep varies head capacity and ρ jointly; a strict FLOP-matched isolation is left to future work. Wall-clock latency in our unfused implementation is dominated by per-expert kernel launches, a gap closed by fused MoE kernels (Gale et al., 2023; Hwang et al., 2023).

Future Work. Deeper backbones at larger scale, fused dispatch kernels, knowledge distillation (Hinton et al., 2015) to dense students, and validation on fine-grained domains.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. doi: 10.48550/arXiv:1308.3432.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34600–34613, 2022.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017. doi: 10.1109/cvpr.2017.195.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Heber, Laurentiu Comanici, Agatha Lazaridou, et al. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pp. 4057–4086, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/cvpr.2009.5206848.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–40, 2022.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 5, 2023.

Xumeng Han, Longhui Wei, Zhiyang Dou, Zipeng Wang, et al. ViMoE: An empirical study of designing vision mixture-of-experts. *arXiv preprint arXiv:2410.15732*, 2024. doi: 10.48550/arXiv:2410.15732.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2022. doi: 10.1109/TPAMI.2021.3117837.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/cvpr.2016.90.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. doi: 10.48550/arXiv:1704.04861.

Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Olatunji Ruwase, et al. Tutel: Adaptive mixture-of-experts at scale. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 5, 2023.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. doi: 10.48550/arXiv:2401.04088.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. doi: 10.48550/arXiv:1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Lisha Li, Kevin Jamieson, Jeff DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185): 1–52, 2017. doi: 10.48550/arXiv:1603.06560.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *Transactions on Machine Learning Research*, 2024.

Xiaonan Nie, Xupeng Miao, Shijie Cao, Lingxiao Ma, Qibin Liu, Jilong Xue, Youshan Miao, Yi Liu, Zhi Yang, and Bin Cui. Dense-to-sparse gate for mixture-of-experts. *arXiv preprint arXiv:2112.14397*, 2022.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv:2308.00951.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. In *International Conference on Machine Learning*, 2024.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 4780–4789, 2019. doi: 10.1609/aaai.v33i01.33014780.

Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1973.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8583–8595, 2021.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. doi: 10.1109/cvpr.2018.00474.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. doi: 10.48550/arXiv:1701.06538.

Mathurin Videau, Alessandro Leite, Marc Schoenauer, and Olivier Teytaud. Mixture of experts in image classification: What's the sweet spot? *arXiv preprint arXiv:2411.18322*, 2024. doi: 10.48550/arXiv:2411.18322.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. SkipNet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pp. 409–424, 2018.

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

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022. doi: 10.48550/arXiv:2202.08906.

Appendix

A Additional Routing Visualizations

A.1 Routing Heatmaps

The CIFAR-100 routing heatmap (Fig. 4) shows broad superclass-aligned patterns across multiple active experts: columns sorted by 20 superclasses, with seven experts showing superclass-aligned specialization. The CIFAR-10 routing heatmap (Fig. 5) shows expert specialization across 10 classes using the 3-block+BN DW w=2.0 backbone. Five of eight experts show clear class specialization, with Expert 4 dominating automobiles at 0.96. The 10-class setting produces sharper specialization than the CIFAR-100 heatmap, where each expert covers more classes.

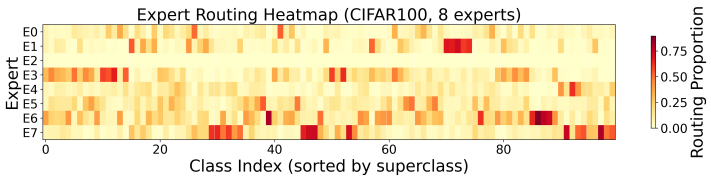

Figure 4: Expert routing heatmap, CIFAR-100 test set (3-block+BN DW w=2.0, seed 123). Columns sorted by 20 superclasses; seven experts show superclass-aligned specialization.

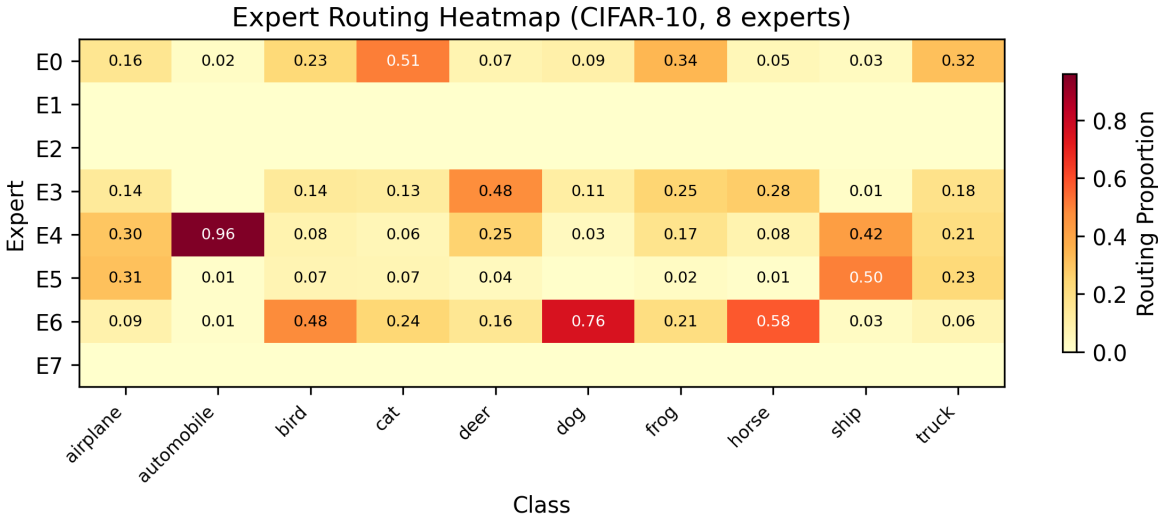

Figure 5: Expert routing heatmap, CIFAR-10 test set (3-block+BN DW w=2.0, seed 2024). Five of eight experts show clear class specialization.

A.2 t-SNE of CIFAR-100 Routing

Fig. 6 shows t-SNE embeddings of the CIFAR-100 test set colored by expert assignment (left) and by class label (right). The two panels show visually consistent structure, but the expert assignments are not independent of the class-cluster geometry and we treat the alignment as qualitative rather than as a quantitative claim of semantic specialization.

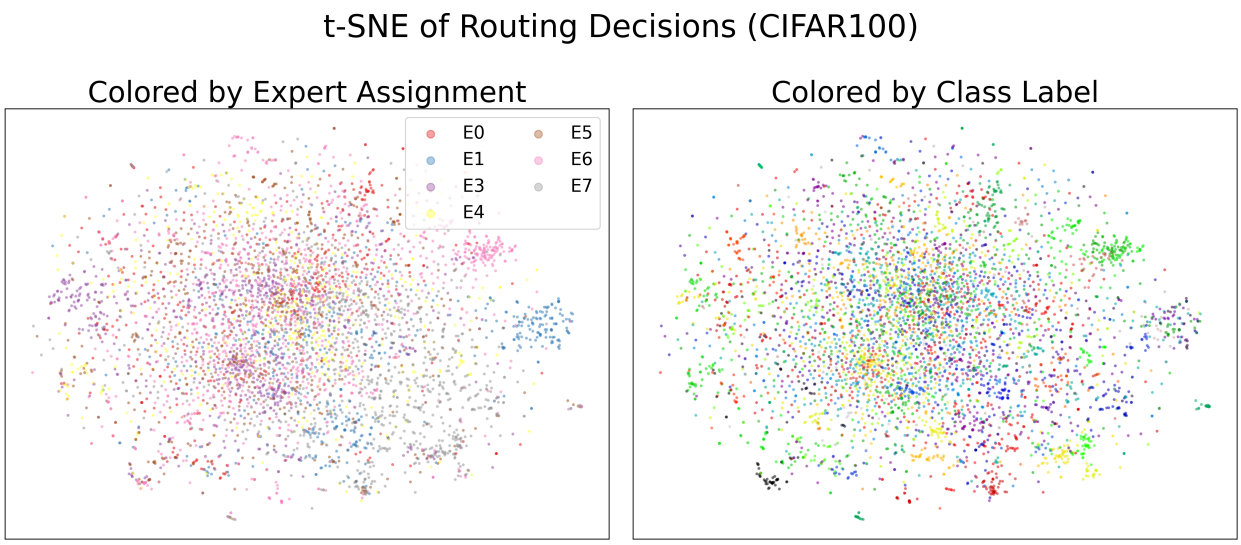

Figure 6: t-SNE of CIFAR-100 routing (3-block+BN DW w=2.0, seed 123). Left: by expert assignment. Right: by class label.

B Training Dynamics and Task-Complexity Scaling

Figures 7, 8, and 9 show training dynamics for the three smaller-scale datasets. Across all three, expert usage stabilizes early in training, MoE surpasses the dense baseline by mid-training, and the gap is maintained with comparable train–test generalization, suggesting improved representation rather than overfitting.

The per-dataset training-dynamics figures above complement the cross-dataset scaling summary in Fig. 3 (Section 4.4).

C Temperature Schedule Ablation Details

Table 7 presents temperature schedule ablations on the standard CIFAR-10 backbone. The configuration with linear annealing to $\tau_{\min}$=0.1 achieves +0.92% peak over the dense baseline but collapses in late stages (final accuracy 80.49%). Sigmoid schedules avoid collapse entirely (+0.03% gap, stable). Setting $\tau_{\min}$=0.3 (as used on Tiny-ImageNet and ImageNet) prevents collapse with linear schedules. These results motivated the per-dataset schedule choices described in Section 4.1. The configurations all include the additive utility bias, which is itself ablated separately in Section J.

Table 7: Temperature schedule ablations on the standard CIFAR-10 backbone. All rows include the additive utility bias; the utility ablation is in Section J.

Config	Dense	Peak	Final	ΔFLOPs
τ: 1.0→0.1 (linear)	83.34	**84.26**	80.49	−9.4%
τ: 1.0→0.5 (linear)	84.00	81.20	76.50	−36.2%
τ: 1.0→0.13 (sigmoid, κ=7)	83.38	83.41	83.41	−8.9%

Fig. 10 illustrates the same sensitivity as accuracy trajectories over training.

D Hyperparameter Search Details

Table 8 lists the four hyperparameters varied by the evolutionary search on CIFAR-10 (Section 3.4). Ranges are narrow by design: the search protocol is a lightweight tuning loop rather than a NAS contribution. The

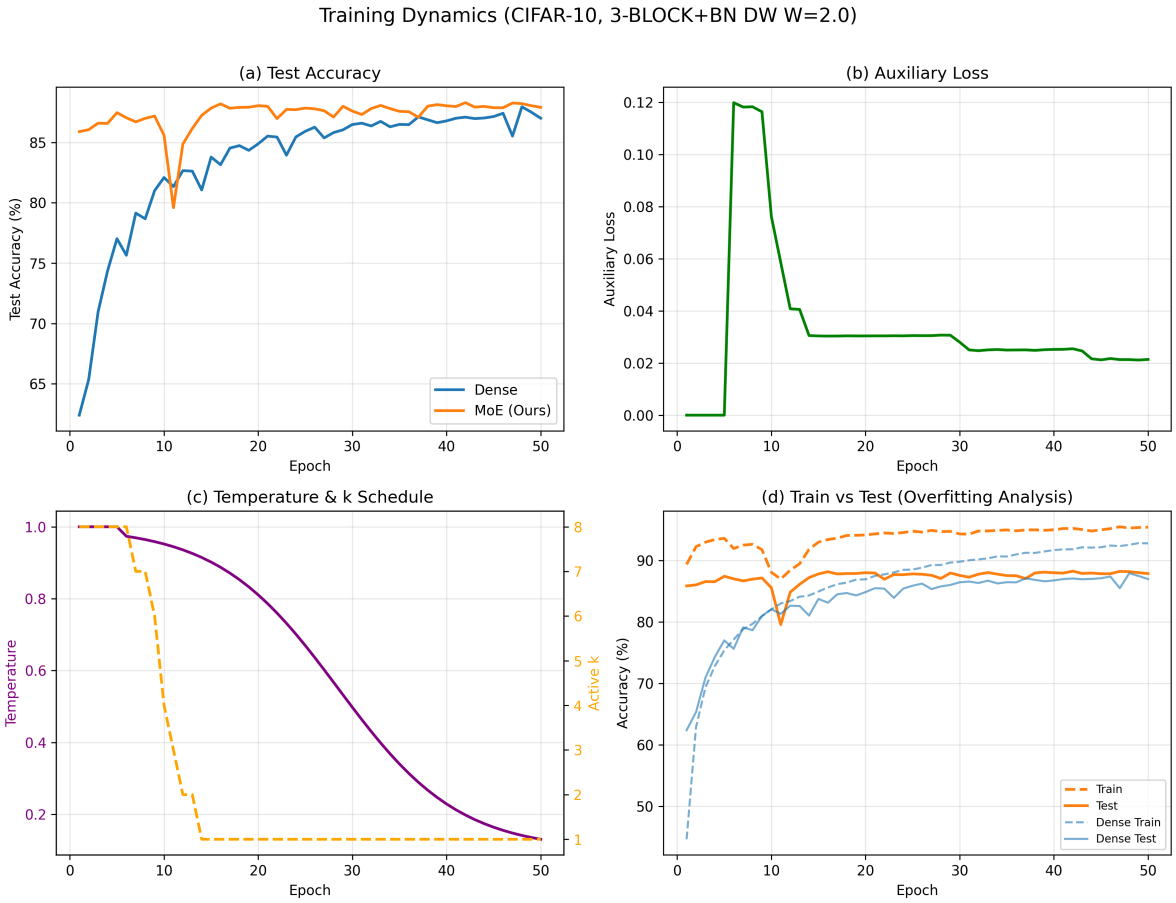

Figure 7: Training dynamics on CIFAR-10 (3-block+BN DW w=2.0, seed 2024). MoE surpasses dense after epoch 10.

three routing hyperparameters (c, λ_{lb}, λ_{ent}) transfer to all three evaluation datasets without retuning; only training details (h, $\tau_{\min}$, warmup epochs) are adjusted per dataset.

Table 8: Evolutionary search ranges and optimized values. A utility weight λ_u was also searched ($[0.08, 0.15]$, best 0.087) but found operationally negligible; see development note in Section 3.

Parameter	Range	Mut. σ	Best
Width scale w	$[0.65, 0.85]$	0.020	0.717
Capacity c	$[1.05, 1.20]$	0.015	1.064
Load-bal. λ_{lb}	$[0.01, 0.02]$	0.001	0.019
Entropy λ_{ent}	$[0.015, 0.030]$	0.002	0.024

E Per-Sample Soft Gating Derivation

Standard Soft MoE (Puigcerver et al., 2024) computes dispatch weights via:

$$D_{ij} = \frac{\exp(\phi_{ij})}{\sum_{i'=1}^{B} \exp(\phi_{i'j})}, \quad \phi_{ij} = x_i^\top s_j, \tag{8}$$

where x_i is the i-th input token and s_j the j-th slot embedding. The softmax is over the batch (token) dimension (i). Each slot j receives a weighted combination $\tilde{x}_j = \sum_i D_{ij} x_i$ of all input tokens.

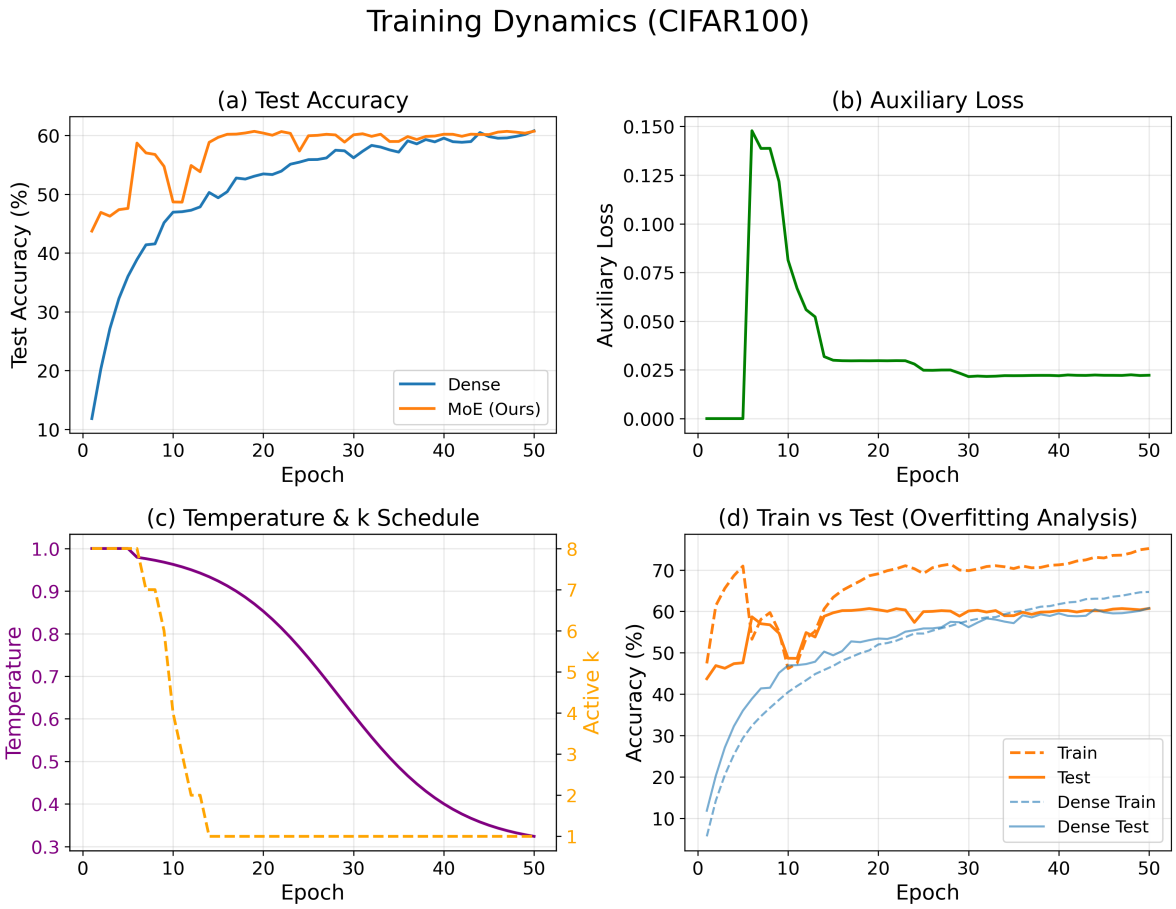

Figure 8: Training dynamics on CIFAR-100 (3-block+BN DW w=2.0, seed 123). (a) Accuracy gap. (b) Auxiliary loss. (c) Schedule. (d) Train/test accuracy.

In Vision Transformers, tokens are semantically related patches from the same image, so this averaging produces a meaningful aggregate. In our per-sample CNN classification setting, each "token" is an *independent image*. With batch size B and E experts (each with one slot), the dispatch weights become:

$$D_{ij} \approx \frac{1}{B} \quad \text{for large } B, \tag{9}$$

because the softmax over B unrelated inputs produces near-uniform weights. Each expert slot thus receives an approximately uniform average of unrelated images, destroying discriminative information.

Our *per-sample soft gating* (Eqs. 6–7) addresses this by applying the softmax over experts instead of tokens:

$$w_i^{(b)} = \frac{\exp(h_b^\top s_i/\tau)}{\sum_{j=1}^{E} \exp(h_b^\top s_j/\tau)}, \quad o_b = \sum_{i=1}^{E} w_i^{(b)} \cdot \text{Expert}_i(h_b). \tag{10}$$

This preserves per-sample discriminative information: each sample is independently routed through all experts with sample-specific combination weights, eliminating the cross-sample averaging. The resulting model is equivalent to a soft attention over expert outputs, weighted by input–slot compatibility.

Table 9 reports the 5-seed aggregate (referenced from Section 5). Per-seed numbers are in Appendix H.

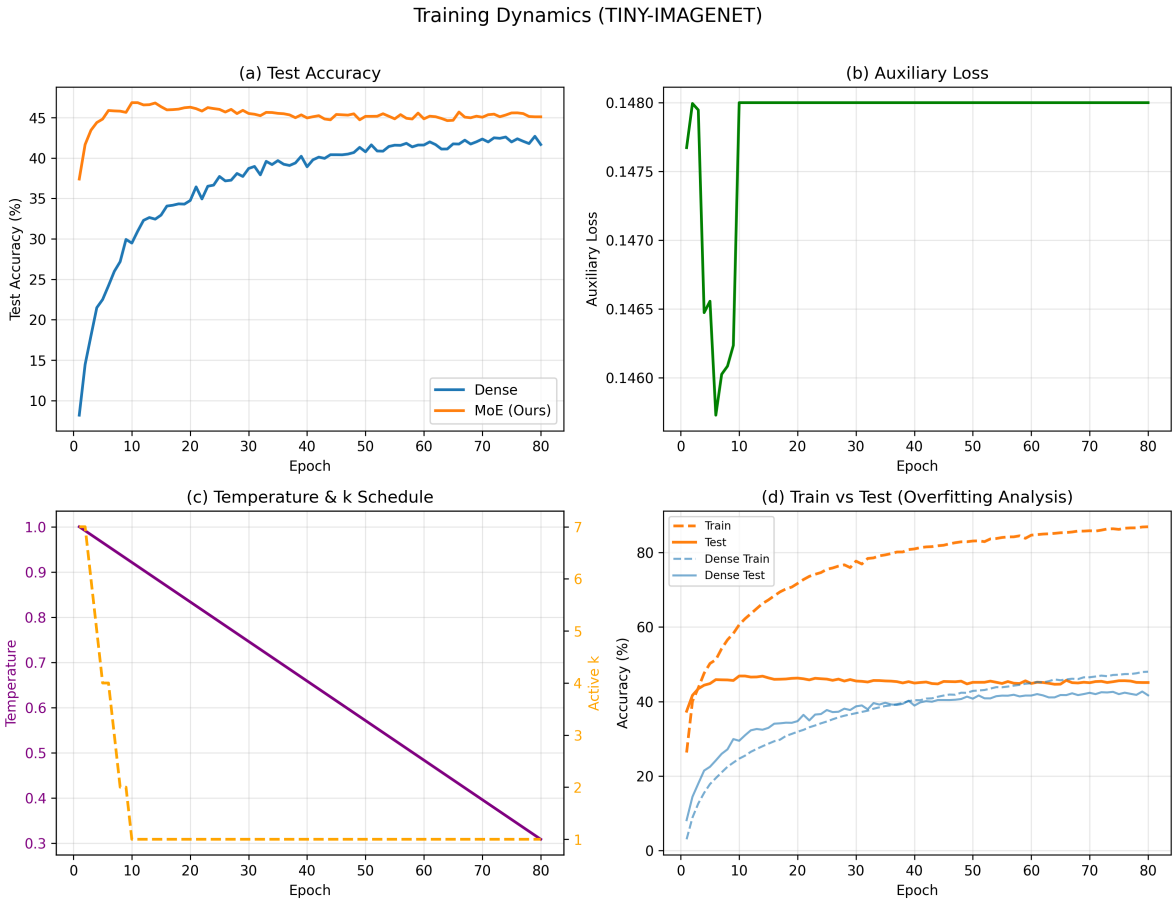

Figure 9: Training dynamics on Tiny-ImageNet (DW w=1.2, seed 42). MoE outperforms dense by +4.2% at its best epoch; routing entropy decreases smoothly from ∼2.0 to ∼1.5 as temperature anneals.

Table 9: Per-sample soft gating diagnostic (5-seed mean±s.d., peak validation accuracy). Batch-axis dispatch is the dominant Soft MoE failure mode but not the only one.

Dataset	Dense	Soft MoE	Gap	sd	t
CIFAR-100	57.53	59.22	**+1.69**	0.31	+12.02
Tiny-ImageNet	45.83	44.70	−1.13	0.97	−2.60
ImageNet-1K	70.33	69.67	−0.66	0.08	−19.45

F Published Efficiency Baselines

Table 10 contextualizes our best models against published baselines across all four datasets. Our study uses lightweight backbones to isolate MoE routing effects under controlled conditions; the comparison clarifies the accuracy–efficiency operating point relative to standard architectures trained with stronger recipes.

Two observations emerge. First, the absolute accuracy gap between our lightweight models and published baselines is substantial on CIFAR/Tiny-ImageNet (20–30 percentage points), reflecting the deliberately constrained backbone design and basic training recipe rather than a deficiency of the MoE mechanism. Second, on ImageNet where we use standard pretrained backbones, our dense baselines match published numbers (72.26% vs. torchvision's 72.15% for MobileNet-V2), confirming that the negative MoE gaps on ImageNet are attributable to the low ρ regime rather than implementation issues. The key contribution

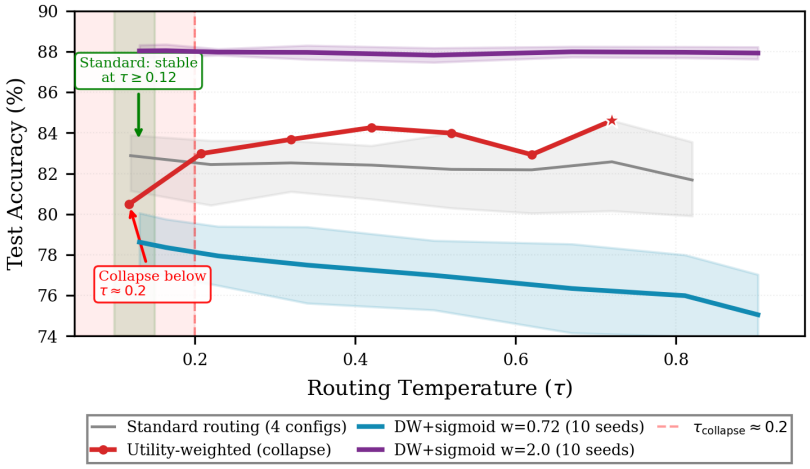

Figure 10: Temperature–accuracy trajectories (k=1, CIFAR-10). Standard routing (gray) is stable to low τ; routing with additive utility bias (red) collapses under aggressive annealing. Sigmoid schedules (blue/purple, 10 seeds) avoid collapse. The utility bias was operationally negligible (Section J); the collapse reflects temperature sensitivity, not the bias itself.

Table 10: Absolute accuracy of our best models vs. published baselines. Published numbers use standard training recipes (AdamW, CutMix, 200+ epochs); our models use basic recipes (Adam/SGD, 50–80 epochs) to control for training effects. †Approximate published numbers.

Dataset	Model	Acc (%)	FLOPs
CIFAR-10	ResNet-18 (standard recipe)†	∼95	556M
	MobileNet-V2†	∼94	300M
	Ours (DW w=2.0 + MoE)	88.41	∼24M
	Ours (DW w=0.72 + MoE)	78.62	∼3M
CIFAR-100	ResNet-18 (standard recipe)†	∼78	556M
	MobileNet-V2†	∼78	300M
	Ours (DW w=2.0 + MoE)	61.29	∼24M
Tiny-IN	ResNet-18 (standard recipe)†	∼66	1.8G
	MobileNet-V2†	∼64	300M
	Ours (DW w=1.2 + MoE)	46.35	∼90M
ImageNet	ResNet-18 (torchvision)	69.76	1.8G
	MobileNet-V2 (torchvision)	72.15	300M
	Our Dense (MobNetV2 fine-tuned)	72.26	300M

of our study is the *relative* MoE–dense improvement under matched conditions, which isolates the effect of routing from confounding training variables.

G Inference Latency Measurements

Table 11 reports inference latency and throughput for all four routing methods at batch size 128, measured on an RTX 5090. The analysis of these numbers—and why our unfused sparse MoE is the slowest method despite a lower FLOP count—is given in Section 5.2.

Table 11: Inference latency (ms) and throughput (kimg/s) at batch size 128. Measured on RTX 5090, CUDA sync (50 warmup + 200 timed passes).

	C-100		Tiny-IN		IN-1K	
Method	ms	k/s	ms	k/s	ms	k/s
Dense	0.25	519.3	1.01	126.2	11.3	11.3
MoE k=1	36.38	3.5	37.44	3.4	46.8	2.7
Soft MoE	0.63	204.2	1.31	97.6	11.4	11.2
Expert Choice	7.39	17.3	7.62	16.8	17.6	7.3

H Tier-C Strengthening Experiments: Per-Seed Results

This section reports per-seed accuracies for the three Tier-C experiments aggregated in Sections 4.3, 4.6, and 5: the controlled ρ-sweep on CIFAR-10, the per-sample soft gating diagnostic across three datasets, and the backbone-MoE k=1 ablation on ImageNet-1K. All 50 source JSONs are listed in `phase2_moe_backbone/TIER_C_FINDINGS.md` of the supplementary code package; aggregation is performed by `scripts/cluster/aggregate_summary.py`.

H.1 C1 — CIFAR-10 ρ-sweep (per-seed)

Each row reports final-epoch test accuracy for one (config, seed) pair. Six configurations × five seeds = 30 runs.

Table 12: Per-seed final-epoch test accuracy (%) for the CIFAR-10 ρ-sweep.

Config	Seed	Dense	MoE	Gap
Std. h=128	42	84.56	81.84	−2.72
Std. h=128	123	84.43	82.23	−2.20
Std. h=128	456	83.66	81.40	−2.26
Std. h=128	777	84.58	82.29	−2.29
Std. h=128	2025	84.48	83.06	−1.42
Std. h=512	42	83.77	82.80	−0.97
Std. h=512	123	83.80	83.56	−0.24
Std. h=512	456	83.38	83.12	−0.26
Std. h=512	777	83.42	82.82	−0.60
Std. h=512	2025	83.19	83.89	+0.70
Std. h=2048	42	82.90	85.21	+2.31
Std. h=2048	123	83.42	85.39	+1.97
Std. h=2048	456	83.26	82.85	−0.41
Std. h=2048	777	83.03	83.96	+0.93
Std. h=2048	2025	82.61	83.15	+0.54
DW h=128	42	78.01	78.43	+0.42
DW h=128	123	76.99	77.26	+0.27
DW h=128	456	75.54	75.95	+0.41
DW h=128	777	78.00	74.75	−3.25
DW h=128	2025	76.22	72.00	−4.22
DW h=512	42	76.06	78.41	+2.35
DW h=512	123	76.56	76.72	+0.16
DW h=512	456	75.83	79.94	+4.11
DW h=512	777	76.29	79.73	+3.44
DW h=512	2025	76.53	75.99	−0.54
DW h=2048	42	74.61	81.04	+6.43
DW h=2048	123	75.81	81.12	+5.31
DW h=2048	456	75.56	81.98	+6.42
DW h=2048	777	77.36	80.25	+2.89
DW h=2048	2025	76.15	79.06	+2.91

H.2 C2 — Per-sample soft gating (per-seed)

Five seeds per dataset; peak validation accuracy.

Table 13: Per-seed peak validation accuracy (%) for the per-sample soft gating diagnostic.

Dataset	Seed	Dense	Soft MoE	Gap
CIFAR-100	42	58.16	59.58	+1.42
CIFAR-100	123	57.99	59.35	+1.36
CIFAR-100	456	57.65	59.41	+1.76
CIFAR-100	777	56.75	58.89	+2.14
CIFAR-100	2025	57.08	58.85	+1.77
Tiny-ImageNet	42	45.46	43.55	−1.91
Tiny-ImageNet	123	46.40	44.72	−1.68
Tiny-ImageNet	456	45.64	45.92	+0.28
Tiny-ImageNet	777	45.96	44.15	−1.81
Tiny-ImageNet	2025	45.67	45.16	−0.51
ImageNet-1K	42	70.36	69.72	−0.64
ImageNet-1K	123	70.32	69.68	−0.64
ImageNet-1K	456	70.32	69.72	−0.60
ImageNet-1K	777	70.34	69.69	−0.65
ImageNet-1K	2025	70.33	69.53	−0.80

H.3 C3 — Backbone-MoE k=1 ablation (per-seed)

Five seeds, peak top-1 validation accuracy on ImageNet-1K. Same backbone, same $\rho \approx 38\%$, same pretrained initialization as the headline k=2 result; only the routing top-k changes.

Table 14: Per-seed peak top-1 (%) for the backbone-MoE k=1 ablation on ImageNet-1K.

Seed	Dense	MoE k=1	Gap
42	69.99	68.06	−1.93
123	70.10	67.96	−2.14
456	70.04	67.91	−2.13
777	70.03	67.76	−2.27
2025	69.97	68.04	−1.93
Mean±s.d.	70.03 ± 0.05	67.95 ± 0.12	$\mathbf{-2.08 \pm 0.15}$

H.4 Backbone-MoE k=2 headline (per-seed)

The 5-seed per-seed accuracies for the headline backbone-MoE result on ImageNet-1K (Section 4.6). Companion to the k=1 ablation in the previous subsection.

Table 15: Per-seed peak top-1 (%) for backbone-MoE k=2 on ImageNet-1K (ResNet-18 layer3/4 replaced with MoEConv2d, E=8, $\rho \approx 38\%$).

Seed	Dense	MoE k=2	Gap
42	70.17	71.29	+1.12
123	70.05	71.46	+1.41
456	69.95	71.17	+1.21
777	69.84	71.02	+1.18
2025	70.17	71.09	+0.92
Mean±s.d.	70.04 ± 0.14	71.21 ± 0.17	$\mathbf{+1.17 \pm 0.18}$

H.5 Reproducibility pointers

The supplementary code package contains, under `phase2_moe_backbone/`:

- `TIER_C_FINDINGS.md` — canonical record with full per-seed numbers and source-JSON paths.
- `results_c1_rho/`, `results_c2_persample/`, `results_c3_k1/` — 50 `results.json` files (30 + 15 + 5).
- `scripts/cluster/aggregate_summary.py` — aggregator with three schema parsers (`parse_c1_flat`, `parse_c2_nested`, `parse_c3_nested`).

I CIFAR-10 Development Configuration Sweep

The CIFAR-10 development sweep over backbone width, expert hidden dimension, temperature schedule, and routing variants (referenced in Section 4.2). Multi-seed rows report mean±s.d. (final-epoch test accuracy); single-seed rows are exploratory references. ΔFLOPs: negative = reduction, positive = increase vs. dense.

Table 16: CIFAR-10 configuration summary. *Utility bias was found operationally negligible (Section J).

Config	Dense	MoE	Gap	ΔFLOPs
Wide 32/64	85.11	83.89	−1.22	−5.2%
Slim (h=320)	83.39	83.28	−0.11	−8.3%
Slim (h=304)	84.07	83.20	−0.87	−8.9%
Ultra-slim 16/32	82.51	81.15	−1.36	−13.5%
Sigmoid sched.	83.38	83.41	+0.03	−8.9%
+ utility bias*	83.34	84.26	+0.92	−9.4%
DW prototype	78.03	77.40	−0.63	−22.0%
10-seed (Opt. DW, w=0.72):				
Mean±s.d.	77.35±1.34	78.62±1.51	$+1.28 \pm 1.26$	−22.7%
p=.011, d=1.01, CI [+0.38, +2.18]				
10-seed (3blk+BN DW, w=2.0, opt. cfg):				
Mean±s.d.	87.87±0.23	88.41±0.30	$+0.54 \pm 0.16$	−4.3%
$p<10^{-5}$, d=3.31, CI [+0.42, +0.65]				

J Utility Bias Ablation

The multi-dataset results in Section 4 were obtained with an additive utility bias in the routing logits (λ_u=0.087, the value selected by the evolutionary search). To test whether this mechanism contributes to the observed gains, we ran a matched CIFAR-100 ablation with utility disabled (λ_u=0), holding all other hyperparameters fixed.

Table 17: Utility ablation. CIFAR-10: single-seed development reference. CIFAR-100: 5-seed mean±s.d. (transfer). †Paired test of utility effect on CIFAR-100: p=0.87.

Dataset	Configuration	Dense	MoE	Gap
C-10	Std + utility	83.34	84.26	+0.92
	DW + no utility	78.03	77.40	−0.63
	DW + utility (opt.)	77.35	78.62	+1.28
C-100†	DW + no utility	42.54 ± 2.09	45.63 ± 2.72	$+3.09 \pm 1.63$
	DW + utility (opt.)	41.38 ± 2.10	44.37 ± 1.61	$+2.82 \pm 2.14$

On CIFAR-100 (transfer), disabling utility yields a gap statistically indistinguishable from the full recipe (paired p=0.87). Code inspection confirmed the cause: the additive utility bias was operationally negligible relative to learned router margins, not changing any top-k selections in the tested models. The observed

gains are therefore not attributable to the utility bias. The CIFAR-10 development rows are retained for completeness but should be interpreted in light of this finding.