ROUTING THE LOTTERY: ADAPTIVE SUBNETWORKS FOR HETEROGENEOUS DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

The Lottery Ticket Hypothesis (LTH) posits that large neural networks contain sparse subnetworks, or winning tickets, that can be trained in isolation to match the performance of their dense counterparts. The mask defining such a ticket is extracted from data but static at inference: a single subnetwork serves the entire data distribution, and input-conditional computation arises through activations rather than parameters. We ask whether the structure of the data distribution is also reflected at the parameter level: what family of tickets emerges when ticket extraction is conditioned on partitions of the data, and how is that family organized? To study this, we propose *Routing the Lottery* (RTL), an adaptive pruning framework that reveals such data-conditional subnetworks, which we term *adaptive tickets*. Rather than focusing on compression, we use pruning as a probe to study how computation is organized within a network under controlled data partitions. We analyze these subnetworks through mask overlap and structural decomposition calibrated against chance-level null models, uncovering systematic variation in parameter sharing across data-dependent pathways. Our results show that classical universal winning tickets can be interpreted as structured superpositions of adaptive pathways within a bounded sparsity regime. At extreme sparsity levels, this relationship largely breaks down as universal pruning converges to structurally degenerate solutions. Overall, RTL reframes pruning as an analytical tool for studying data-dependent neural pathways and provides a new lens for understanding how modular computational structure emerges in sparse deep networks.

1 INTRODUCTION

The trajectory of deep learning is shaped by a persistent tension between scale and structure. Modern neural networks achieve remarkable performance by scaling to billions of parameters (Kaplan et al., 2020; Hoffmann et al., 2022), at a substantial computational cost (Schwartz et al., 2020; Xu et al., 2018). Yet this success rests on architectures whose parameter utilization is uniform across inputs (Nguyen et al., 2020; Bakhtin et al., 2022; Schrittwieser et al., 2020); where input-conditional routing exists, as in Mixture-of-Experts (Shazeer et al., 2017), it is engineered explicitly rather than emerging from the data. Biological intelligence offers a different template: sparse activity and functional specialization, with distinct subsystems recruited depending on context (Laughlin et al., 1998; Attwell & Laughlin, 2001; Sengupta et al., 2013; Sengupta & Stemmler, 2014; Yeo et al., 2015; Dobs et al., 2022). A key open question is thus whether modern networks can organize their computation in a similarly structured way, allocating different pathways to different regions of the input distribution (Ben-David et al., 2010; Rebuffi et al., 2017).

Pruning provides a natural lens for this question. The Lottery Ticket Hypothesis (LTH) (Frankle & Carbin, 2019) showed that dense networks contain sparse subnetworks, "winning tickets," that can be trained in isolation to match full-model performance. The mask defining such a ticket is extracted from data but static at inference (Evci et al., 2020; Zhou et al., 2019b; Frankle et al., 2021; Burkholz et al., 2022): one subnetwork serves the entire data distribution, and input-conditional computation arises through activation patterns within it rather than through the parameters themselves. Mechanistic interpretability shows that this within-mask flexibility is substantial: the same network can implement qualitatively different computations depending on input context, with an internal organization far more structured and modular than surface-level inspection suggests (Olah et al., 2020; Elhage et al., 2021; Wang et al., 2023; Olsson et al., 2022). What remains open is whether this

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.

organization is also visible at the parameter level: do the tickets that pruning discovers themselves reflect the structure of the data distribution?

We move beyond asking whether a single winning ticket exists and instead ask: *what family of tickets emerges within a network, and how is that family organized across the data distribution?* To study this, we introduce *Routing the Lottery* (RTL), an adaptive pruning framework. Using predefined partitions of the input distribution, RTL discovers *adaptive tickets*—sparse subnetworks associated with different regions of the data distribution—that generalize the universal winning ticket to a data-conditioned family. Rather than converging to a global mask, pruning reveals multiple overlapping subnetworks whose organization reflects the partition. By analyzing their overlap structure, we show that classical winning tickets can be interpreted as structured superpositions of adaptive pathways only within a bounded sparsity regime: universal and adaptive tickets remain aligned at moderate sparsity, but diverge under extreme pruning as pathway-specific structure is discarded. Together, these findings position pruning not only as a compression tool, but as an analytical lens for studying how shared and specialized computation is organized within neural networks.

2 Related work

A central line of work studies conditional computation and modular routing, in which neural networks dynamically select subsets of parameters for each input. Mixture-of-Experts architectures demonstrate that sparse gating can scale model capacity by activating specialized expert subnetworks (Shazeer et al., 2017; Chung et al., 2025). In closely related routing-based and gated architectures, learning decomposes into competing computational pathways: such systems exhibit a "neural race" favoring shared, generalizing pathways (Saxe et al., 2022), face module collapse and interference in practice (Rosenbaum et al., 2019), and their gating structures shape learning dynamics and feature competition (Kleinman et al., 2024; Li & Sompolinsky, 2022).

In parallel, the Lottery Ticket Hypothesis reframes pruning as a mechanism for discovering sparse subnetworks that match dense performance (Frankle & Carbin, 2019; Frankle et al., 2021), but its classical procedure yields a single mask that is static at inference and shared across the entire data distribution. Subsequent work explores mask-based specialization: task-specific masks on a shared backbone (Mallya et al., 2018), iterative pruning that packs multiple tasks into one network (Mallya & Lazebnik, 2018), and supermasks over a fixed random network (Wortsman et al., 2020). These approaches show masks isolate computation and mitigate interference, but they rely on explicit task identity and treat masks as allocation mechanisms over fixed or pretrained weights. In concurrent and independent work, Caso et al. (2026) extract class-specific sparse subnetworks trained with a KL-based uncertainty loss and composed by weight summation; RTL instead uses the unmodified LTH prune–rewind protocol, allows learned overlap on a single shared tensor, extends beyond classification, and analyzes the structural relationship between adaptive and universal tickets.

A complementary line of work studies interpretability and model probing. Attribution and feature-visualization methods probe which inputs drive predictions and what internal units prefer (Simonyan et al., 2014; Selvaraju et al., 2017; Erhan et al., 2009; Olah et al., 2017). Mechanistic interpretability goes further, reverse-engineering the algorithms implemented by trained networks (Olah et al., 2020; Elhage et al., 2021): it has revealed circuits responsible for specific behaviors (Olsson et al., 2022; Wang et al., 2023), polysemantic neurons encoding concepts in superposition (Elhage et al., 2022) that sparse autoencoders can disentangle (Bricken et al., 2023; Templeton et al., 2024), and parameter subsets causally linked to functional roles (Nanda et al., 2023; Bereska & Gavves, 2024). Crucially for our purposes, these analyses show that the same network can implicitly harbor multiple overlapping pathways whose recruitment depends on input context. RTL is inspired by this perspective: rather than learning routing functions or allocating masks over fixed weights, we use pruning to surface latent data-dependent subnetworks, connecting sparsity to modular computation as an analytical framework rather than as a deployment strategy.

3 Method

We propose an adaptive pruning framework that extends LTH to support multiple specialized subnetworks. An overview of the method is shown in Fig. 1. Rather than identifying a single "winning ticket", our approach discovers distinct subnetworks, each tailored to a specific data subset.

Let $f(x;\theta)$ denote a neural network parameterized by $\theta \in \mathbb{R}^d$, with task $\mathcal{T}$ and dataset $\mathcal{D}$. In standard pruning, a binary mask $m \in \{0,1\}^d$ selects a subnetwork f_m via element-wise multiplication:

$$f_m = f(x; m \odot \theta), \tag{1}$$

where $\odot$ denotes the Hadamard product. The LTH posits that there exists a mask m^* such that the corresponding subnetwork, when trained in isolation from random initialization, matches the performance of the dense model, i.e.,

$$\mathcal{L}_{f_m}(\mathcal{D}, \mathcal{T}) = \mathcal{L}_f(\mathcal{D}, \mathcal{T}), \tag{2}$$

where $\mathcal{L}_:$ denotes the model's performance on task $\mathcal{T}$ and dataset $\mathcal{D}$.

While the classical IMP procedure yields a single mask m^* that is static at inference and shared across the entire data distribution, we hypothesize that different data subsets $\mathcal{D}_k$ benefit from distinct subnetworks.

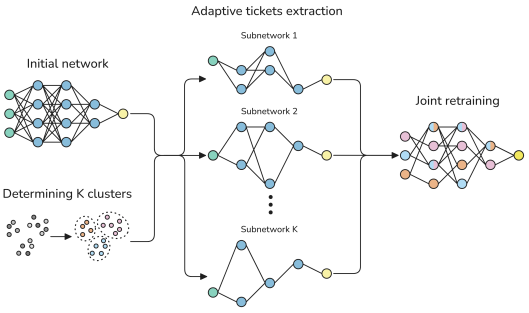

Figure 1: Adaptive pruning pipeline. First, the dataset is divided into subsets via a predefined clustering procedure. We then extract adaptive tickets, and finally perform joint retraining.

3.1 Formulation of the adaptive pruning algorithm

Given a dataset $\mathcal{D}$ and task $\mathcal{T}$, we partition $\mathcal{D}$ into K subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ corresponding to classes or clusters. These labels can be obtained from supervised annotations or derived through unsupervised clustering, enabling flexible adaptation across settings.

For each subset $\mathcal{D}_k$, our method learns a dedicated mask m_k, yielding a subnetwork:

$$\varphi_{m_k} = f(x; m_k \odot \theta), \quad \text{for } x \in \mathcal{D}_k. \tag{3}$$

Given this setup, our objective is to find the set of masks $\mathcal{M}^* = \{m_1^*, \ldots, m_K^*\}$ that minimizes the empirical loss on task $\mathcal{T}$ under a sparsity constraint:

$$\mathcal{M}^* = \underset{\{m_1,\ldots,m_K\}}{\arg\min} \sum_{k=1}^{K} \mathbb{E}_{(x,y)\sim\mathcal{D}_k} \left[\ell\left(y, f\left(x;\, m_k \odot \theta\right)\right)\right] \tag{4}$$

$$\text{subject to} \quad \|m_k\|_0 \le (1-s)\, d \quad \forall k, \tag{5}$$

where ℓ denotes the loss function and $s \in (0,1)$ denotes the target sparsity, so that $(1-s)\,d$ bounds the number of nonzero parameters per subnetwork.

Algorithm 1 Adaptive ticket extraction

Input: Network $f(x;\theta_0)$, training data $\mathcal{D}$, number of subsets K, target sparsity s, pruning factor p, training steps T.
Output: Set of K masks $\mathcal{M}$.

Randomly initialize $f(x;\theta_0)$.
$m_k \leftarrow \mathbf{1}^{|\theta_0|}$ for $k = 1, \ldots, K$.
$\mathcal{D}_K = \{d_1, ..., d_K\}$ {partition of $\mathcal{D}$ into K subsets.}
$\theta \leftarrow \theta_0$
while $\min_k$ sparsity$(m_k) < s$ **do**
  for $k = 1$ **to** K **do**
    $d_k \leftarrow k$-th subset from $\mathcal{D}_K$
    $\theta_T^{(k)} \leftarrow \mathrm{Train}\big(f(\cdot;\, m_k \odot \theta_0),\, d_k,\, T\big)$
    $m_k \leftarrow$ zero out fraction p of active entries of m_k with lowest $|\theta_T^{(k)}|$
    $\theta \leftarrow \theta_0$.
  end for
end while

3.2 Adaptive tickets extraction

We now describe how adaptive tickets are extracted for each data subset (see Alg. 1). Starting from a random initialization $f(x;\theta_0)$, we create an equal-sized binary mask m_k for each of the K target subsets, initialized to ones. The dataset is then partitioned into K disjoint subsets $\mathcal{D}_K = \{d_1, \ldots, d_K\}$ using a predefined rule, such as automatic clustering. Importantly, RTL is agnostic to how these subsets are defined, allowing flexibility across diverse tasks and data modalities.

Each pruning iteration proceeds sequentially over all subsets, and masking is cumulative: at every iteration, the currently masked network $f(x; m_k \odot \theta_0)$ is trained for T steps to obtain temporary parameters $\theta_T^{(k)}$, so training always operates on the subnetwork surviving all previous pruning rounds. We then remove a fraction p of the currently active weights of m_k, selecting those with the lowest magnitude in $\theta_T^{(k)}$; already-pruned weights never re-enter the mask. Pruning is performed globally: active weights are ranked by magnitude across the entire prunable scope (all masked layers jointly) rather than per layer, so local layer sparsities may diverge from the global sparsity level. The remaining weights are then reset to their initial values θ_0. This process repeats until each mask reaches the desired sparsity level s, producing the final set of K adaptive masks $\mathcal{M} = \{m_1, \ldots, m_K\}$. Each mask defines a specialized subnetwork $f(x; m_k \odot \theta_0)$.

3.3 JOINT RETRAINING OF ADAPTIVE TICKETS

After obtaining the sparse masks $\mathcal{M} = \{m_1, \ldots, m_K\}$, we perform a *joint retraining phase* to refine subnetwork performance and reinforce specialization. This step, outlined in Alg. 2, is crucial to our methodology because it preserves the sparsity structure discovered during mask extraction (Alg. 1) and fine-tunes only the active weights without altering mask topology.

The process begins with balanced dataset creation. As before, the training set $\mathcal{D}$ is partitioned into K subsets $\mathcal{D}_K = \{d_1, \ldots, d_K\}$, each corresponding to a target class or cluster. Each subset d_k is divided into mini-batches $\mathcal{B}^k = \{B_1^k, \ldots, B_{M_k}^k\}$. To ensure synchronized updates across subnetworks, we repeat batches from smaller subsets cyclically until all subsets contain the same number of batches, $M = \max(M_1, \ldots, M_K)$. This balancing step guarantees that each subnetwork receives the same number of gradient updates per epoch, even when subsets differ in size.

Algorithm 2 Joint retraining of the adaptive tickets

Input: Network $f(x;\theta)$, training data $\mathcal{D}$, number of subsets K, training epochs N, learning rate η, mask set $\mathcal{M}$.
Output: K optimized adaptive tickets $f_k(\cdot) = f(\cdot; m_k \odot \theta)$ over shared parameters θ.

Balanced dataset creation
$\mathcal{D}_K = \{d_1, ..., d_K\}$ {partition of $\mathcal{D}$ into K subsets.}
for $k = 1$ **to** K **do**
  $d_k \leftarrow k$-th subset from $\mathcal{D}_K$
  $\mathcal{B}^k \leftarrow \{B_1^k, ..., B_{M_k}^k\}$ {set of batches from d_k}
  $M_k \leftarrow |\mathcal{B}^k|$
end for
$M \leftarrow \max(M_1, \ldots, M_K)$.
for $k = 1$ **to** K **do**
  Repeat batches in $\mathcal{B}^k$ cyclically until $|\mathcal{B}^k| = M$.
end for

Model training
for $j = 1$ **to** N **do** {Training epochs}
  for $m = 1$ **to** M **do** {Batch indices}
    for $k = 1$ **to** K **do** {Subnetwork indices}
      $m_k \leftarrow \mathcal{M}[k]$.
      $f_k(\cdot) \leftarrow f(\cdot; m_k \odot \theta)$ {k-th subnetwork}
      $(x_m^k, y_m^k) \leftarrow B_m^k$. {input and label }
      $\theta \leftarrow \theta - \eta \left(\nabla_\theta \mathcal{L}(f_k(x_m^k), y_m^k) \odot m_k\right)$
    end for
  end for
end for

During joint retraining, each subnetwork $f(x; m_k \odot \theta)$ is trained exclusively on its corresponding data subset d_k. We interleave mini-batches from different subsets and apply gradient updates to the shared dense parameter tensor θ, masking out gradients for pruned weights. Specifically, when processing a batch from subset k, the shared parameters are updated as:

$$f_k(\cdot) \coloneqq f(\cdot;\, m_k \odot \theta) \tag{6}$$

$$\theta \leftarrow \theta - \eta \left(\nabla_\theta \mathcal{L}(f_k(x^k), y^k) \odot m_k\right) \tag{7}$$

where η is the learning rate, $\mathcal{L}$ denotes the empirical loss, (x^k, y^k) represents the input data and corresponding label from the k-th subset, and f_k denotes the k-th subnetwork view of the shared tensor θ. Parameters shared across masks receive gradients from multiple partitions, enabling transfer but also potential interference. Mask overlap therefore determines where cross-partition interaction can occur.

4 EXPERIMENTS

We evaluate RTL across four domains, controlling for architecture, sparsity schedule, and training budget throughout to isolate the effects of specialization.

Implicit Neural Representations (INRs). Image reconstruction with coordinate-based networks on 10 ADE20K images (Zhou et al., 2019a), where segmentation masks define region-level subsets and reconstruction quality is measured by PSNR. The number of adaptive tickets K is therefore not a tuned hyperparameter: it equals the number of semantic regions per image, ranging from 3 to 15 across images (Appendix B). This setting is also the primary environment for our structural analyses.

CIFAR-10 and CIFAR-100. On CIFAR-10 (Krizhevsky, 2009), each class receives its own adaptive subnetwork (clean partitions); on CIFAR-100, classes are grouped into eight coarse semantic clusters via unsupervised text-based clustering (ambiguous, overlapping partitions; Appendix D). We report balanced accuracy at 25%, 50%, and 75% sparsity (Appendix C).

Speech enhancement. A realistic heterogeneous setting in which three acoustic environments (*indoor*, *outdoor*, *transportation*) define the specialization subsets; performance is measured by SI-SNRi (Appendix E).

Routing protocol. Partition labels are assumed known (oracle routing) during ticket extraction and joint retraining in all tasks, and at evaluation time for INRs (each pixel's subnetwork is given by

the segmentation mask, known by construction) and speech enhancement (the environment label selects the subnetwork). Reported CIFAR test results are label-free: each test image passes through all K subnetworks, and the prediction is the argmax over the K sigmoid scores. Oracle routing is deliberate: fixing partitions lets us attribute discovered pathway structure to the data distribution and gives our structural metrics a fixed reference, whereas a learned router would make the partition an outcome of optimization, confounding what we measure.

We evaluate predictive performance against a classical IMP baseline, i.e., iterative magnitude pruning with weight rewinding to the original initialization—the canonical procedure used to identify winning tickets (Frankle & Carbin, 2019). RTL and IMP share the same initialization, architecture, sparsity schedule, and training budget in all comparisons; RTL's total training cost is roughly twice that of IMP due to the joint retraining phase (Appendix F). We then use RTL as an analytical framework through five complementary analyses: (i) a **partition-noise control** (INRs, CIFAR-100), testing whether the gains depend on semantically meaningful partitions by progressively randomizing cluster assignments; (ii) **structural collapse and specialization dynamics** (INRs) across iterative pruning; (iii) **zero-shot mask transfer** (INRs) to fully trained, unpruned networks; (iv) **semantic alignment** (CIFAR-10) between structural and semantic similarity of subnetworks; and (v) the **relationship between universal and adaptive tickets** (INRs), analyzed with overlap and coverage metrics calibrated against chance-overlap null models.

5 Results

5.1 Performance on each task

Table 1 summarizes RTL's performance across tasks under varying sparsity levels. Across all settings, RTL achieves higher performance than IMP, indicating that partitioning data into subsets better preserves task-relevant structure during pruning. This is consistent with the hypothesis that data-dependent subnetworks retain parameters important for specific regions of the input distribution, rather than applying uniform compression across all inputs.

5.2 Partition-noise control

The gains in Table 1 admit two alternative explanations: they could arise from the mere act of partitioning the data and training multiple subnetworks, or from the additional effective capacity of maintaining K masks. To rule both out, we evaluate RTL under progressively randomized partitions. We randomly swap X% of cluster assignments (100% = fully random, size-matched partitions). Table 2 reports INR results averaged over the 10 ADE20K images.

Two observations follow. First, the RTL advantage is eliminated, and even reversed, under random partitions: at 100% noise, performance drops to the IMP level at 50% sparsity and below IMP at 75% sparsity. Random partitioning is not merely unhelpful—at higher sparsity it is actively harmful, because capacity is fragmented across subnetworks that do not correspond to any coherent structure

Table 1: Performance comparison of RTL and IMP across tasks under different sparsity levels.

Method	**INR (PSNR ↑)**			**CIFAR-10 (Bal. Acc. ↑)**			**CIFAR-100 (Bal. Acc. ↑)**			**Speech Enh. (SI-SNRi ↑)**		
	25%	50%	75%	25%	50%	75%	25%	50%	75%	25%	50%	75%
RTL	**18.86**	**17.25**	**14.87**	**0.781**	**0.778**	**0.772**	**0.765**	**0.751**	**0.759**	**7.248**	**7.178**	**6.992**
IMP	15.94	14.72	12.69	0.758	0.742	0.707	0.716	0.712	0.705	6.885	6.97	6.967

Table 2: Partition-noise control on INRs. PSNR under progressively randomized partitions, averaged over ADE20K images.

Sparsity	**RTL (semantic)**	**Noise 25%**	**Noise 50%**	**Noise 75%**	**Noise 100%**	**IMP**
25%	**18.86**	17.97	17.17	16.51	16.36	15.94
50%	**17.25**	15.37	14.99	14.32	14.70	14.72
75%	**14.87**	12.76	12.07	11.50	11.45	12.69

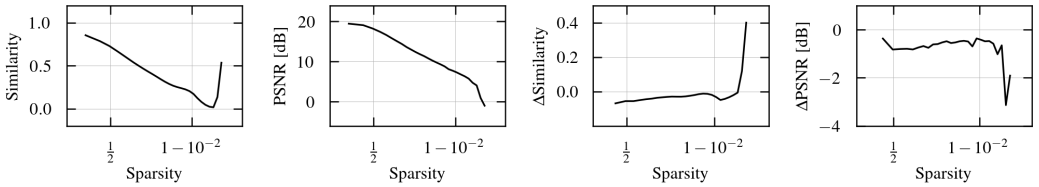

Figure 2: PSNR and pairwise Jaccard similarity as functions of sparsity, together with their finite differences.

in the data. Second, degradation scales with the noise level, yielding a dose–response relationship: the benefit of RTL grows continuously with how much latent structure the partition captures.

This control preserves the number of masks and each mask's capacity: per-mask sparsity is identical across noise conditions, so the gains cannot be attributed to the mere availability of K masks of matched size (the union of masks may still vary with learned overlap, so total union capacity is not fixed by construction). The same pattern holds, and is in fact stronger, on CIFAR-100 under class-level partition noise, where fully random cluster membership drives RTL far below the IMP baseline at every sparsity level (Appendix C).

5.3 Structural collapse and specialization dynamics

To understand how sparsity interacts with latent network structure, we analyze the evolution of adaptive subnetworks throughout iterative pruning. This reveals distinct phases of specialization and identifies the regime in which subnetworks remain both functionally stable and structurally distinct.

Phase I: Progressive differentiation. As sparsity increases, subnetworks initially become more distinct. Pairwise Jaccard similarity decreases monotonically from ≈ 0.85 to a minimum of ≈ 0.016, indicating progressive structural separation between adaptive tickets (Fig. 2). During this phase, reconstruction quality degrades gradually, with PSNR decreasing from ≈ 19.5 dB to ≈ 4–5 dB.

This differentiation is accompanied by a steady increase in heavily specialized weights, whose fraction rises to ≈ 0.68, while the fraction of generalized weights decreases from 1.0 to a minimum of ≈ 0.14 (Fig. 3). These trends indicate that pruning progressively reallocates capacity away from shared computation and toward data-dependent pathways.

Interestingly, this specialization is not uniform across depth. Layers 1–4 become increasingly sparse and highly specialized, while Layer 5 remains substantially denser throughout pruning, suggesting that the output layer preserves the shared capacity required for stable reconstruction.

Phase II: Peak specialization regime. Near global sparsity of $\approx 99.4\%$–99.6%, the system reaches a transition point at which specialization is maximized. In this regime, mask similarity is minimal, heavily specialized weights peak, and performance remains functional despite extreme sparsity.

This identifies a practical operating region for RTL: subnetworks extracted in this regime are maximally differentiated while still preserving task performance.

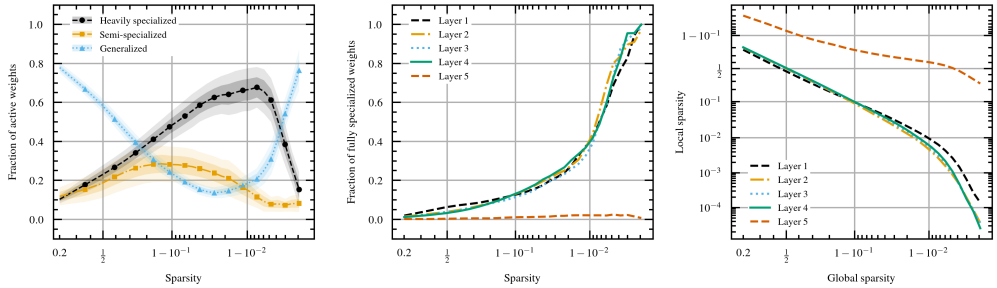

Figure 3: Weight specialization dynamics. Left: fraction of active weights categorized as heavily specialized ($< 1/3$ masks), semi-specialized, and generalized ($> 2/3$ masks). Middle: per-layer specialization fraction. Right: local layer sparsity vs. global sparsity.

Phase III: Structural collapse. Beyond $\approx 99.6\%$ sparsity, the system undergoes an abrupt transition. Finite differences reveal a sharp sign reversal in pairwise similarity, changing from gradual decreases to large positive jumps, while PSNR exhibits catastrophic drops exceeding 3 dB per pruning step.

At the same time, the fraction of heavily specialized weights collapses, generalized weights rapidly re-emerge, and pairwise similarity increases sharply from near-zero overlap to ≈ 0.53. This indicates that extreme pruning destroys specialized pathways and forces subnetworks to converge to increasingly overlapping solutions. Notably, this collapse occurs shortly before complete pruning, suggesting that the critical failure is not gradual capacity loss but a sharp structural crossover in which subnetworks can no longer maintain distinct functional roles.

5.4 Probing latent structure via zero-shot mask transfer

To assess whether RTL captures structure already present in dense models rather than artifacts introduced by pruning, we transfer learned masks to a fully trained, unpruned network. We consider two conditions: (1) structural composition, in which subsets of weights identified by RTL (generalized and specialized vs. complementary subsets) are applied at 50% sparsity, and (2) selective pathway masking, in which regions identified as specialized at 2% density are removed (full results in Appendix B.14). All values below are reported as absolute PSNR, where negative values indicate reconstruction quality below the original signal noise floor due to abrupt masking without adaptation. All results in this section are computed on a single ADE20K image (index 0), since the analysis depends on the number of segmented objects; similar patterns were observed across the other images (see Appendix B).

Table 3 reports results for 50% sparsity masks transferred to the unpruned backbone. The combined generalized and specialized subset achieves a PSNR of -1.09 dB, substantially outperforming both random masking (-4.10 dB) and the complement of the RTL-selected subset (-4.18 dB). Although all masks degrade performance in the zero-shot setting, the superior preservation of signal under RTL-derived masks suggests that the identified subsets capture functionally meaningful structure already present in the dense model.

Table 3: Zero-shot mask transfer on unpruned models.

Method	Avg	Obj1	Obj2	Obj3	Obj4
Dense	14.26	30.36	13.62	16.55	31.08
RTL	-1.09	0.70	-0.44	-2.74	2.87
IMP (LTH)	-1.90	-0.07	-1.32	-3.39	-1.50
Shared	-2.57	-1.63	-1.50	-4.87	-0.01
Specialized	-4.33	7.19	-4.74	-3.08	-2.92
Random	-4.10	2.86	-4.19	-3.90	-2.04
Complement	-4.18	7.36	-4.54	-3.16	-2.96

We additionally transfer the universal LTH mask at the same sparsity. The LTH mask (-1.90 dB) outperforms all structural controls, which is itself consistent with the superposition view: at this moderate sparsity, the universal ticket already approximates a combination of adaptive pathways and inherits part of their transferable structure. The RTL-derived mask nonetheless outperforms it on average and on all four individual objects. The resulting ordering, RTL above LTH above the structural controls, is exactly what the superposition view predicts: the universal ticket is an entangled approximation of the adaptive family, and disentangling the pathways explicitly recovers strictly more of the dense model's latent functional organization. In contrast, retaining only specialized weights leads to lower performance (-4.33 dB). While these weights preserve localized information, they lack the shared components required for stable reconstruction across the full input, indicating that coherent computation depends on both shared and specialized parameters.

Transferred masks also retain recoverable structure: under short fine-tuning with masks held fixed, RTL masks start from a better zero-shot initialization, recover faster, and converge to higher final PSNR than random and reversed masks (Appendix B.13).

Selective pathway masking shows a complementary effect: removing an individual specialized pathway disproportionately degrades specific semantic objects, indicating that different regions of the model contribute unevenly to different components of the input (Appendix B.14).

5.5 Semantic alignment of adaptive subnetworks

To assess whether adaptive subnetworks reflect semantic structure in the data, we measure the Spearman correlation between semantic similarity derived from WordNet and pairwise mask simi-

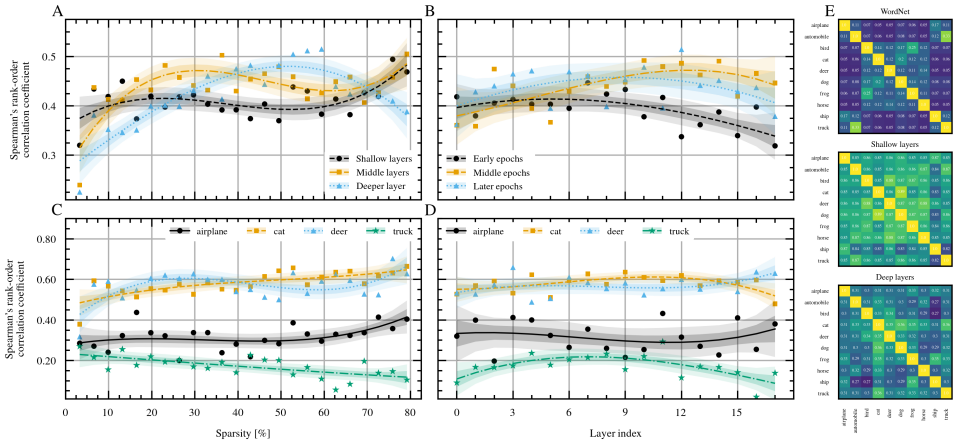

Figure 4: Semantic and structural correlation analysis. (A) Spearman correlation between semantic similarity (WordNet) and mask similarity across pruning levels for shallow, middle, and deep layers. (B) Correlation across network depth at early, middle, and late training stages. (C) Correlation across pruning levels and (D) across depth for representative classes. (E) WordNet similarity matrix (top) and corresponding mask similarity matrices for shallow (middle) and deep (bottom) layers.

larity. Higher correlation indicates that semantically related classes tend to share more parameters, suggesting that pruning preserves meaningful structural relationships.

Figure 4A shows how this alignment evolves across pruning levels. Shallow and middle layers exhibit steadily increasing correlation throughout pruning, rising from approximately 0.37/0.32 to 0.48/0.49 at 79% sparsity. This suggests that moderate pruning progressively removes redundant parameters while preserving shared low- and mid-level features among semantically related classes.

Deep layers follow a different trajectory. Correlation initially increases from 0.29 to a peak of approximately 0.48 near 50–55% sparsity, indicating that pruning strengthens semantic specialization in higher-level representations. Beyond this point, however, correlation declines toward 0.38 as sparsity approaches 80%. This degradation coincides with the onset of subnetwork collapse observed in Section 5.3, where excessive pruning forces adaptive subnetworks toward structurally degenerate solutions. Thus, the late-stage decline likely reflects the erosion of meaningful specialization.

Figure 4B analyzes alignment across network depth at different stages. Early in training, correlations remain relatively flat before gradually decreasing in deeper layers, indicating weak semantic organization. As optimization progresses, semantic alignment strengthens across the network, with the middle and late training stages exhibiting clear peaks in intermediate-to-deep layers. This suggests that semantic structure emerges progressively and becomes increasingly localized to deeper representations.

Class-specific trends (Figure 4C–D) vary substantially across classes and are reported as exploratory analyses in Appendix C.8. Figure 4E further shows that shallow layers remain highly overlapping and uniform, whereas deeper layers develop a clearer block structure aligned with WordNet groupings.

5.6 Structural relationship between universal and adaptive tickets

To study the relationship between classical LTH winning tickets and RTL-discovered pathways, we analyze whether universal winning tickets can be interpreted as compositions of adaptive subnetworks discovered under data-dependent partitions. Intuitively, if LTH winning tickets represent a universal solution to heterogeneous data, then their active parameters should largely coincide with, and be explainable through, the union of adaptive tickets extracted by RTL.

For each sparsity level, we compare the universal LTH mask against the set of adaptive masks using five structural metrics (definitions are provided in Appendix A): *coverage*, which measures what fraction of the universal mask is explained by the union of adaptive tickets; *precision*, which measures what fraction of adaptive parameters are retained by the universal mask; *overlap*, which measures the average number of adaptive tickets that reuse a parameter from the universal mask; *core ratio*, which

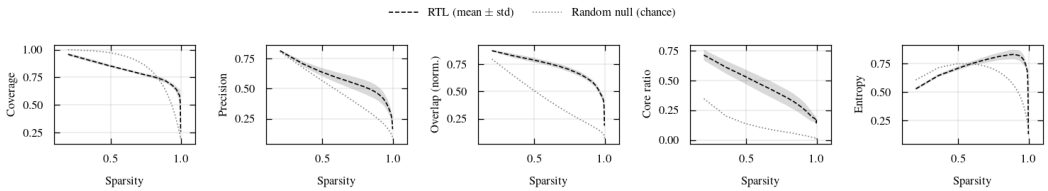

Figure 5: Structural relationship between universal LTH masks and RTL adaptive tickets.

quantifies the fraction of universal parameters shared across all adaptive tickets; and *entropy*, which measures how evenly universal parameters are distributed across pathways.

We calibrate metrics against random masks matched in number and per-layer density (Appendix B.15), interpreting only excess over chance baseline. The calibrated analysis reveals three regimes (Fig. 5). At low and moderate sparsity (<80%), universal winning tickets align strongly with adaptive pathways in absolute terms: coverage remains high (>0.75), precision exceeds 0.5, and overlap is above 0.6. However, the null itself saturates in this regime—random unions cover nearly all parameters (null coverage 0.95–1.0)—so high raw coverage carries little evidence on its own. The above-chance signal is instead carried by the sharing structure: the core ratio exceeds its null by a factor of 2–5 (0.71 vs. 0.35 at 20% sparsity; 0.35 vs. 0.07 at 79%), and normalized overlap and pairwise mask similarity likewise remain far above chance. Universal parameters are reused across pathways far more consistently than random placement allows, indicating a genuine shared backbone.

In the sparse regime (approximately 83–99.5%), chance overlap becomes negligible and every alignment metric exceeds its null by a wide margin: coverage reaches up to 2.3× its null (0.54 vs. 0.24 at 99.3% sparsity), precision roughly 2×, normalized overlap 3–4×, and core ratio 5–13× (0.16 vs. 0.01). Entropy remains above null throughout this regime, peaking near 0.83 where universal parameters are most evenly distributed across specialization levels. The superposition interpretation is therefore strongest precisely where it is hardest to obtain by chance: most parameters retained by the universal ticket are recoverable from adaptive pathways, and are reused across them, well beyond what density alone predicts.

Beyond approximately 99.5% sparsity, the excess over chance contracts sharply, but not uniformly across metrics. Pairwise Jaccard and coverage approach their density-matched nulls (0.69 vs. 0.68 and 0.77 vs. 0.73 at 99.8% sparsity), and entropy falls below null, while the universal-conditioned sharing metrics retain residual excess (precision 0.48 vs. 0.34, normalized overlap 0.19 vs. 0.03). Paired per-image comparisons confirm this picture: all excesses remain statistically significant, but those of pairwise similarity and coverage shrink by an order of magnitude relative to the structured regime (Appendix B.15). Convergence toward chance at collapse is therefore substantial but partial and metric-dependent: much of the apparent re-convergence of masks at extreme sparsity is accounted for by surviving weights concentrating in a few dense layers rather than by meaningful re-organization, although a residual of universal-aligned sharing structure persists.

6 Conclusion

We introduced Routing the Lottery (RTL), an adaptive pruning framework that extends classical winning-ticket extraction from a single universal mask, static at inference, to a family of data-conditional subnetworks, which we term *adaptive tickets*. Rather than treating pruning purely as compression, RTL uses sparsity to study how computation is organized across heterogeneous inputs within a shared network.

Our analysis shows that adaptive subnetworks emerge only within a bounded sparsity regime: as pruning progresses, subnetworks initially differentiate into structured patterns of shared and specialized pathways, but under extreme sparsity they collapse toward overlapping solutions, accompanied by abrupt performance degradation and loss of structural diversity. Across tasks, adaptive tickets expose meaningful internal structure, including shared-specialized decompositions, robustness to imperfect partitions, and increasing semantic alignment in deeper layers. Together, these results suggest that classical winning tickets can be interpreted as structured superpositions of adaptive pathways only within a limited sparsity regime, positioning pruning as a tool not only for compression, but also for analyzing neural organization.

AI USE STATEMENT

In this work, we did not use generative AI tools for any task requiring disclosure. Specifically, we have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, or interpret results. Generation of synthetic data sets, translation, cleaning and reformatting of datasets, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability, including polishing prose and condensing sections to meet the page limit. We have reviewed all AI-assisted work. AI assistance was restricted to the surface form of already-written text: all technical content, claims, numbers, and structure were authored by the authors, and every edited passage was read back against the original by an author to confirm that no claim, qualifier, or attribution was altered, weakened, or introduced. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

D Attwell and S B Laughlin. An energy budget for signaling in the grey matter of the brain. *J Cereb Blood Flow Metab*, 21(10):1133–1145, October 2001.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of *Diplomacy* by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Rebekka Burkholz, Nilanjana Laha, Rajarshi Mukherjee, and Alkis Gotovos. On the existence of universal lottery tickets. In *International Conference on Learning Representations*, 2022.

Francesco Caso, Samuele Fonio, Simone Monaco, Nicola Saccomanno, and Fabrizio Silvestri. Composable sparse subnetworks via maximum-entropy principle. In *International Conference on Learning Representations*, 2026.

Youngseog Chung, Dhruv Malik, Jeff Schneider, Yuanzhi Li, and Aarti Singh. Beyond parameter count: Implicit bias in soft mixture of experts. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Katharina Dobs, Julio Martinez, Alexander J. E. Kell, and Nancy Kanwisher. Brain-like functional specialization emerges spontaneously in deep neural networks. *Science Advances*, 8(11):eabl8913, 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

D. Erhan, Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. In *Technical report, University of Montreal*, 2009.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021.

Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1577–1586, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. IEEE, 2015.

Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. Tau urban acoustic scenes 2020 3class, development dataset, February 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Michael Kleinman, Alessandro Achille, and Stefano Soatto. Critical learning periods emerge even in deep linear networks. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Simon B. Laughlin, Rob R. de Ruyter van Steveninck, and John C. Anderson. The metabolic cost of neural information. *Nature Neuroscience*, 1(1):36–41, May 1998. ISSN 1546-1726.

Qianyi Li and Haim Sompolinsky. Globally gated deep linear networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, pp. 72–88, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01224-3.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023.

Thai Son Nguyen, Sebastian Stueker, and Alexander H. Waibel. Super-human performance in online low-latency recognition of conversational speech. In *Interspeech*, 2020.

Christopher Olah, Ludwig Schubert, and Alexander Mordvintsev. Feature visualization. *Distill*, 2017.

Christopher Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 506–516, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Chandan KA Reddy, Vishak Gopal, Ross Cutler, Ebrahim Beyrami, Roger Cheng, Harishchandra Dubey, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. In *Proc. Interspeech 2020*, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation. *ArXiv*, abs/1904.12774, 2019.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The neural race reduction: Dynamics of abstraction in gated networks. In *International Conference on Machine Learning*, pp. 19287–19309. PMLR, 2022.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, dec 2020.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12): 54–63, November 2020. ISSN 0001-0782.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

Biswa Sengupta and Martin B. Stemmler. Power consumption during neuronal computation. *Proceedings of the IEEE*, 102(5):738–750, 2014.

Biswa Sengupta, Simon B. Laughlin, and Jeremy E. Niven. Balanced excitatory and inhibitory synaptic currents promote efficient coding and metabolic efficiency. *PLOS Computational Biology*, 9(10):1–12, 10 2013.

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

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Xiaowei Xu, Yukun Ding, Sharon Xiaobo Hu, Michael Niemier, Jason Cong, Yu Hu, and Yiyu Shi. Scaling for edge inference of deep neural networks. *Nature Electronics 2018 1:4*, 1:216–222, 4 2018. ISSN 2520-1131.

B. T. Thomas Yeo, Fenna M. Krienen, Simon B. Eickhoff, Siti N. Yaakub, Peter T. Fox, Randy L. Buckner, Christopher L. Asplund, and Michael W.L. Chee. Functional specialization and flexibility in human association cortex. *Cerebral Cortex*, 25(10):3654–3672, 10 2015. ISSN 1047-3211.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019a.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. *Deconstructing lottery tickets: zeros, signs, and the supermask*. Curran Associates Inc., Red Hook, NY, USA, 2019b.

A METRICS

A.1 STRUCTURAL SIMILARITY METRICS

To analyze the structural organization of adaptive tickets, we compute several metrics based on extracted binary masks.

A.1.1 PAIRWISE JACCARD SIMILARITY

To measure structural overlap between two subnetworks represented by binary masks M_i and M_j, we use the Jaccard similarity:

$$J(M_i, M_j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|}. \tag{8}$$

This metric ranges from 0 to 1, where 0 indicates disjoint subnetworks and 1 indicates identical masks. Unless stated otherwise, we compute pairwise similarities between adaptive tickets and report size-weighted averages across layers.

A.1.2 SPECIALIZATION RATIO

To quantify the fraction of parameters unique to a given adaptive ticket M_t, we define:

$$\mathrm{Spec}(M_t) = \frac{\left|M_t \setminus \bigcup_{o \neq t} M_o\right|}{|M_t|}. \tag{9}$$

This metric also ranges from 0 to 1. Higher values indicate stronger specialization, corresponding to a larger fraction of parameters used exclusively by a given subnetwork, whereas lower values indicate greater parameter sharing across tickets.

A.2 STRUCTURAL METRICS FOR UNIVERSAL AND ADAPTIVE TICKET ANALYSIS

To analyze the relationship between universal winning tickets obtained via standard IMP pruning and adaptive tickets discovered by RTL, we define a set of structural metrics that characterize alignment, sharing, and specialization between masks.

Let $m_{\text{LTH}} \in \{0,1\}^d$ denote the binary universal winning ticket mask obtained from IMP, and let $\{m_i\}_{i=1}^K$ denote the set of adaptive masks discovered by RTL for K data partitions. We define the adaptive union mask as:

$$m_{\text{union}} = \bigvee_{i=1}^{K} m_i,$$

where $\vee$ denotes element-wise logical OR.

For each parameter w, we define its pathway count as:

$$c(w) = \sum_{i=1}^{K} m_i(w),$$

which measures how many adaptive tickets retain parameter w.

A.2.1 COVERAGE

Coverage measures the fraction of the universal ticket explained by adaptive tickets:

$$\text{Coverage} = \frac{|m_{\text{LTH}} \cap m_{\text{union}}|}{|m_{\text{LTH}}|}.$$

Coverage ranges from 0 to 1.

- 1: all universal parameters are recovered by adaptive tickets,
- 0: there is no overlap between the universal and adaptive masks.

High coverage indicates that the universal winning ticket can largely be decomposed into adaptive pathways.

A.2.2 PRECISION

Precision measures the fraction of adaptive parameters retained by the universal ticket:

$$\text{Precision} = \frac{|m_{\text{LTH}} \cap m_{\text{union}}|}{|m_{\text{union}}|}.$$

Precision ranges from 0 to 1.

- high precision: adaptive pathways are mostly contained within the universal ticket,
- low precision: adaptive pathways contain many parameters absent from the universal mask.

A.2.3 CONDITIONAL OVERLAP

Conditional overlap measures the average number of adaptive tickets that reuse a universal parameter, conditioned on the parameter being covered by at least one adaptive ticket:

$$\text{Overlap} = \frac{\sum_{w \in m_{\text{LTH}}} c(w)}{|m_{\text{LTH}} \cap m_{\text{union}}|}.$$

Since overlap ranges from 1 to K, we report a normalized version:

$$\text{Overlap}_{\text{norm}} = \frac{\text{Overlap} - 1}{K - 1},$$

which maps overlap to $[0, 1]$.

- 0: each covered universal parameter belongs to exactly one adaptive pathway,
- 1: all covered universal parameters are shared across all pathways.

Higher overlap indicates stronger parameter sharing and a larger shared computational backbone across adaptive tickets.

A.2.4 CORE RATIO

Core ratio measures the fraction of universal parameters that belongs to the shared backbone present in all adaptive tickets:

$$\text{Core} = \frac{|\{w \in m_{\text{LTH}} : c(w) = K\}|}{|m_{\text{LTH}}|}.$$

Core ratio ranges from 0 to 1.

- high values indicate a large shared backbone,
- low values indicate reduced universal sharing and increased specialization.

A.2.5 PATHWAY ENTROPY

To quantify how evenly universal parameters are distributed across pathways, we define:

$$p_j = \frac{|\{w \in m_{\text{LTH}} : c(w) = j\}|}{|m_{\text{LTH}}|},$$

for $j \in \{0, 1, \ldots, K\}$.

Entropy is computed as:

$$H = -\sum_{j=0}^{K} p_j \log p_j,$$

and normalized as:

$$H_{\text{norm}} = \frac{H}{\log(K+1)},$$

yielding values in $[0, 1]$ independent of the number of pathways. The normalizer is $\log(K+1)$ rather than $\log K$ because the reuse count takes $K+1$ possible values, $j \in \{0, 1, \ldots, K\}$.

Higher entropy indicates a more diverse distribution of parameter reuse across specialization levels, whereas lower entropy indicates concentration in a narrow subset of pathway types.

In practice:

- low entropy often corresponds to dominance of either shared core parameters or highly specialized parameters,
- high entropy indicates structurally diverse pathway composition.

B IMPLICIT NEURAL REPRESENTATION

B.1 TASK SETUP

We consider the standard INR formulation in which continuous pixel coordinates (x, y) are mapped to RGB values. The full image is reconstructed through point-wise evaluation of the network over all pixel locations.

Experiments are conducted on 10 images selected from the training split of the ADE20K dataset to ensure diversity in scene categories and luminosity and color characteristics. Specifically, we use images with indices 0, 3733, 7304, 8399, 11708, 12963, 13783, 14236, 18813, and 23790. Each image is treated independently, and a separate model is trained for each image. The selected images and their corresponding preprocessed semantic segmentation masks are shown in Fig. 6. All INR experiments use the full set of 10 images, except for the analysis in Sec. 5.4, where results are reported only for the first image (index 0) because that analysis depends on the number of segmented objects; similar patterns were observed across all images.

Semantic segmentation annotations are used to define classes: all masks corresponding to the same semantic object category within an image are merged into a single class (e.g., multiple instances of *tree* are treated as one mask).

The number of adaptive tickets K is not a tuned hyperparameter: for each image, it equals the number of unique semantic classes in that image's segmentation mask, ranging from 3 to 15 across the 10 images. Per-image values are reported in Table 4. Because K varies across images while all other conditions are held fixed, the per-image results in Table 6 also serve as a natural sweep over K: RTL outperforms IMP on every image at every sparsity level, and the collapse dynamics are qualitatively identical across the full range of K. Since this per-image evidence is observational (K covaries with image content), a controlled ablation that decouples K from content is provided in Appendix B.12.

Table 4: Number of adaptive tickets K per ADE20K image, equal to the number of unique semantic regions in the image's segmentation mask.

Image index	0	3733	7304	8399	11708	12963	13783	14236	18813	23790
K	4	15	9	4	12	3	6	8	8	4

The dataset contains small unassigned or noisy regions, typically at object boundaries. These regions are reassigned to a neighboring semantic class, which empirically stabilizes training for both RTL and baseline models.



Figure 6: Samples from the ADE20K dataset used in the INR experiments, shown together with their corresponding preprocessed semantic segmentation masks.

B.2 Positional Encoding

Input coordinates (x, y) are encoded using Fourier features before being passed to the network, following standard INR practice. This encoded representation forms the sole input to the RTL model.

B.3 Model Architectures

All models are multilayer perceptrons (MLPs) with ReLU activations. For RTL, the network consists of five layers with the following input-–output dimensions: $[[128, 34], [128, 128], [128, 128], [128, 128], [3, 128]]$.

For the standard baseline, the input is augmented with a learned class embedding corresponding to the semantic mask of each pixel, increasing the first-layer dimensionality to $[128, 50]$. All subsequent layers are identical to those used in RTL. For the analysis in Sec. 5.4, however, we use the baseline without class embeddings (i.e., input dimensionality $[128, 34]$) to ensure mask compatibility and enable transfer to the dense model.

B.4 Training and pruning protocol

All models are trained using the Adam optimizer with a learning rate of 0.01. Training proceeds for 10,000 optimization steps, after which pruning is applied.

For the performance evaluation in Sec. 5.1, pruning is performed by globally removing the 4,096 lowest-magnitude weights per iteration (ranked across all prunable layers jointly) until the target sparsity levels (25%, 50%, and 75%) are reached. For all other analyses, we instead use global iterative magnitude pruning with a fixed pruning ratio of 20% of the currently active weights per step, enabling finer resolution near extreme sparsity and more precise estimation of pathway dynamics.

Following each pruning step, all remaining (non-pruned) weights are rewound to their initial values, consistent with the IMP-style protocol used throughout the paper. This prune-and-rewind process is repeated until the target sparsity is reached.

B.5 Initialization and fairness considerations

Because the RTL and baseline models differ in input dimensionality, care is taken to ensure fair initialization. We initialize the larger baseline network, including the class-embedding input, and manually remove the weights corresponding to the class-embedding dimensions to obtain the RTL initialization. This ensures that all shared parameters are initialized identically across methods.

The same initialization seed is used for all images.

B.6 Evaluation protocol

Reconstruction quality is evaluated using peak signal-to-noise ratio (PSNR). In the main text, PSNR is reported as a single scalar value averaged over all pixels and all images. For completeness, we additionally report per-image PSNR values in the appendix to illustrate variability across samples.

B.7 Image reconstruction results

Table 5 reports reconstruction performance for the INR experiment on ADE20K images at varying sparsity levels.

Table 5: INR results on the ADE20K dataset. Methods are evaluated at 25%, 50%, and 75% sparsity using PSNR, together with the number of trainable parameters.

Method	**PSNR ↑**			**#Params ↓**		
	25%	50%	75%	25%	50%	75%
RTL (ours)	**18.86**	**17.25**	**14.87**	47.9K	35.4K	20.5K
IMP	15.94	14.72	12.69	**40.4K**	**26.9K**	**13.5K**

Across all sparsity regimes, RTL consistently outperforms the standard IMP baseline. At 25% sparsity, RTL achieves a PSNR of 18.86, exceeding the single-model IMP baseline by almost 3 dB. This performance gap remains substantial as sparsity increases, with RTL maintaining advantages of 2.53 dB and 2.18 dB at 50% and 75% sparsity, respectively.

Notably, RTL achieves these gains while retaining a larger number of parameters than the single-model IMP baseline. This behavior mirrors observations from the classification and speech-enhancement experiments: enforcing a single global pruning mask across heterogeneous subsets, in this case semantically distinct image regions, leads to suboptimal allocation of model capacity. By contrast, RTL enables region-specific subnetworks that preserve functionally important parameters, resulting in higher reconstruction fidelity under identical computational budgets.

As sparsity increases, performance degrades for both methods, but RTL degrades more gracefully, indicating that specialization is especially beneficial in high-sparsity regimes, where competition for shared parameters intensifies. These results demonstrate that adaptive pruning can effectively capture

semantic structure even in coordinate-based representations, extending the benefits of specialization beyond dataset-level tasks to within-image semantic decomposition.

Per-image PSNR values are reported in Appendix B.8, confirming that the observed trends are consistent across images with diverse scene content and appearance characteristics.

B.8 PER-IMAGE INR RESULTS

Table 6 reports per-image reconstruction performance for the INR experiment on 10 ADE20K images. Results are shown for RTL and the single-model IMP baseline at 25%, 50%, and 75% sparsity, measured using PSNR.

Table 6: Per-image PSNR for the INR experiment on 10 ADE20K images. Results are reported for RTL and the single-model IMP baseline at 25%, 50%, and 75% sparsity.

	PSNR ↑					
Sample	**RTL**			**IMP (single model)**		
	25%	50%	75%	25%	50%	75%
1	13.73	12.08	9.68	11.51	10.14	8.39
2	10.40	9.27	7.94	8.60	7.95	6.83
3	17.59	16.92	15.66	16.32	15.45	13.96
4	24.46	22.58	20.56	21.58	20.19	17.79
5	18.52	16.88	13.99	14.63	12.99	10.96
6	19.01	17.23	14.50	16.63	15.44	13.24
7	25.73	23.14	19.30	19.81	18.62	15.96
8	15.55	14.08	11.97	12.47	11.45	9.79
9	19.34	17.73	15.41	16.81	15.43	13.17
10	24.31	22.63	19.68	21.07	19.53	16.78

Across all samples and sparsity levels, RTL consistently achieves higher PSNR than the single-model IMP baseline. This performance gap is observed for images with diverse scene content and appearance characteristics, confirming that the aggregate improvements reported in the main text are not driven by a small subset of favorable samples.

The advantage of RTL is particularly pronounced at higher sparsity levels, where enforcing a single global pruning mask across semantically heterogeneous regions leads to larger reconstruction errors. In contrast, RTL preserves region-specific parameters through specialized subnetworks, resulting in more stable degradation as sparsity increases.

These per-image results complement the averaged PSNR values reported in the main text by demonstrating that the benefits of adaptive pruning in INR settings are consistent across images, rather than arising from outliers or dataset bias. Figure 7 shows the relationship between reconstruction quality and pruning mask similarity for each of the 10 ADE20K images (A–J) used in the INR experiment. For each image, we report PSNR (orange solid line, left axis) and average mask similarity measured by the Jaccard index (blue dashed line, right axis) as a function of sparsity.

Across all images, PSNR decreases gradually as sparsity increases, followed by a sharper drop at high sparsity levels. A similar trend is observed for mask similarity: pruning masks remain relatively stable at low to moderate sparsity, but their similarity rapidly decreases beyond a critical sparsity threshold. This transition is consistent across images, despite substantial differences in scene content and appearance.

Notably, the sharp decline in PSNR closely coincides with the point at which mask similarity reaches its minimum, i.e., with peak differentiation between region-specific subnetworks. This indicates

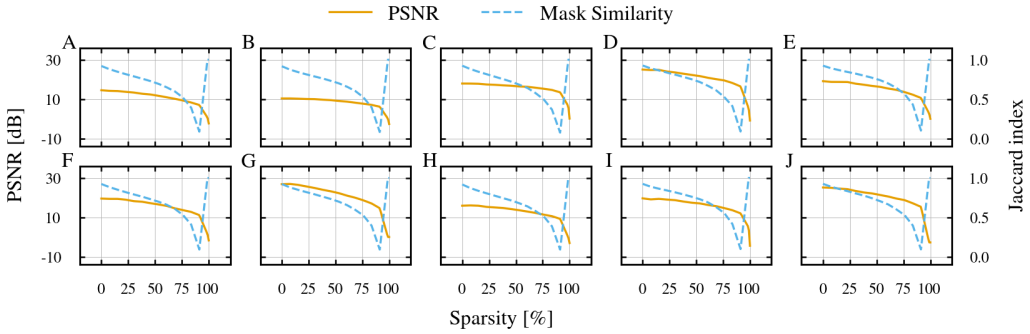

Figure 7: Per-image relationship between reconstruction quality and pruning mask similarity in the INR experiment. For each ADE20K image (A–J), PSNR (orange, left axis) and average mask similarity measured by the Jaccard index (blue dashed, right axis) are shown as functions of sparsity.

that reconstruction quality degrades most significantly at the boundary of the peak-specialization regime, rather than as a direct consequence of parameter removal alone. At very high sparsity, mask similarity approaches zero, reflecting near-disjoint subnetworks operating at the edge of the collapse regime analyzed in Sec. 5.3, in which masks subsequently re-converge toward overlapping degenerate solutions.

These per-image trends mirror the averaged results reported in the main text and further support the interpretation that maintaining distinct, region-specific pruning masks is critical for preserving reconstruction performance in high-sparsity INR settings.

B.9 INR SUBNETWORK SIMILARITY VS. RECONSTRUCTION QUALITY

We further analyze the relationship between subnetwork specialization and reconstruction quality in the INR setting by examining how per-region PSNR correlates with mask similarity.

Fig. 8 reports results for a single ADE20K image containing four semantic regions. For each region-specific subnetwork, we plot PSNR together with the average Jaccard similarity of its pruning mask to all other region masks across sparsity levels. PSNR remains relatively stable while mask similarity is low, but degrades sharply once similarity increases, indicating the onset of subnetwork collapse.

Fig. 9 extends this analysis to a second ADE20K image with fifteen semantic regions. Despite the larger number of classes, the same trend holds across all regions (A–O): reconstruction quality deteriorates rapidly once subnetworks lose structural distinctiveness under aggressive pruning.

To isolate correlation effects independent of absolute scale, Fig. 10 replots the 15-region case with both PSNR and mask similarity mean-centered (panels A–O). Across regions, fluctuations in reconstruction quality closely track changes in mask similarity, revealing an inverse specialization

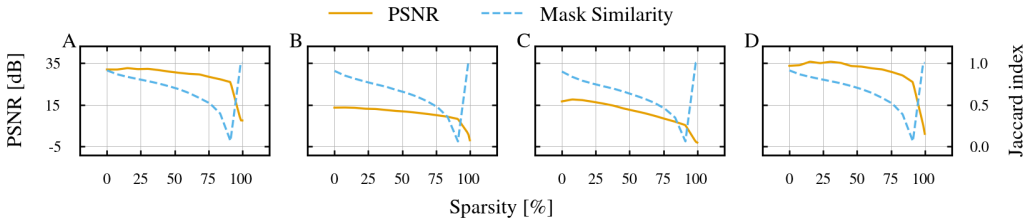

Figure 8: Per-class PSNR and subnetwork similarity for a single ADE20K image with four semantic regions. PSNR (solid) and average mask similarity to all other region-specific subnetworks (dashed) are shown as functions of sparsity. Performance degradation coincides with a rapid increase in mask similarity, indicating subnetwork collapse at high pruning ratios.

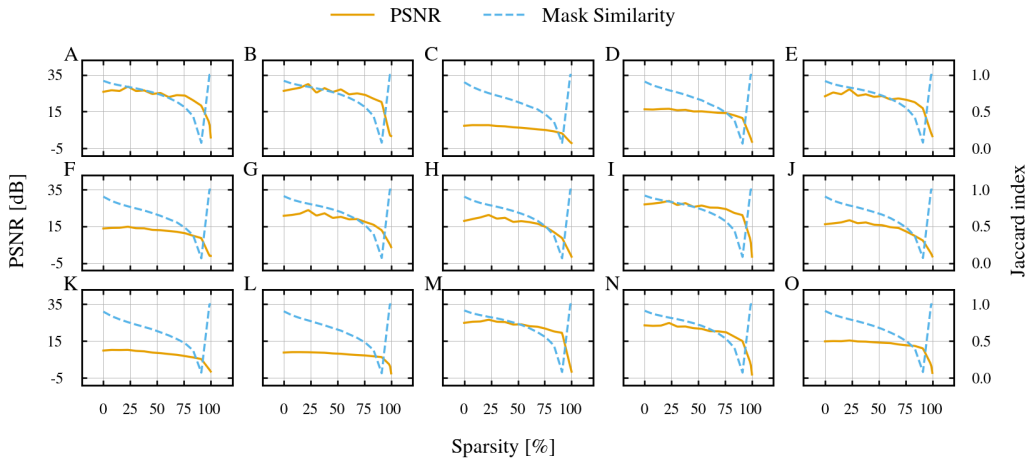

Figure 9: Per-class PSNR and subnetwork similarity for a second ADE20K image with fifteen semantic regions. Each subplot (A–O) corresponds to one semantic region. As sparsity increases, PSNR declines gradually until a sharp drop aligns with increasing similarity between region-specific pruning masks, reflecting loss of structural specialization.

effect: regions that maintain higher PSNR tend to exhibit greater mask overlap, while regions with lower PSNR show increased structural specialization. This suggests that well-reconstructed regions rely on more shared parameters, whereas harder regions benefit from stronger subnetwork differentiation.

Panel P summarizes this relationship at 50% sparsity by plotting PSNR against mask similarity for all regions and fitting a linear regression. We observe a strong monotonic relationship, with an overall Spearman rank correlation of $\rho \approx 0.982$. This confirms that subnetwork similarity is a strong predictor of reconstruction quality in INRs, even within a single image.

Together, these results reinforce the central claim of RTL: preserving subnetwork distinctiveness is critical not only across datasets or tasks, but also for fine-grained within-image semantic specialization. Mask similarity thus provides a reliable, label-free diagnostic for identifying oversparsification and impending performance collapse in coordinate-based representations.

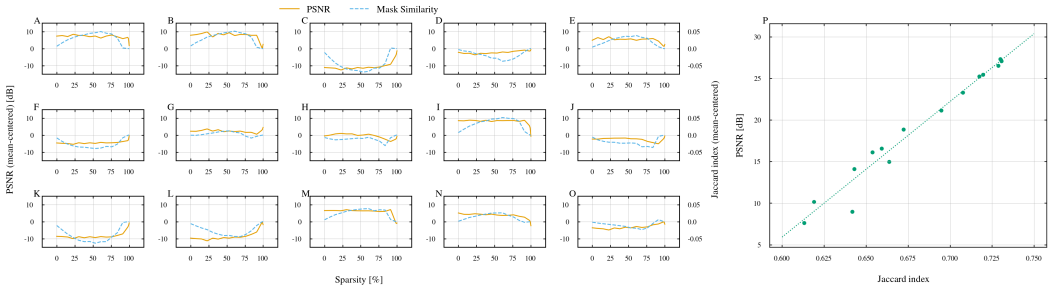

Figure 10: Correlation between reconstruction quality and subnetwork similarity in INRs. Panels A—O show mean-centered PSNR and mean-centered mask similarity for the fifteen-region image. Panel P reports a linear regression between the two quantities at 50% sparsity across regions, revealing a strong monotonic relationship (Spearman $\rho \approx 0.982$).

B.10 Qualitative INR Reconstructions under High Sparsity

To complement the quantitative PSNR analysis, we provide qualitative visualizations of INR reconstructions under increasing sparsity levels. Figures 12, 13, and 14 show reconstructed images at 50%, 75%, and 90% sparsity, respectively.

Each figure contains reconstructions for all 10 ADE20K images used in our experiments (rows). For each image, we display three columns: the ground-truth target image, the reconstruction produced by RTL, and the reconstruction produced by a single-mask IMP baseline. All models are trained under identical architectural, optimization, and sparsity constraints.

At 50% sparsity (Fig. 12), both methods recover the overall scene structure, but RTL consistently preserves finer details and sharper object boundaries. Differences are particularly noticeable in textured regions and along semantic edges, where IMP reconstructions exhibit mild blurring and loss of contrast.

At 75% sparsity (Fig. 13), the qualitative gap widens. IMP reconstructions show clear degradation in high-frequency content, with washed-out colors and incomplete reconstruction of small structures. In contrast, RTL maintains more faithful geometry and color consistency across most images, indicating that region-specialized subnetworks better preserve semantically important parameters under aggressive pruning.

At the extreme 90% sparsity regime (Fig. 14), the difference becomes pronounced. IMP often collapses to coarse, low-detail approximations, with significant artifacts and loss of semantic coherence. RTL reconstructions, while degraded relative to lower sparsity levels, retain recognizable object shapes, clearer region boundaries, and more stable color distributions. This qualitative evidence aligns with the PSNR trends reported in the main paper, confirming that adaptive, region-specific pruning enables more graceful degradation as sparsity increases.

Overall, these visual results demonstrate that RTL not only improves average reconstruction metrics, but also yields perceptually superior outputs, especially in the high-sparsity regime where competition for shared parameters is most severe.

B.11 Seed sensitivity

To quantify robustness to initialization, we replicate the INR experiment with three random seeds (seed 1 corresponds to the numbers reported in the main text). Table 7 reports PSNR as mean $\pm$ standard deviation across seeds.

Table 7: Seed sensitivity of the INR experiment. PSNR reported as mean $\pm$ std over three random seeds.

Method	**25%**	**50%**	**75%**
RTL	**18.51 $\pm$ 0.34**	**17.17 $\pm$ 0.11**	**14.75 $\pm$ 0.12**
IMP	16.17 $\pm$ 0.29	14.78 $\pm$ 0.12	12.81 $\pm$ 0.12

The pattern is stable across initializations: RTL outperforms IMP for every individual seed at every sparsity level, with per-seed gaps ranging from 1.69 to 2.92 dB, well above the across-seed variability (std $\leq$ 0.34 dB). The specialization and collapse dynamics are likewise qualitatively unchanged across seeds. Within each comparison, all methods share identical initializations, so the reported RTL-vs-IMP gaps are not driven by favorable draws.

B.12 Controlled ablation of the number of tickets K

The per-image variation of K (Table 4) provides an observational sweep, but K covaries there with image content. To isolate the effect of K, we fix a single image (index 3733, 15 semantic regions) and construct coarser partitions of the *same* data by hierarchically merging its segments, evaluating $K \in \{2, 4, 8, 15\}$ under identical training with the same three seeds as Appendix B.11; $K = 1$ corresponds to the IMP baseline.

Merging is deterministic and hierarchical. Each segment is summarized by a 5-dimensional feature vector: its mean color in the normalized image space (3 dimensions) and its centroid with coordinates scaled to $[0, 1]$ (2 dimensions), so that both components contribute at comparable scale. Segments are merged by Ward agglomerative clustering over these features, producing a single merge tree that is cut at $K = 8$, 4, and 2. The resulting partitions are therefore nested (verified explicitly): each coarser level is exactly the previous level with specific merges applied, so differences across K are attributable to the merges themselves rather than to unrelated re-partitionings. No spatial adjacency constraint is imposed, so a merged supercluster may be spatially disconnected (e.g., visually similar regions visible through separate gaps); for coordinate-based reconstruction this is appropriate, as the subtask is defined by pixel content rather than contiguity. Supercluster labels are reassigned by descending pixel count purely for stable output, and the procedure involves no randomness. Table 8 reports PSNR as mean $\pm$ std over the three seeds.

Table 8: Controlled ablation of K on a single ADE20K image (index 3733) with hierarchically merged semantic partitions. PSNR reported as mean $\pm$ std over three seeds; $K = 1$ is the IMP baseline.

Partition granularity	**25%**	**50%**	**75%**
$K = 15$ (original regions)	**10.13 $\pm$ 0.25**	**9.25 $\pm$ 0.24**	**7.83 $\pm$ 0.12**
$K = 8$ (merged)	10.15 $\pm$ 0.27	9.07 $\pm$ 0.30	7.62 $\pm$ 0.12
$K = 4$ (merged)	9.90 $\pm$ 0.19	9.05 $\pm$ 0.16	7.73 $\pm$ 0.12
$K = 2$ (merged)	9.13 $\pm$ 0.32	8.66 $\pm$ 0.16	7.24 $\pm$ 0.09
$K = 1$ (IMP)	8.75 $\pm$ 0.23	8.04 $\pm$ 0.11	6.87 $\pm$ 0.11

With image content, data, training protocol, and seeds all held fixed, performance improves with partition granularity and saturates around $K = 8$–15, where differences fall within seed variability. Every $K \geq 2$ outperforms the $K = 1$ baseline for every individual seed at every sparsity level, with a minimum per-seed gap of 0.17 dB at $K = 2$ and at least 0.63 dB for $K \geq 4$. RTL's advantage is therefore not an artifact of any particular choice of K, and even the coarsest semantic split already yields gains.

B.13 Recovery after mask transfer

To test whether the masks transferred in Sec. 5.4 preserve recoverable functional structure, we fine-tune the masked dense model for 1,000 optimization steps while keeping the masks fixed. As shown in Fig. 11, RTL masks not only start from a substantially better zero-shot initialization, but also recover faster and converge to higher final PSNR than both random and reversed masks. This suggests that RTL-derived masks align more closely with the latent functional organization of the dense model.

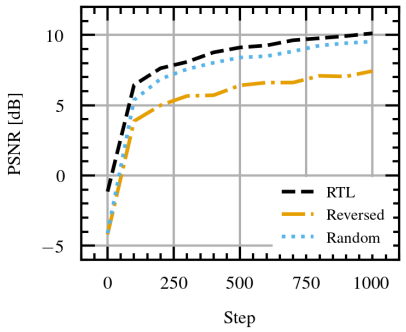

Figure 11: Model recovery after mask transfer. Fine-tuning the unpruned model for 1,000 steps with transferred masks held fixed.

B.14 Selective pathway masking in the dense model

Complementing the zero-shot transfer analysis in Sec. 5.4, we selectively remove individual pathways identified as specialized at $\approx$2% density from the fully trained, unpruned model. Table 9 reports per-object PSNR after removing each pathway in turn.

While overall PSNR decreases in all cases, per-object metrics show that individual masks disproportionately degrade specific objects, indicating that different regions of the model contribute unevenly to different semantic components of the input. For Objects 2–4, the largest degradation coincides with the removal of the object's own pathway (bold diagonal). Object 1 is an exception: its largest degradation arises from removing pathway 3 rather than its own, so the selectivity is relative rather

Table 9: Selective pathway masking in the unpruned model. Bold marks the largest degradation for each object.

Method	Avg	Obj1	Obj2	Obj3	Obj4
Unmasked	14.26	30.36	13.62	16.55	31.08
Masked 1	4.80	11.05	5.87	2.43	8.09
Masked 2	3.77	10.09	**4.26**	2.36	7.84
Masked 3	3.73	**9.97**	4.76	**1.42**	7.20
Masked 4	4.60	10.88	5.30	2.83	**6.72**

than strictly diagonal. This is consistent with the observation in Sec. 5.4 that coherent reconstruction depends on both shared and specialized parameters: objects whose reconstruction leans more heavily on shared structure are degraded most by whichever pathway removal disrupts that shared structure, rather than by the removal of their nominally dedicated pathway.

B.15 Null calibration of structural metrics

To ensure that the structural metrics of Sec. 5.6 reflect genuine organization rather than chance overlap, we compare each observed value against a null model. For every image and pruning iteration, we record the per-layer active counts of each adaptive mask (these differ across layers and masks because pruning is global) and sample null mask sets by placing the same number of active entries uniformly at random within each layer. This preserves the number of tickets K, each mask's size, and its per-layer density profile, while destroying all positional structure. The universal LTH mask is kept fixed, so the null answers whether the real adaptive masks explain the real universal ticket better than chance-matched ones. Null statistics are averaged over 100 random draws. Across all reported metrics, the Monte Carlo standard deviation of the null is below 0.04; note that this quantifies variability of the null estimate only, not variability of the observed statistic, which we assess separately via the paired per-image analysis below. Consistent with the conventions used throughout the paper, all metrics are computed per layer and averaged across layers.

Table 10 reports observed and null values at representative sparsity levels, averaged over the 10 ADE20K images. Three regimes are visible. Below $\approx$80% sparsity, the null saturates for coverage (0.95–1.0) while the core ratio and pairwise Jaccard similarity remain well above chance. Between $\approx$83% and $\approx$99.5% sparsity, every alignment metric exceeds its null by a wide margin.

Table 10: Null calibration of structural metrics on INRs. Cells report observed / null values, averaged over the 10 ADE20K images. Null masks match the adaptive tickets in number and per-layer density.

Metric	**20%**	**79%**	**91%**	**97%**	**99.3%**
Coverage	0.96 / 1.00	0.76 / 0.81	0.70 / 0.57	0.63 / 0.34	0.54 / 0.24
Precision	0.82 / 0.81	0.52 / 0.34	0.44 / 0.23	0.33 / 0.15	0.24 / 0.11
Overlap (norm.)	0.87 / 0.80	0.68 / 0.27	0.58 / 0.18	0.47 / 0.13	0.38 / 0.10
Core ratio	0.71 / 0.35	0.35 / 0.07	0.25 / 0.04	0.19 / 0.02	0.16 / 0.01
Entropy (norm.)	0.53 / 0.60	0.81 / 0.67	0.82 / 0.52	0.76 / 0.34	0.61 / 0.20
Pairwise Jaccard	0.85 / 0.67	0.54 / 0.12	0.40 / 0.05	0.27 / 0.02	0.12 / 0.01



Figure 12: Qualitative INR reconstructions at 50% sparsity. Rows correspond to 10 ADE20K images. Columns show the ground-truth target, RTL reconstruction, and IMP reconstruction. RTL preserves sharper edges and finer details than IMP, particularly in textured and semantically complex regions.



Figure 13: Qualitative INR reconstructions at 75% sparsity. RTL maintains more faithful structure and color consistency, while IMP exhibits increased blurring and greater loss of high-frequency details. The gap between methods becomes more visible as sparsity increases.



Figure 14: Qualitative INR reconstructions at 90% sparsity. Under extreme pruning, IMP reconstructions often collapse to coarse approximations with severe artifacts. RTL degrades more gracefully, preserving recognizable object shapes and semantic boundaries across images.

C Vision Model and Training Setup

C.1 Model architecture and pruning scope

For all computer vision experiments on CIFAR-10 and CIFAR-100, we use GhostNet Han et al. (2019) as the backbone. To control model capacity and isolate the effects of adaptive pruning, we

retain the convolutional stem followed by the first nine Ghost bottleneck blocks, discarding the deeper stages of the original network.

All convolutional layers are implemented as masked convolutions and constitute the pruning scope. Batch normalization layers, squeeze-and-excitation modules, and the final classifier remain dense. Under this configuration, the model contains approximately 126K prunable parameters.

We compare RTL against two baselines: (i) *IMP (single model)*, which learns a single shared subnetwork across all classes, and (ii) *IMP (multiple models)*, which independently prunes one model per class without weight sharing.

The output head differs by method: RTL and multi-model IMP subnetworks are one-vs-rest binary classifiers with a single sigmoid output ($C = 1$), consistent with the voting evaluation protocol (Appendix C.3), whereas the single-model IMP baseline uses a standard multiclass head ($C = 10$ for CIFAR-10, $C = 100$ for CIFAR-100).

The retained GhostNet variant follows the standard Ghost bottleneck design, combining inexpensive depthwise convolutions with pointwise expansions and optional downsampling and squeeze-and-excitation. Spatial resolution is progressively reduced via strided depthwise convolutions in selected bottlenecks. A detailed summary of the architecture, including channel dimensions and downsampling stages, is provided in Table 11 for reproducibility.

Table 11: GhostNet backbone used for CIFAR experiments. Only convolutional layers are pruned. All convolutions are followed by batch normalization and ReLU unless stated otherwise.

Stage	**Block type**	**Input → Output**	**Kernel / Stride**	**Notes**
Stem	Conv2D	$3 \to 16$	3×3 / 2	Initial downsampling
1	GhostBottleneck	$16 \to 16$	$1 \times 1, 3 \times 3$ / 1	No shortcut
2	GhostBottleneck	$16 \to 24$	3×3 / 2	Downsampling + shortcut
3	GhostBottleneck	$24 \to 24$	$1 \times 1, 3 \times 3$ / 1	Shortcut
4	GhostBottleneck	$24 \to 40$	5×5 / 2	SE + shortcut
5	GhostBottleneck	$40 \to 40$	$1 \times 1, 3 \times 3$ / 1	SE
6	GhostBottleneck	$40 \to 80$	3×3 / 2	Downsampling + shortcut
7	GhostBottleneck $\times 3$	$80 \to 80$	$1 \times 1, 3 \times 3$ / 1	Repeated blocks
8	Conv1×1	$80 \to 184$	1×1 / 1	Channel expansion
Head	Conv + Pool	$184 \to 128$	1×1	Global average pooling
Classifier	Linear	$128 \to C$	–	See note below

C.2 Optimization and training protocol

All models are trained using the Adam optimizer Kingma & Ba (2015) with a learning rate of $1e{-4}$ and no weight decay. Each pruning iteration consists of two phases, (i) pruning and (ii) joint retraining, both of which run for 10 epochs. Batch sizes are set to 320 for CIFAR-10 and 256 for CIFAR-100.

To ensure a fair comparison, all methods (RTL, single-model IMP, and multi-model IMP) share the same random Kaiming initialization He et al. (2015). RTL and multi-model IMP subnetworks are trained as binary classifiers with balanced mini-batches, effectively doubling the number of sample presentations per epoch. To match total data exposure, the single-model IMP baseline is trained for 20 epochs.

C.3 Evaluation protocol

Evaluation runs after every joint retraining phase on the full, unbalanced CIFAR test set (10,000 images), where an image is a positive for cluster c if its class maps to c. We record balanced accuracy

(primary metric, due to class imbalance under the one-vs-rest setup), accuracy, F1, precision, and recall, along with the current sparsity and the mean pairwise Jaccard similarity between cluster masks.

We considered two evaluation protocols. In the *per-cluster binary* protocol, each cluster's mask is applied, sigmoid scores are thresholded at 0.5, and per-subnetwork metrics are computed against the cluster's positives; this protocol assumes the partition label at test time. In the *voting* protocol, every test image is passed through all K subnetworks, and the predicted cluster is the argmax over the K sigmoid scores; no labels are used at test time. All final scores reported in the paper use the voting protocol. The per-cluster binary protocol yields higher scores, but we consider it artificial in a classification setting, since one must already name a class to ask whether a sample belongs to it; the label-free voting numbers are therefore the more practical and conservative comparison against IMP.

C.4 Pruning schedule and compute

For the experiments in Sec. 5.1, pruning follows a fixed schedule, globally removing the 4,096 lowest-magnitude weights (ranked across all prunable layers jointly) per pruning step for each subnetwork until target sparsity levels of 25%, 50%, and 75% are reached. For all other analyses, pruning proceeds iteratively by globally removing 20% of the currently active weights at each pruning step, enabling finer resolution near extreme sparsity and more precise pathway estimation.

All experiments are executed on a single NVIDIA H100 GPU. Full RTL and multi-model IMP runs require approximately 6 hours, whereas the single-model IMP baseline completes in roughly 45 minutes.

C.5 CIFAR-10 subnetwork specialization

Table 12 summarizes CIFAR-10 results under class-specific pruning. RTL consistently achieves the highest balanced accuracy across all sparsity levels. At 25% sparsity, RTL attains a balanced accuracy of 0.781, outperforming both baselines (0.758 for single-model IMP and 0.712 for multi-model IMP). This advantage persists at 50% sparsity (0.778 vs. 0.742) and remains competitive even at 75% sparsity (0.772 vs. 0.760 for multi-model IMP), despite using an order of magnitude fewer parameters.

RTL also achieves the highest precision at all sparsity levels (0.282 / 0.276 / 0.257), exceeding the single-model baseline and matching or surpassing the multi-model alternative. This indicates that RTL subnetworks retain stronger class selectivity under pruning, reducing false positives despite increased sparsity.

Critically, these gains come with exceptional parameter efficiency. RTL uses only 103K / 71K / 38K parameters at 25% / 50% / 75% sparsity, compared with 944K / 629K / 314K for multi-model IMP, nearly 10$\times$ more, to achieve comparable accuracy. This underscores RTL's ability to deliver high performance without sacrificing compactness.

C.6 Cluster-aware pruning on CIFAR-100

Table 12 also reports CIFAR-100 results, where subnetworks are specialized to semantically derived clusters. Despite the inherent ambiguity in this partitioning, RTL again achieves the highest balanced accuracy across all sparsity levels: 0.765 (25%), 0.751 (50%), and 0.759 (75%). Notably, RTL outperforms both baselines even at 75% sparsity, where model capacity is most constrained, demonstrating robustness to imperfect data grouping.

In terms of efficiency, RTL uses only 107K / 75K / 38K parameters, comparable to the single-model baseline and drastically fewer than the multi-model baseline (944K / 629K / 314K). This reaffirms that RTL achieves strong specialization without requiring redundant, over-parameterized subnetworks, making it especially suitable for scenarios with limited memory or compute.

C.7 Partition-noise control on CIFAR-100

Complementing the INR partition-noise control in Sec. 5.2, we evaluate RTL on CIFAR-100 under class-level partition noise: for noise level $X\%$, $X\%$ of the 100 classes are reassigned to different clusters, so 100% noise corresponds to semantically random cluster membership. Table 13 reports

Table 12: Results on CIFAR-10 with class-specific subnetworks and on CIFAR-100 with cluster-specialized subnetworks. We report balanced accuracy, precision, and the number of remaining parameters at three sparsity levels. RTL is compared against (a) IMP with a single shared subnetwork and (b) IMP trained independently per class (multiple models). Best values for each metric and sparsity level are bolded. Clusters are derived from semantic embeddings (see Section 4).

	Method	Balanced accuracy ↑			Precision ↑			#Params ↓		
		25%	50%	75%	25%	50%	75%	25%	50%	75%
CIFAR-10	RTL (ours)	**0.781**	**0.778**	**0.772**	**0.282**	**0.276**	0.257	103K	71K	38K
	IMP (single model)	0.758	0.742	0.707	0.200	0.198	0.187	**94K**	**63K**	**32K**
	IMP (multiple models)	0.712	0.710	0.760	0.233	0.222	**0.262**	944K	629K	314K
CIFAR-100	RTL (ours)	**0.765**	**0.751**	**0.759**	**0.298**	**0.290**	**0.289**	107K	75K	38K
	IMP (single model)	0.716	0.712	0.705	0.208	0.204	0.204	**94K**	**63K**	**32K**
	IMP (multiple models)	0.712	0.700	0.744	0.276	0.271	0.286	944K	629K	314K

balanced accuracy under the label-free voting protocol, with the single-model IMP baseline shown for reference.

Table 13: Class-level partition-noise control on CIFAR-100. Balanced accuracy under the label-free voting protocol.

Partition	25%	50%	75%
RTL (semantic clusters)	**0.765**	**0.751**	**0.759**
RTL (noise 50%)	0.642	0.611	0.589
RTL (noise 100%, random)	0.580	0.573	0.534
IMP (single model)	0.716	0.712	0.705

The pattern mirrors, and in fact strengthens, the INR result. With fully random class-to-cluster assignments, RTL falls far below the universal-pruning baseline at every sparsity level (e.g., 0.580 vs. 0.716 at 25% sparsity), and degradation is monotonic in the noise level. When partitions do not align with semantic structure, forcing subnetworks to specialize is actively harmful, confirming that RTL's gains stem from partition-aligned structure rather than from partitioning or added capacity per se.

C.8 Class-level semantic alignment trends

Complementing the aggregate semantic-alignment results in Sec. 5, per-class trends (Figure 4C–D in the main text) reveal substantial variation. Semantically coherent classes such as *cat* and *deer* maintain consistently high correlations, increasing from approximately 0.48/0.43 to 0.65/0.67 with pruning. In contrast, *truck* declines monotonically from 0.23 to 0.12, while *airplane* shows a U-shaped pattern, suggesting partial reorganization prior to collapse. We note that each per-class correlation is computed from only nine class-pair comparisons, so these trends should be read as exploratory rather than as statistically established effects.

C.9 Collapse dynamics across classification tasks

To study the failure modes of adaptive tickets under increasing sparsity, we track both balanced accuracy and average pairwise mask similarity across pruning levels. We define *subnetwork collapse* as the regime in which excessive pruning forces previously distinct adaptive subnetworks to converge toward overlapping parameter sets, reducing structural specialization.

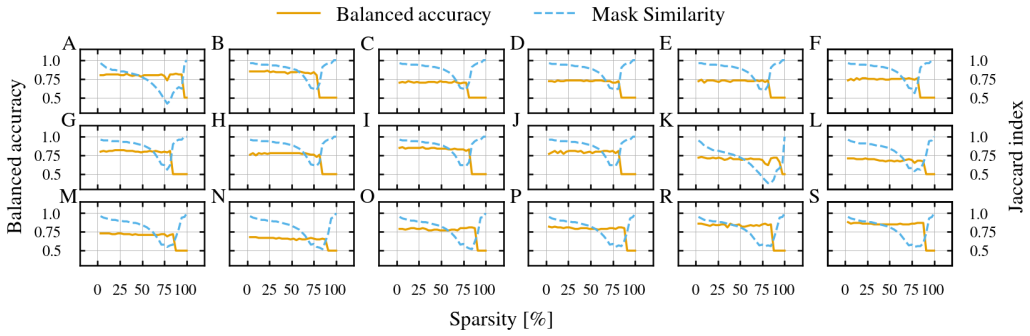

Figure 15: Mask collapse analysis on CIFAR-10 and CIFAR-100. Each subplot shows balanced accuracy (solid line) and average mask similarity to other subnetworks (dashed line) across pruning levels. Plots A–J correspond to CIFAR-10 and K–S to CIFAR-100.

Figure 15 shows that across both CIFAR-10 and CIFAR-100, performance remains stable while subnetworks preserve low-to-moderate similarity. However, beyond a dataset-dependent sparsity threshold, mask similarity rises sharply and coincides with a sudden degradation in balanced accuracy. This transition indicates that RTL performance depends not only on retaining sufficient parameters, but also on preserving structural separation between data-dependent pathways.

In CIFAR-10 (Fig. 15A–J), where subnetworks are extracted per class, most pathways maintain strong performance until approximately 70–80% sparsity. Beyond this point, adaptive masks rapidly converge, accompanied by an abrupt accuracy drop. The relatively late collapse suggests that, when classes are well separated, subnetworks can remain distinct under substantial pruning before being forced into overlap.

A similar pattern is observed on CIFAR-100 (Fig. 15K–S), where subnetworks are learned over semantically derived clusters. Here, performance remains stable while similarity is bounded, but deteriorates once pruning induces excessive overlap between masks. Compared with CIFAR-10, collapse typically occurs at slightly higher sparsity, consistent with greater feature sharing between semantically related clusters.

Across both datasets, mask similarity consistently increases prior to or alongside performance degradation, suggesting that structural overlap provides a label-free indicator of impending collapse. These results highlight that successful adaptive pruning depends on maintaining a balance between parameter sharing and specialization: once sparsity forces excessive overlap between adaptive subnetworks, the pathway structure uncovered by RTL collapses toward a degenerate shared solution, resulting in catastrophic loss of task performance.

D Semantic Clustering Procedure

To construct coarse semantic subsets for CIFAR-100, we apply an unsupervised clustering pipeline based on textual class descriptions rather than visual features.

First, we extract text embeddings for each of the 100 CIFAR-100 class names using the CLIP text encoder. These embeddings are then reduced in dimensionality using UMAP to facilitate clustering in a lower-dimensional space. Finally, we apply HDBSCAN to the reduced embeddings to obtain cluster assignments.

This procedure yields eight semantic clusters. While the resulting clusters are semantically coherent, they are not perfectly aligned with visual similarity, intentionally introducing noise and overlap. This imperfect alignment better reflects real-world scenarios in which data partitions are ambiguous or only approximately defined. The full cluster composition is reported in Table 14.

All clustering hyperparameters are fixed across experiments and are not tuned to downstream task performance. As shown in Sec. 5, RTL remains effective under this setting, demonstrating robustness to non-ideal specialization boundaries and highlighting the benefits of adaptive pruning beyond strictly class-aligned scenarios.

Table 14: Semantic clusters used in the CIFAR-100 experiments. Each cluster defines a data subset to which a specialized subnetwork is assigned.

Cluster	CIFAR-100 Classes
1 (Aquatic animals)	aquarium fish, dinosaur, dolphin, flatfish, ray, seal, shark, trout, whale
2 (People & objects)	baby, bed, bicycle, bottle, bowl, boy, bridge, can, castle, chair, clock, couch, cup, girl, house, keyboard, lamp, man, plain, plate, road, rocket, skyscraper, table, telephone, television, wardrobe, woman
3 (Mammals)	bear, beaver, camel, cattle, chimpanzee, elephant, fox, hamster, kangaroo, leopard, lion, mouse, otter, porcupine, possum, rabbit, raccoon, shrew, skunk, squirrel, tiger, wolf
4 (Reptiles & insects)	bee, beetle, butterfly, caterpillar, cockroach, crab, crocodile, lizard, lobster, mushroom, snail, snake, spider, turtle, worm
5 (Vehicles)	bus, lawn mower, motorcycle, pickup truck, streetcar, tank, tractor, train
6 (Natural scenes)	cloud, forest, maple tree, mountain, oak tree, palm tree, pine tree, sea, willow tree
7 (Fruits)	apple, orange, pear, sweet pepper
8 (Flowers)	orchid, poppy, rose, sunflower, tulip

E SPEECH ENHANCEMENT

E.1 DATASET CONSTRUCTION

We use clean speech samples from the DNS Challenge 2020 dataset Reddy et al. (2020) and mix them with environmental noise from the TAU Urban Acoustic Scenes 2020 dataset Heittola et al. (2020). Noise samples are grouped into three acoustic scenes: *indoor*, *outdoor*, and *transportation*. Each scene defines a distinct subset used to train a specialized subnetwork.

E.2 MODEL ARCHITECTURE

For the speech enhancement task, we use a lightweight U-Net–style architecture Ronneberger et al. (2015) operating on complex STFT representations. The network takes a 2-channel input (real and imaginary components) and predicts a 2-channel complex ratio mask.

The model consists of five encoder stages and five decoder stages. Each encoder stage applies a masked 2D convolution followed by ELU activation and batch normalization. Downsampling is performed only along the time axis using a stride of $(1, 2)$, preserving frequency resolution. The encoder channel dimensions are $[12, 12, 12, 24, 48]$. The decoder mirrors the encoder using masked transposed convolutions with symmetric kernel sizes and strides. Skip connections concatenate encoder features with decoder inputs, doubling the channel dimensionality prior to decoding. The final decoder layer outputs a 2-channel complex mask.

All convolutions use 3×3 kernels, except for the first and last layers, which use 3×5 kernels to capture a wider temporal context. This compact design intentionally limits model capacity, allowing improvements from adaptive pruning to be isolated from architectural scaling effects. A detailed layer-by-layer specification of the network is provided in Table 15 for reproducibility.

E.3 SIGNAL PROCESSING AND LOSS

Input audio consists of 10-second, 16-bit waveforms. We compute STFTs using a 1024-sample window and a 256-sample hop size, producing spectrograms of shape $[2, 626, 513]$. The model predicts a mask of identical shape, which is applied to the noisy spectrogram.

Training uses the weighted source-to-distortion ratio (wSDR) loss. Unlike the vision experiments, this task does not involve negative samples; consequently, all methods are trained for the same number of epochs.

Table 15: Speech enhancement U-Net architecture. All layers use ELU activation and batch normalization. Stride $(1, 2)$ downsamples only along the time axis.

Stage	Type	Channels (in $\rightarrow$ out)	Kernel / Stride
Enc-1	Conv2D	$2 \rightarrow 12$	3×5 / $(1, 2)$
Enc-2	Conv2D	$12 \rightarrow 12$	3×3 / $(1, 2)$
Enc-3	Conv2D	$12 \rightarrow 12$	3×3 / $(1, 2)$
Enc-4	Conv2D	$12 \rightarrow 24$	3×3 / $(1, 2)$
Enc-5	Conv2D	$24 \rightarrow 48$	3×3 / $(1, 2)$
Dec-1	TConv2D	$48 \rightarrow 24$	3×3 / $(1, 2)$
Dec-2	TConv2D	$48 \rightarrow 12$	3×3 / $(1, 2)$
Dec-3	TConv2D	$24 \rightarrow 12$	3×3 / $(1, 2)$
Dec-4	TConv2D	$24 \rightarrow 12$	3×3 / $(1, 2)$
Dec-5	TConv2D	$24 \rightarrow 2$	3×5 / $(1, 2)$

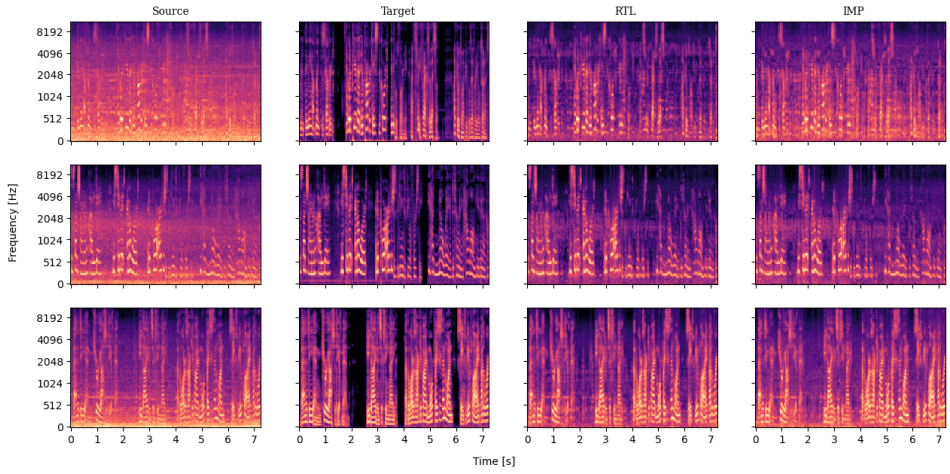

Figure 16: Qualitative speech enhancement results at 50% sparsity. Each row corresponds to a different acoustic environment, and the columns show the noisy input, clean target, RTL output, and IMP output. RTL more effectively suppresses noise and preserves harmonic speech structure across all environments.

E.4 Optimization and runtime

We use the Adam optimizer with a learning rate of $1\mathrm{e}{-4}$ and no weight decay. Both pruning and retraining phases run for 10 epochs. All models share identical initial weights. Experiments are conducted on a single NVIDIA H100 GPU. Full RTL and multi-model IMP runs take approximately 10 hours, while the single-model IMP baseline completes in about 8 hours.

E.5 Qualitative Analysis of Speech Enhancement

Fig. 16 provides a qualitative comparison of speech enhancement results across three acoustic environments at 50% weight sparsity. Each row corresponds to a distinct environment class, while the columns show the noisy input spectrogram, the clean target, and the enhanced outputs produced by RTL and the IMP baseline.

Across all environments, RTL more faithfully reconstructs the time-frequency structure of clean speech. Harmonic components are clearer, transient events are better preserved, and noise-dominated regions are more effectively suppressed. In contrast, the IMP baseline exhibits residual noise,

Table 16: Speech enhancement results on the DNS Challenge 2020 and TAU Urban Acoustic Scenes 2020 datasets. Methods are evaluated at 25%, 50%, and 75% sparsity using SI-SNRi, together with the number of trainable parameters.

Method	**SI-SNRi ↑**			**#Params ↓**		
	25%	50%	75%	25%	50%	75%
RTL (ours)	**7.248**	**7.178**	**6.992**	30.2K	20.9K	11.6K
IMP (single model)	6.885	6.97	6.967	**28.0K**	**18.7K**	**9.3K**
IMP (multiple model)	5.295	5.775	5.899	84.1K	56.1K	28.0K

smeared harmonics, and reduced contrast between speech and background, particularly in the mid- and high-frequency bands.

The qualitative gap becomes especially apparent in challenging conditions, where environment-specific noise patterns dominate the input. RTL subnetworks, specialized to each acoustic scene, recover speech structure with higher temporal coherence and sharper spectral detail, whereas the single-mask IMP model struggles to balance denoising across heterogeneous conditions.

These visual results are consistent with the quantitative SI-SNRi improvements reported in Sec. 5 and further support the claim that adaptive, environment-aligned pruning yields more effective representations than a single global sparse model.

E.6 Speech Enhancement in realistic environments

Table 16 reports speech enhancement performance across three acoustic environments: indoor, outdoor, and transportation. RTL achieves the highest SI-SNRi at all sparsity levels, reaching 7.248, 7.178, and 6.992 at 25%, 50%, and 75% sparsity, respectively, and consistently outperforming both IMP baselines. This suggests that environment-specific subnetworks better capture the distinct spectro-temporal characteristics of each noise type, leading to better waveform reconstruction.

The single-mask IMP model performs reasonably well, likely because it learns a general-purpose denoising strategy. However, it still trails RTL, indicating that specialization yields tangible gains even in a non-classification setting. In contrast, the multi-model IMP baseline underperforms despite its larger parameter budget (84.1K vs. 32.0K at 25% sparsity), suggesting that independent pruning without shared initialization or joint retraining is less effective.

RTL achieves these gains with only modest model-size overhead: 30.2K / 20.9K / 11.6K parameters, only slightly larger than the single-mask model and less than half the size of the multi-model alternative. Notably, even the most aggressively pruned RTL subnetwork (12.3K parameters) surpasses both baselines in SI-SNRi, underscoring the efficiency of the approach.

These results show that RTL generalizes beyond controlled vision benchmarks: when subnetworks align with meaningful real-world structure, here acoustic environments, they deliver better performance while remaining compact and deployable.

F Cost Accounting

This section consolidates the parameter, inference, and training cost accounting for RTL and the IMP baseline across all four evaluation domains.

F.1 Parameter terminology

Throughout the paper, we distinguish the following parameter counts. *Dense* parameters denote all parameters of the network. *Prunable* parameters denote the subset within the pruning scope (e.g., all convolutional weights for the CIFAR models; batch normalization, squeeze-and-excitation, and classifier weights remain dense and are excluded). *Per-mask active* parameters denote the nonzero

entries of a single mask m_k; at mask-level sparsity s, each subnetwork has $(1 - s)$ of the prunable parameters active. *Union active* parameters denote the nonzero entries of $\bigvee_k m_k$, i.e., parameters retained by at least one adaptive ticket; this is the storage-relevant count for RTL, since pruned-everywhere parameters need not be stored. Reported parameter counts for RTL in the task tables refer to union active parameters rounded to the nearest 1K; Table 17 below reports the same quantities at higher precision, which accounts for small apparent discrepancies (e.g., 94K vs. 94.3K).

F.2 Total active parameters

Because RTL maintains K masks over a shared parameter tensor, mask-level sparsity percentages and total active-parameter counts must be distinguished. Table 17 reports the total number of active parameters, i.e., the size of the union of all adaptive masks, together with the corresponding effective (union) sparsity, alongside the IMP baseline at matched mask-level sparsity.

Table 17: Total active parameters and effective sparsity of the union of adaptive masks, compared with the IMP baseline at matched mask-level sparsity.

Domain	Quantity	25%	50%	75%
INR	IMP active weights	40.4K	26.9K	13.5K
	RTL union of masks	47.9K	35.4K	20.5K
	RTL effective (union) sparsity	11.2%	34.3%	61.9%
Speech enh.	IMP active weights	28.0K	18.7K	9.3K
	RTL union of masks	30.2K	20.9K	11.6K
	RTL effective (union) sparsity	19.2%	44.0%	68.9%
CIFAR-10	IMP active weights	94.3K	62.9K	31.5K
	RTL union of masks	103.2K	71.2K	38.2K
	RTL effective (union) sparsity	18.0%	43.5%	69.7%
CIFAR-100	IMP active weights	94.3K	62.9K	31.5K
	RTL union of masks	106.6K	74.5K	38.3K
	RTL effective (union) sparsity	15.3%	40.8%	69.6%

These numbers permit a matched-budget comparison by offsetting sparsity levels. Comparing RTL at 50% against IMP at 25%, the RTL union is smaller in every domain (INR: 35.4K vs. 40.4K, speech enhancement: 20.9K vs. 28.0K, CIFAR-10: 71.2K vs. 94.3K, CIFAR-100: 74.5K vs. 94.3K), yet RTL outperforms IMP in every case (17.25 vs. 15.94 PSNR, 7.178 vs. 6.885 SI-SNRi, 0.778 vs. 0.758 and 0.751 vs. 0.716 balanced accuracy). The same holds at the 75%-vs-50% offset. Across all four domains, RTL outperforms IMP while storing fewer total active parameters. Independently, the partition-noise control (Sec. 5.2) holds the number of masks and per-mask capacity fixed while destroying partition structure, and the gains vanish, indicating that the improvements are not explained by per-mask capacity alone.

F.3 Inference cost

Each input uses exactly one adaptive ticket. For INRs and speech enhancement, inference cost is therefore equal to the IMP baseline at the same mask-level sparsity, disregarding the cost of the routing decision itself: each input activates exactly one subnetwork, at the same sparsity as the corresponding IMP mask, so per-input active parameters and FLOPs match the baseline. The exception is CIFAR under the label-free voting protocol (Appendix C.3), where all K subnetworks are evaluated per input, requiring K sparse forward passes.

F.4 TRAINING COST

We distinguish two notions of training cost, which differ and should not be conflated. (i) *Data exposure of the evaluated network*: matched across methods by construction, since joint retraining occurs after rewinding to initialization, so the evaluated network does not see more data than the baseline. (ii) *Total training compute*: roughly twice that of IMP. The mask-extraction phase costs approximately the same as IMP, since each subnetwork is trained only on its own data subset, so one extraction iteration over all K subnetworks processes the dataset approximately once. CIFAR is the exception: binary classifiers are trained with positive and negative samples, doubling data exposure, and we correspondingly doubled the IMP baseline's training budget to keep the comparison fair (Appendix C). The overhead comes from the joint-retraining phase, a second phase needed only to produce an evaluable network at a given sparsity; it can be skipped entirely when no evaluation at a given sparsity is required.