CHANNEL-REMIX: EFFICIENT TASK-SPECIFIC KNOWLEDGE TRANSFER VIA PRETRAINED CHANNEL REMIXING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale pretraining endows vision foundation models (VFMs) with rich, transferable visual knowledge, yet their compute and memory costs put much of it out of reach in practice. Knowledge distillation compresses this knowledge by transferring it to a compact student under the VFM's supervision. However, while such supervision provides a strong training signal, it leaves the student's parameter space unconstrained. Consequently, bridging the gap between teacher representations and viable student weights typically demands substantial task data and long training. We propose *Channel-Remix*, which uses the VFM not only as a teacher but as a library of pretrained weight directions from which the student is built. This recasts knowledge transfer as selecting and recombining channels that pretraining has already learned, a problem that requires far less task data than learning new weights. Given a parameter budget and a small calibration set, *Channel-Remix* retains a core subset of channels and sparsely mixes pruned ones into them to form new task-adapted channels with zero runtime cost. By grounding both the learning signal and the student's parameter space in the VFM, *Channel-Remix* achieves high data efficiency, matching LoRA-based distillation using 3–4$\times$ less calibration data. Across 16 tasks, 7 pretraining families, and 5 model sizes, it improves pruned models by 4.6% on average and closes the gap to the original VFM to under 2% at 50% compression, outperforming distillation baselines under full fine-tuning, PEFT, and post-pruning reconstruction while training fewer parameters. Moreover, because every student is built from the same pool of VFM directions, models for different tasks become directly comparable, opening a route to analyzing which knowledge is shared across tasks and which is task-specific.

1 INTRODUCTION

Vision foundation models (VFMs) are pretrained on massive data at substantial cost, under the premise that the resulting knowledge can be reused across many applications. Recent large VFMs have borne out this premise, delivering strong performance on a wide range of tasks (Siméoni et al., 2025). However, deploying these models means inheriting their expensive inference requirements. For applications that cannot accommodate their memory, latency, or energy demands, much of the value of this pretraining remains inaccessible unless it can be transferred to smaller models.

Knowledge distillation transfers this knowledge into a compact student by using the VFM as a teacher (Hinton et al., 2015). Richer objectives strengthen the training signal by incorporating both the teacher's predictions and its intermediate representations (Romero et al., 2015; Zhao et al., 2022). Yet a stronger signal does not eliminate the cost of teacher-guided optimization, which is expensive and may need to be repeated for each new task or deployment budget. Nor does it reduce the dependence on data: optimizing all student parameters can still require substantial task data and risks overfitting when such data is scarce (Liang et al., 2021; Ko et al., 2023).

To lower this optimization burden, modern pipelines pair compression with parameter-efficient fine-tuning (PEFT). Compression methods, such as structured pruning or low-rank factorization, yield compact models, typically followed by reconstruction to recover lost accuracy (Frantar & Alistarh, 2022; Kim et al., 2020). PEFT methods, such as LoRA or adapters, restrict task adaptation to a small

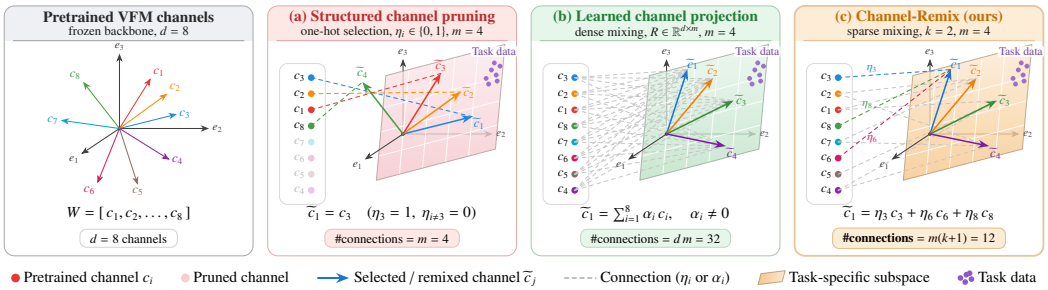


Figure 1: **Overview of ways to build a compact student from VFM channels.** (a) Structured pruning keeps a subset of channels. (b) Dense projection mixes all d channels into each compact channel. (c) *Channel-Remix* mixes only $k{+}1$, fitting the task subspace with fewer connections.

set of parameters (Chen et al., 2022; Hu et al., 2022; Lian et al., 2022). However, learning fewer parameters does not inherently make adaptation data-efficient. Because PEFT still optimizes weight updates in arbitrary, unconstrained directions, estimating even low-rank matrices from scarce calibration data remains prone to overfitting (Ruan et al., 2025; Shin et al., 2024; Haruta et al., 2026).

Therefore, the path to data efficient knowldge transfer requires more than learning fewer parameters. Low-rank adapters constrain the rank of an update but still learn its directions from the adaptation data, indeed, increasing LoRA's rank in limited-data settings can improve training fit while impairing generalization (Mai et al., 2025; Borse et al., 2024). The hard part is learning useful channels from scratch, however few they may be. Could this burden be eased by giving the student pretrained channels to draw from? A VFM provides exactly such a foundation: its channels already encode features shaped in pretraining. We therefore ask whether the student's weights can be built from these channels directly. Doing so, free the downstream data from synthesizing new channels, and is spent only on selecting and recombining channels already established in pretraining.

Structured channel pruning represents the most rigid form of this idea: every compact channel is strictly selected from the original pretrained channels (Li et al., 2017), confining the student to a coordinate-aligned subspace of the teacher as shown in Figure 1 (a). While highly data-efficient, this hard constraint sacrifices expressiveness whenever optimal task directions do not align with isolated channels. Learned weight projections relax this restriction by synthesizing compact channels as linear combinations of the teacher's channels (Zhmoginov et al., 2024). As depicted in Figure 1 (b), dense projections allow compact directions to draw upon the full span of the source weights. Yet, allowing every compact channel to depend on every original channel introduces an abundance of task-specific mixing coefficients, reintroducing the very optimization cost and overfitting risks that limited data cannot support.

To resolve this trade-off, we propose *Channel-Remix*. Rather than optimizing unconstrained weight updates or estimating dense linear projections, *Channel-Remix* constructs each compact model's channel by using a retained channel as an anchor direction and refining it with a sparse blend of discarded pretrained channels as shown in Figure 1 (c). A small calibration set jointly determines which discarded channels contribute and how strongly they contribute. This provides the flexibility to synthesize new channels beyond the original ones, while keeping the number of learned coefficients small enough to be estimated reliably from limited data. Once learned, the sparse mixtures are folded into the anchor weights, producing a compact model with no additional cost at inference.

Our key contributions are:

- **A foundational role for VFMs beyond supervision.** Rather than using the VFM only as a teacher, we build the student from its pretrained channels. This recasts knowledge transfer from constructing new weights into selecting and recombining directions learned during pretraining.
- **Task-specific subspace discovery via sparse recombination.** *Channel-Remix* creates new task-specific channels by sparsely blending the pretrained VFM channels, guided by a small calibration set. The blending coefficients fold directly into standard weights, adding zero inference overhead.
- **Comprehensive evaluation across tasks, models, calibration sizes.** Across 16 tasks and 7 pretraining schemes, *Channel-Remix* consistently improves compressed models by 4.6% on average and closes the performance gap to the uncompressed model to under 2% at 50% compression. Moreover, *Channel-Remix* demonstrates strong data efficiency, matching the performance of LoRA-based distillation while using only one-third to one-quarter of the calibration samples.

2 Related Work

Knowledge Distillation. Knowledge distillation trains a compact student under the supervision of a larger teacher (Hinton et al., 2015; Touvron et al., 2021; Zhao et al., 2022), often by matching intermediate features or attention maps (Romero et al., 2015; Zhang et al., 2022). However, teacher supervision does not remove the data appetite of student training: reproducing the teacher's capabilities still requires substantial task data and long optimization.

Parameter-Efficient Fine-Tuning (PEFT). PEFT reduces training cost by updating a small set of parameters, such as adapters (Chen et al., 2022), modulation vectors (Lian et al., 2022), low-rank updates (Hu et al., 2022; Liu et al., 2024), or singular vectors (Meng et al., 2024; Lingam et al., 2024). Yet fewer parameters do not by themselves constrain the update directions, so with scarce data PEFT remains prone to overfitting (Shin et al., 2024; Haruta et al., 2026). This suggests that data efficiency requires constraining the update space, not only its size.

Structured Pruning and Channel Compression. Structured pruning removes channels, heads, or MLP dimensions for hardware-friendly speedups (Li et al., 2017; Fang et al., 2023), often followed by post-pruning compensation (Frantar & Alistarh, 2022). To retain some of the removed capacity, Neuron Merging (Kim et al., 2020) merges similar neurons, and Projectable Models (Zhmoginov et al., 2024) learn dense projections into compact channels. However, pruning discards non-selected directions entirely, Neuron Merging relies on task-agnostic similarity, and dense projections introduce too many coefficients for small calibration sets.

Weight-Space Manipulation. Model soups (Wortsman et al., 2022) and task arithmetic (Ilharco et al., 2023) operate directly in weight space, merging or editing the weights of separately fine-tuned models without additional training. However, the resulting model remains the same size as the original. How to recombine directions within a single VFM to simultaneously compress and adapt it remains largely unexplored.

3 Methodology

Transferring knowledge from a vision foundation model (VFM) to a compact smaller model is challenging when task data is scarce. As established in §2, existing methods supply stronger supervision or fewer parameters, but leave update directions unconstrained, making optimization prone to overfitting. *Channel-Remix* addresses this by letting the VFM shape both the learning signal and the parameter space: compact weights are constructed directly from its pretrained channels. Below, we formalize this objective and describe our approach.

3.1 Problem Formulation and Approach

Let F_T denote a pretrained VFM with frozen parameters θ_T, B a parameter budget, and $\mathcal{D}_c = \{(x_n, y_n)\}_{n=1}^{N}$ a small task calibration set. Given a compact architecture F_S satisfying the parameter budget B, our goal is to find student weights θ_S that generalize from limited data:

$$\min_{\theta_S} \mathcal{L}_{\text{task}}(F_S(\theta_S);\, \mathcal{D}_c) + \mathcal{L}_{\text{transfer}}(F_S(\theta_S),\, F_T(\theta_T);\, \mathcal{D}_c) \quad \text{subject to} \quad \text{size}(F_S) \leq B, \quad (1)$$

where $\mathcal{L}_{\text{task}}$ is the downstream task loss and $\mathcal{L}_{\text{transfer}}$ measures agreement with the VFM's outputs and representations. Knowledge distillation addresses this problem by using the VFM as a teacher: it defines $\mathcal{L}_{\text{transfer}}$ to match the teacher's predictions or intermediate features and optimizes all of θ_S freely. When $|\mathcal{D}_c|$ is small, however, this unconstrained optimization must estimate far more parameters than the calibration data can reliably determine.

Our work, *Channel-Remix*, addresses this by additionally constraining the compact model parameters θ_S to a structured subspace derived from the VFM's parameters θ_T: each channel from θ_S is constructed as a sparse combination of pretrained channel weights from θ_T. This reduces the effective dimensionality of the parameter space, biasing it toward directions that large-scale pretraining has already learned, while also reducing the number of mixing coefficients that need to be learned due to sparse combinations. Through this formulation, the VFM shapes both the learning signal and the space of parameters, allowing limited calibration data to identify effective compact weights.

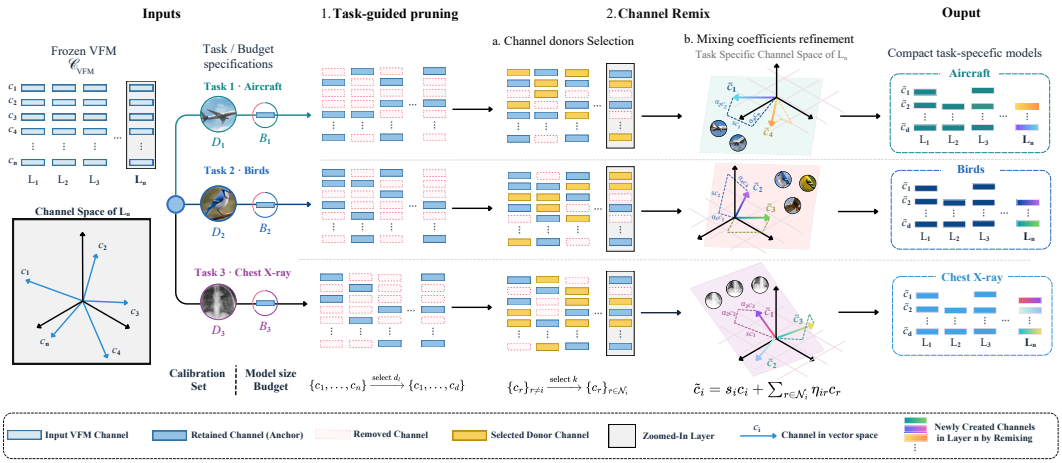


Figure 2: **Method overview.** Starting from the VFM's channel pool (left), pruning keeps a budget of anchor channels (blue). **Channel Remixing (center)** (a) selects donors (yellow) from the discarded channels, then (b) mixes each anchor with K donors to form new task-adapted directions. Shown for layer L_n only; all layers are remixed jointly. The new channels (mixed colors) fold into the model weights, yielding compact task-specific models with zero inference overhead as outputs (right).

3.2 Overview of Channel-Remix

Figure 2 illustrates the *Channel-Remix* pipeline. We treat the VFM as a pool of pretrained directions, where each layer contributes channels ($c_1, \ldots, c_n$: MLP hidden units and attention coordinates). Given a calibration set $\mathcal{D}_c$, our goal is to assemble a compact model that respects a parameter budget B while maximizing retained task knowledge. First, task-guided channel selection (§3.3) identifies a layer-dependent subset of retained *anchor channels* (blue in Figure 2) meeting budget B. To recover knowledge discarded during pruning without increasing size, channel remixing (§3.4) pairs each anchor c_i with k discarded donor channels (yellow) and linearly combines them into task-specific directions. Finally, weight folding s (§3.5) absorbs these combinations directly into standard weight matrices, preserving budget B and inference cost without introducing runtime overhead.

3.3 Task-Guided Channel Selection

To instantiate a student model meeting the budget B, we first partition the VFM's parameter space into retained and discarded channels. As depicted in Figure 2, this corresponds to structured pruning across layers, removing MLP intermediate units alongside their input–output slices and removing attention heads or head dimensions to preserve consistent projection ranks. The budget allocation across layers reflects task sensitivity, allowing more critical layers to retain higher capacity.

Crucially, *Channel-Remix* requires only the resulting binary partition into retained and discarded sets, making the framework agnostic to the underlying selection strategy. While our main experiments adopt CarveVFM (Mecharbat et al., 2026) to allocate layerwise widths, any standard structured pruning criterion can serve as this initial step (ablated in Appendix C.8). This selection yields an initial compact Transformer-based student where every surviving channel maps directly to a counterpart in the VFM. We designate each such channel as an anchor: it preserves the original pretrained weights and serves as the primary coordinate that subsequent remixing will augment.

3.4 Channel Remixing

Structured pruning inevitably discards channels carrying valuable knowledge. Also, task-critical information can be distributed across channels whose individual pruning appears harmless but whose collective omission degrades performance. *Channel-Remix* mitigates this by remixing discarded donor channels into the retained anchors, recovering lost knowledge without expanding capacity. As shown in Figure 2, remixing have two steps guided by $\mathcal{D}_c$: donor selection, which identifies K complementary channels per anchor, and coefficient refinement, which optimizes their linear blend.

a. Donor Channel Selection. Donor selection restricts contributors to a sparse set of K donors per anchor ($K \ll C$). Because a donor's utility depends non-linearly on interactions with co-selected channels, greedy one-shot ranking is suboptimal. We therefore employ a successive-halving strategy (Blumensath & Davies, 2009; Li et al., 2018) outlined in Algorithm 1. For an anchor c_j, candidates from the pruned pool $\mathcal{P}_\ell$ are initially pre-filtered by feature similarity to form a candidate set $\mathcal{N}_j$. We then alternate between briefly fitting mixing coefficients on $\mathcal{D}_c$ and pruning the lowest-scoring candidates based on coefficient magnitude, gradient sensitivity, and mutual redundancy, iteratively halving $|\mathcal{N}_j|$ until K donors remain.

Algorithm 1 Iterative Donor Selection for Anchor Channel c_j

Require: Anchor c_j, pruned pool $\mathcal{P}_\ell$, calibration set $\mathcal{D}_c$, donor budget K.
1: Initialize candidate set $\mathcal{N}_j \subset \mathcal{P}_\ell$ with top-32 channels ranked by cosine similarity to c_j.
2: **for** target size $s \in \{16, 8, K\}$ **do**
3: Fit mixing coefficients $\{\eta_{jr}\}_{r \in \mathcal{N}_j}$ on $\mathcal{D}_c$ for E epochs.
4: Score each candidate $r \in \mathcal{N}_j$ via magnitude, loss gradient, and redundancy.
5: Prune lowest-scoring candidates: $\mathcal{N}_j \leftarrow \text{Top-}\, s(\mathcal{N}_j)$.
6: **end for**
7: **return** Final donor set $\mathcal{N}_j$ with $|\mathcal{N}_j| = K$.

b. Mixing Coefficient Refinement. Once the donor neighborhoods $\mathcal{N}_j$ are fixed, each compact channel $\tilde{c}_j$ is parameterized as a sparse linear combination:

$$\tilde{c}_j = s_j \, c_j + \textstyle\sum_{r \in \mathcal{N}_j} \eta_{jr} \, c_r, \tag{2}$$

where scalar s_j modulates anchor scale and η_{jr} controls donor contribution. Geometrically, Eq. equation 2 synthesizes a new basis vector within the subspace spanned by the VFM's pretrained weights, recovering expressive directions inaccessible to rigid pruning alone.

With structural connectivity fixed, we optimize $\mathbf{\Phi} = \{s_j, \eta_{jr}\}$ on $\mathcal{D}_c$ using a composite objective:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{KD}} + \lambda_2 \mathcal{L}_{\text{rep}} + \lambda_3 \mathcal{L}_{\text{reg}}. \tag{3}$$

Here, $\mathcal{L}_{\text{task}}$ is cross-entropy on task labels. The transfer terms regularize optimization toward the teacher's manifold: $\mathcal{L}_{\text{KD}} = \tau^2 \, \text{KL}(p_T^\tau \,\|\, p_S^\tau)$ aligns soft prediction probabilities, while $\mathcal{L}_{\text{rep}} = \text{NMSE}(z_S, z_T) + \mathcal{L}_{\text{attn}}$ matches intermediate feature maps and attention distributions. Finally, $\mathcal{L}_{\text{reg}} = \sum_j (|s_j - 1|^2 + \sum_r |\eta_{jr}|^2)$ acts as an inductive bias, penalizing excessive divergence from the original anchor weights and preventing overfitting on scarce samples.

Channel Remixing Across Transformer Layers. Applying Eq. equation 2 requires respecting architectural weight couplings across layer types. In feed-forward networks (MLPs), hidden dimension j couples the column slices of the up-projection ($W_{\text{gate}}, W_{\text{up}}$) with the row slices of the down-projection (W_{down}) and the intermediate bias, requiring them to share identical mixing coefficients $\{s_j, \eta_{jr}\}$. In multi-head attention modules, query (Q) and key (K) projections share a unified mixing map to preserve dot-product geometry, whereas value (V) and output projection (O) share a distinct second map. When pruning operates at the head level, retained heads recruit donors from discarded heads; when pruning operates within heads, donors are drawn strictly within the same head coordinate space. Full matrix formulations, donor eligibility criteria, and gated MLP variants (e.g., SwiGLU) are detailed in Appendix A.

3.5 Weight Folding and Deployment

During remixing, the compact student retains explicit mixing graphs connecting anchors and their recruited donors. Once optimization concludes, *weight folding* collapses each remixed channel $\tilde{c}_j$ directly into a static compact weight matrix: $W_S = \mathbf{M} W_T$, where $\mathbf{M} \in \mathbb{R}^{C_S \times C_T}$ is a sparse recombination matrix whose j-th row contains s_j at anchor index j and η_{jr} at donor indices $r \in \mathcal{N}_j$. Multiplying the frozen VFM projection matrices by $\mathbf{M}$ absorbs all donor contributions offline into standard linear projections. The mixing graphs and donor weights are then discarded, leaving a deployed model that strictly respects budget B with identical depth, layer interfaces, and runtime latency to a standard pruned baseline.

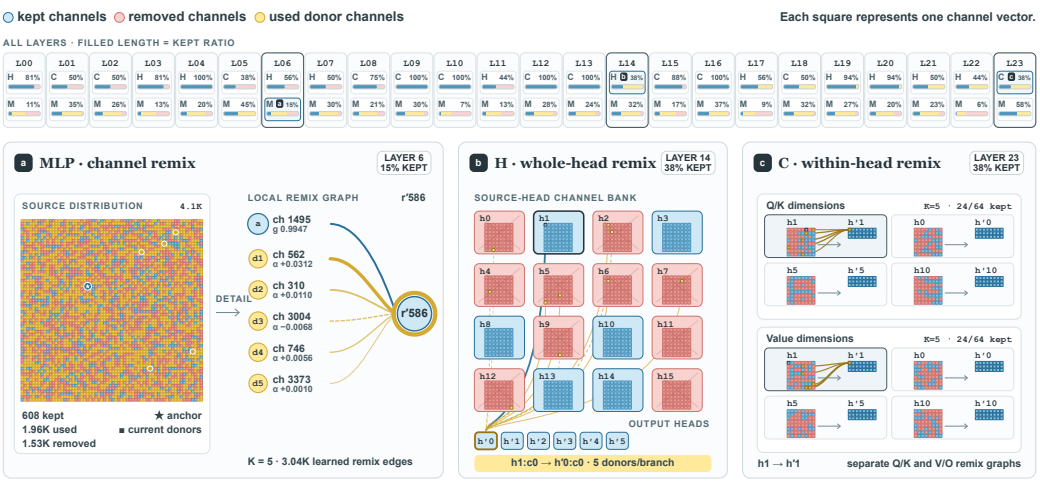


Figure 3: **Overview of channel remixing graphs and layer allocations.** *Up:* Layerwise proportions of retained anchors (blue), recruited donors (yellow), and discarded channels (red) across attention (H or C) and MLP (M) blocks in ViT-L. *Down:* (a) A single MLP anchor reinforced with $K{=}5$ donors; (b) cross-head remixing combining 16 source heads into 6 compact heads; (c) within-head remixing with separate graphs for Q/K and V/O. Weight folding merges these connections into static matrices, creating the compact model with zero inference overhead.

Figure 3 visualizes this parameter construction across the network. The backbone overview displays layerwise retention ratios alongside the recruited donor allocations. Panel (a) traces a single compact MLP channel back to its anchor ($s = 0.9947$) and $K{=}5$ complementary donors within a pruned intermediate layer. Panels (b) and (c) detail the attention layers: whole-head remixing (b) recombines channels from a 16-head source bank into 6 compact output heads ($\mathrm{h}'0$–$\mathrm{h}'5$), while within-head remixing (c) operates over pruned sub-head dimensions ($24/64$ channels kept) using separate remix graphs for Q/K and V/O projections. Crucially, these connections depict how compact weights are synthesized, not runtime routing: after folding, every donor contribution is pre-computed into static weights, introducing zero inference overhead. We then freeze the folded backbone and train a lightweight linear classifier on target task labels.

4 Experiments

4.1 Experimental Setup

Scope and Benchmarks. We evaluate on sixteen vision datasets spanning general recognition, fine-grained, textures, scenes, medical imaging, remote sensing, counting, and viewpoint estimation. The main comparison covers ten source-model and compact-budget settings drawn from seven pretraining families: DINOv3 (ViT-S, ViT-S+, ViT-B, ViT-L, and ViT-H+) (Siméoni et al., 2025), MetaCLIP 2 (Chuang et al., 2025), SigLIP 2 (Tschannen et al., 2025), MAE (He et al., 2022), AugReg (Steiner et al., 2022), BiomedCLIP (Zhang et al., 2023), and I-JEPA (Assran et al., 2023).

Protocol and Baselines. All methods start from the same pruned model and adapt it using the same calibration set of $N{=}512$ images, so any difference in performance reflects the method effect. We also examined data efficiency by testing budgets from 256 to 2,048 images. We evaluate all models with linear probing (Alain & Bengio, 2016), the standard VFM protocol (Siméoni et al., 2025): every adapted backbone is frozen, so accuracy reflects representation quality, without the cost and confound of retraining the backbone on each task. The pruned model serves as the lower bound and the uncompressed one as the upper reference. We compare *Channel-Remix* against three groups of baselines: *post-pruning recovery* (B+LN, AdaPrune, Ridge, and Neuron Merging (Kim et al., 2020)), *parameter-efficient adaptation* (SSF (Lian et al., 2022), SVFT (Lingam et al., 2024), AdaptFormer (Chen et al., 2022), LoRA with ranks 2, 4, and 8 (Hu et al., 2022), DoRA, and PiSSA), and *full-weight methods* (block reconstruction, full fine-tuning, and Dense Remix (Zhmoginov et al., 2024)). Training schedules, data splits, and optimizer settings are detailed in Appendix B.4.

Table 1: **Accuracy gain over the compressed model with 512 calibration images** (mean over 16 tasks). Bold outlines mark the two best gains per column. With matched training schedules, Remix (K=5) beats PiSSA (r=8) in 78 of 96 task setting pairs (Wilcoxon $p < 10^{-7}$).

Method	Avg. trainable params	DINOv3 ViT-S	DINOv3 ViT-B	DINOv3 ViT-L	DINOv3 ViT-H+	DINOv3 *Avg.*	Other pretraining Meta-CLIP 2	Other pretraining Sig-LIP 2	Other pretraining I-JEPA	Other pretraining MAE	Other pretraining ViT-AugReg	Other pretraining Biomed-CLIP	Other pretraining *Avg.*
Size, uncomp. (M)		22	87	304	841		86	93	631	86	86	86	
Size, comp. (M)		10	35	150	300		35	35	250	35	35	35	
Compressed acc. (%) [Lower Bound]		74.1	76.5	85.9	86.5	**80.7**	68.2	72.9	70.0	55.7	79.5	51.3	**66.3**
Uncomp. (Δ vs. Comp.) [Upper Bound]		**+8.4**	**+8.5**	**+2.0**	**+1.0**	**+5.0**	**+16.5**	**+12.8**	**+0.2**	**+5.2**	**+1.8**	**+1.9**	**+6.4**
Post-pruning recovery													
B+LN	181k	+3.5	+0.8	+1.1	−5.1	+0.1	+3.2	+3.6	+2.1	+4.4	−0.3	+0.5	+2.3
AdaPrune	25.0M	−0.2	−0.3	−0.2	−1.6	−0.6	−6.7	−9.0	−1.4	−3.1	−2.8	+0.1	−3.8
Ridge	25.0M	−34.7	−33.9	−44.2	−52.4	−41.3	−22.1	−35.7	−4.1	−6.5	−9.1	−2.5	−13.3
Neuron merge	54k	−35.6	−30.9	−45.2	−51.9	−40.9	−26.1	−46.1	−51.7	−19.5	−58.9	−9.5	−35.3
Parameter-efficient adaptation													
SSF	187k	+3.4	+4.5	+1.4	−0.9	+2.1	+2.6	+3.2	+2.8	+3.9	−0.3	+0.7	+2.1
SVFT	177k	+2.3	+3.2	+0.8	−1.7	+1.2	+2.3	+3.2	+2.2	+2.1	−0.2	+1.0	+1.8
AdaptFormer	172k	+1.6	+2.9	+0.5	−1.7	+0.8	+2.2	+3.1	+2.6	+2.0	−0.2	+1.5	+1.9
LoRA (r=2)	157k	+2.9	+4.5	+1.1	−1.1	+1.8	+2.9	+3.3	+2.0	+4.2	+0.1	+1.3	+2.3
LoRA (r=4)	260k	+3.1	+4.8	+1.1	−1.0	+2.0	+3.2	+3.3	+2.2	+5.0	+0.1	+1.4	+2.5
LoRA (r=8)	465k	+3.3	+5.2	+1.2	−0.8	+2.2	+3.2	+3.4	+2.2	+5.7	+0.2	+1.5	+2.7
DoRA (r=8)	491k	+3.4	+5.0	+1.4	−0.9	+2.2	+3.2	+3.6	+2.9	+5.7	+0.2	+1.6	+2.9
PiSSA (r=8)	465k	+3.9	**+6.1**	+1.5	−0.4	+2.8	+3.3	+4.5	+3.5	+6.3	+0.4	+1.9	+3.3
Full tuning													
Block recon.	69.2M	+3.4	+3.4	−2.0	−0.4	+1.1	+4.8	**+5.9**	+3.2	+6.8	+0.6	+3.3	+4.1
Full fine-tune	69.2M	+3.6	+3.7	+0.7	−7.0	+0.3	+4.9	**+5.4**	+2.8	+6.5	+1.1	+1.9	+3.7
Dense remix	70.3M	+4.5	−1.8	−29.0	−42.3	−17.2	−14.1	−30.0	−58.1	−6.4	−12.6	−3.1	−20.7
Channel-Remix													
K=3	173k	**+5.1**	+5.6	**+2.0**	+1.5	+3.6	**+6.7**	+4.7	+3.2	**+8.1**	**+1.3**	+2.9	**+4.5**
K=5	233k	**+5.4**	**+6.4**	**+1.8**	**+1.8**	**+3.8**	**+6.8**	+3.9	**+3.7**	**+8.1**	**+1.4**	**+3.4**	**+4.6**
K=7	292k	**+5.4**	+5.5	**+1.8**	**+2.0**	**+3.7**	**+6.7**	+3.3	**+3.8**	**+8.3**	+1.0	**+3.9**	**+4.5**

4.2 Results & Discussion

Table 1 reports mean accuracy over all sixteen benchmarks with 512 calibration images. The top rows give the compressed model size, its accuracy (Base), and its gap to the uncompressed model. Method rows report the change relative to Base in percentage points, with the top two methods per column outlined in bold.

Recovery over pruning. *Channel-Remix* improves over the compressed model in all ten settings, raising mean accuracy from 70.4% to 75.0% (+4.6 points). Gains are largest under heavy compression: +6.4 for DINOv3 ViT-B/35M, +6.8 for MetaCLIP 2, and +8.1 for MAE. This is consistent with task-relevant information being distributed across co-dependent channels, whose joint signal can be partly recovered by recombining discarded donors. On larger models (ViT-L, ViT-H+), remixing gains +1.8 points and nearly closes the gap to the uncompressed model. These gains are smaller because little headroom remains: many benchmarks are near saturation, and even full fine-tuning gains little (+0.6 on ViT-L/150M).

Comparison with adaptation baselines. Across the six non-DINO families, *Channel-Remix* reaches 70.8% mean accuracy, versus 69.0% for LoRA-8 and 69.6% for PiSSA-8, with 56.6% fewer trainable parameters (K=5: 179k coefficients on average vs. 411k for LoRA-8, excluding classifiers). Unlike low-rank adapters, which can learn arbitrary directions, *Channel-Remix* restricts adaptation to the span of pretrained channels, an inductive bias that can reduce overfitting when calibration data are limited. Adaptation takes about 2 min/task for all methods, and the folded model has the same inference cost as the compressed one (more details in Appendix C.5).

Sparsity stabilizes recombination. Dense channel mixing degrades with limited calibration data. On the six non-DINO families, Neuron Merging and Ridge lose 35.3 and 13.3 points on average, and Dense Remix drops by 17.2 points on DINOv3 and 20.7 on other families. Letting every pruned channel contribute introduces far more free variables than 512 samples can constrain, which can make the mixing maps overfit. Restricting each anchor to a sparse set of donors ($K \in \{3, 5, 7\}$) yields stable gains across all architectures, nearly unchanged from K=3 to K=7 (+4.5 to +4.6 points), so K needs no careful tuning.

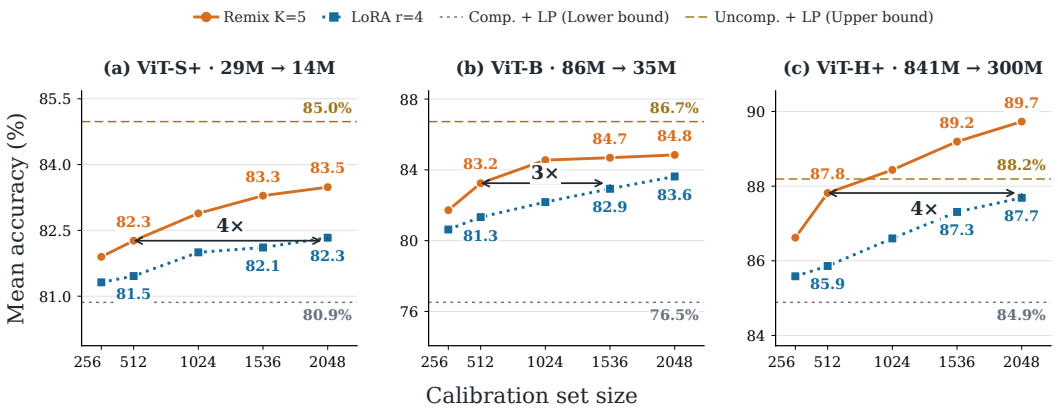


Figure 4: **Backbone adaptation with limited task data.** Sixteen-dataset mean accuracy for *Channel-Remix* ($K = 5$) and LoRA ($r = 4$), with matched adaptation and classifier epochs. Arrows compare 512-image remixing with the indicated LoRA budgets. Pruning and reconstruction use 512 images; linear probes (LP) use full task data. The plotted lower and upper references denote compressed and uncompressed LP references.

Table 2: **Component ablation.** ViT-S/6M, K=3 MLP donors per channel. (a) Test accuracy (%) with full task data when removing one component. (b) Gain from each donor-discovery component on held-out MIT-67, with checkpoints chosen on a calibration holdout.

(a) Construction, full task data					*(b) Donor discovery, held-out MIT-67*				
Variant	Aircraft	Cal-256	MIT-67	Mean	Component (vs.)	N=512	1,024	2,048	Mean
Channel-Remix	40.6	52.4	53.8	**48.9**	Succ. halving (one-shot $32 \to K$)	+4.6	+3.2	−0.4	+2.4
w/o donors (anchor only)	22.5	34.8	37.6	31.7	Full objective (local recon.)	+4.3	+1.3	−0.6	+1.7
w/o support (random)	39.7	50.3	52.7	47.6	Discovery (top-K refine only)	+0.4	−0.9	+2.1	+0.5
w/o support (top-grad.)	37.9	49.0	49.6	45.5					
w/o teacher guidance	34.4	47.6	47.8	43.3	*Channel-Remix* discovery, acc. (%)	30.0	33.4	41.9	35.1
Compressed start	9.8	26.8	24.8	20.5					

Where broader updates help. Remixing does not win on every benchmark. On SigLIP 2, *Channel-Remix* gains 3.9 points, while full fine-tuning and block reconstruction gain 5.4 and 5.9. Because remixing stays within the span of pretrained channels, it trades representational flexibility for sample efficiency; when a task needs features far from the pretraining distribution, unconstrained updates can move beyond this subspace and may retain an edge.

Data efficiency. Restricting adaptation to sparse combinations of pretrained directions lowers the effective dimensionality of the optimization, so fewer examples should suffice. We compare *Channel-Remix* (K=5) with LoRA-4 for calibration budgets from 256 to 2,048 images on all sixteen datasets, with 20 adaptation epochs, 10-epoch classifiers, and fixed pruning and reconstruction. With 512 images, *Channel-Remix* reaches 83.2% on ViT-B/35M, above LoRA-4 with 1,536 images (82.9%), and 87.8% on ViT-H+/300M, above LoRA-4 with 2,048 images (87.7%) (Figure 4). It thus matches LoRA with 3–4× less calibration data, and its accuracy keeps rising from 256 to 2,048 images for all three model sizes, so the parameterization does not saturate early.

Dataset-Level Analysis. *Channel-Remix* outperforms LoRA in 132 of 144 matched settings, confirming systematic gains across architectures and tasks (Appendix B.8). Improvement magnitude scales with task-model alignment: distributed tasks (CLEVR Count: +1.5 on ViT-S+ vs. +12.9 on ViT-H+) and fine-grained classification (Stanford Cars: +17.2) benefit most from richer donor pools and joint feature combinations, whereas localized texture recognition (DTD: +0.2) shows marginal impact. Minor regressions (e.g., smallNORB: −4.0) occur only when retained anchors already span the target subspace, where recruiting redundant donors introduces optimization noise.

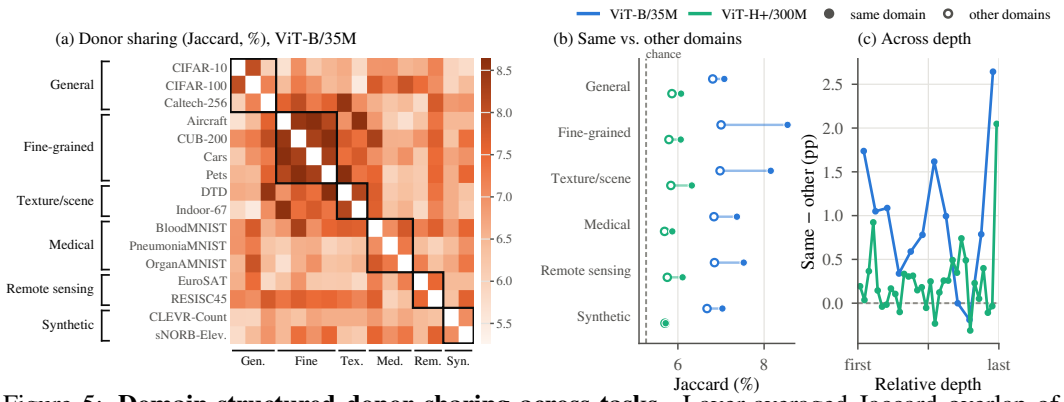


Figure 5: **Domain-structured donor sharing across tasks.** Layer-averaged Jaccard overlap of top-10% donors between task pairs. (a) Cross-task overlap matrix grouped by visual domain on ViT-B/35M. (b) Sharing rates for intra-domain versus cross-domain pairings. (c) Net intra-domain sharing advantage across backbone depth.

5 ANALYSIS AND ABLATION

Shared and Task-Specific Knowledge Because all students originate from the same frozen teacher, their mixing maps share a unified coordinate basis and provide a transparent lens for cross-task interpretability. Measuring *donor sharing* via Jaccard overlap of top-10% donors under same settings (K=5, N=1,024) reveals how visual domains organize within the frozen backbone. As shown in Figure 5, donor recruitment exhibits clear domain clustering: same-domain tasks share consistently more donors than cross-domain pairs across all model scales (e.g., 7.8% vs. 6.9% on ViT-B/35M, $p < 10^{-4}$), with domain affinity concentrating in the earliest and deepest layers.

This emergent structure has two key implications. First, for mechanistic interpretability, it suggests that foundation models segment their channels into domain-salient functional clusters rather than purely generic features. Second, for multi-task and zero-shot transfer, predictable donor sharing suggests that cross-task routing graphs can be shared or predicted a priori, paving the way for joint multi-task synthesis without optimizing separate remixing graphs for every task.

***Channel-Remix* components ablation.** Table 2 removes one component at a time. Donors account for most of the gain: without them, *Channel-Remix* loses 17.3% of the 28.5% it recovers over the compressed start. Teacher guidance follows (5.7%). The donor support must be learned, since fixing it to random or top-gradient donors lowers accuracy on all three tasks. With limited calibration data, successive halving and the full objective help most at N=512 (+4.6 and +4.3), and together let *Channel-Remix* match refinement over all pruned channels while scoring only 32 candidates per channel (Appendix C.7).

6 CONCLUSION

We introduced *Channel-Remix*, a method for transferring knowledge from frozen vision foundation models to smaller models under parameter budgets. *Channel-Remix* combines task-guided channel selection with channel remixing to fold discarded donor features into retained anchor channels. The resulting model deploys standard weights with no parameter or inference overhead, while outperforming parameter-efficient baselines. Our experiments also reveal informative trade-offs. When anchor channels already span the features needed for the target task, recruiting additional donors yields diminishing returns and can introduce some noise. Our formulation is further limited to intra-layer remixing: cross-layer and cross-model composition remain unexplored.

These observations suggest several directions for future work. Learning layer-wise donor sparsity would allow the model to filter redundant donors automatically instead of fixing the budget by hand. Extending remixing across layers, and applying it to language and multimodal foundation models and also desnse vision tasks, would test whether the anchor–donor principle generalizes beyond vision. Finally, folded anchor weights are themselves compact, reusable units, making them a promising substrate for continual and multi-task transfer.

AI USE STATEMENT

Generative AI tools were used only to assist with manuscript drafting and grammatical and language editing. They were not used to develop the method, generate data, implement the research methods, execute experiments, or analyze experimental results. The authors take responsibility for the final manuscript and all reported claims and results.

ETHICS STATEMENT

Channel-Remix is evaluated on existing image-classification benchmarks, including three medical-image benchmarks. The experiments assess classification accuracy and computational efficiency. These benchmark results do not establish clinical safety or effectiveness; compressed models would require application-specific validation before clinical use. The reported evaluation does not measure demographic fairness or privacy, so accuracy and efficiency gains should not be interpreted as evidence of improvements in those properties.

REPRODUCIBILITY STATEMENT

Section 3 introduces the Channel-Remix framework, describing the task-guided channel selection, donor discovery via successive halving, mixing coefficient refinement, and exact weight folding. Appendix A provides explicit sparse remix map definitions, architecture formulations across MLP and attention layers, objective formulations, and the layerwise attention safeguard criteria. Appendix B details the benchmark datasets, source foundation models and compression budgets, baseline configurations, training protocols, and statistical significance tests. Per-dataset classification accuracies, data-efficiency scaling sweeps, and per-task comparisons against LoRA are reported in Appendices B.6, B.7, and B.8, respectively. Appendix C presents extended analyses, including cross-task donor sharing, donor budget and calibration size ablations, rescaling versus recombination experiments, diagnostics, and alternative pruning starting points.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Shubhankar Borse, Shreya Kadambi, Nilesh P Pandey, Kartikeya Bhardwaj, Viswanath Ganapathy, Sweta Priyadarshi, Risheek Garrepalli, Rafael Esteves, Munawar Hayat, and Fatih Porikli. Foura: Fourier low-rank adaptation. *Advances in Neural Information Processing Systems*, 37:71504–71539, 2024.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16664–16678. Curran Associates, Inc., 2022. doi: 10.52202/068431-1212.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, Xinlei Chen, Zhuang Liu, Saining Xie, Wen-tau Yih, Shang-Wen Li, and Hu Xu. Meta CLIP 2: A worldwide scaling recipe, 2025.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16091–16101. IEEE, 2023.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

Shuichiro Haruta, Kazunori Matsumoto, Zhi Li, Yanan Wang, and Mori Kurokawa. RCPU: Rotation-constrained error compensation for structured pruning of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=t6xiPRvynD`.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Woojeong Kim, Suhyun Kim, Mincheol Park, and Geunseok Jeon. Neuron merging: Compensating for pruned neurons. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Jongwoo Ko, Seungjoon Park, Minchan Jeong, Sukjin Hong, Euijai Ahn, Du-Seong Chang, and Se-Young Yun. Revisiting intermediate layer distillation for compressing language models: An overfitting perspective. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 158–175, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.12. URL https://aclanthology.org/2023.findings-eacl.12/.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

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

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJqFGTslg.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of machine learning research*, 18(185):1–52, 2018.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2210.08823.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. Mixkd: Towards efficient distillation of large-scale language models, 2021. URL https://arxiv.org/abs/2011.00593.

Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham K Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: Parameter-efficient fine-tuning with singular vectors. *Advances in neural information processing systems*, 37:41425–41446, 2024.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32100–32121. PMLR, 21–27 Jul 2024.

Zheda Mai, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Quang-Huy Nguyen, Li Zhang, and Wei-Lun Chao. Lessons and insights from a unifying study of parameter-efficient fine-tuning (peft) in visual recognition. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14845–14857. IEEE, 2025.

Subhransu Maji, Juho Kannala, Esa Rahtu, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013.

Lotfi Abdelkrim Mecharbat, Abdelrahman Elsayad, Ibrahim Almakky, Martin Takáč, and Mohammad Yaqub. CarveVFM: Carving specialists out of generalists, automatic task-specific compression of vision foundation models, 2026. Under review.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413–420, 2009.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6550.

Jiacheng Ruan, Mingye Xie, Jingsheng Gao, Xian Gao, Suncheng Xiang, Ting Liu, and Yuzhuo Fu. TTE: Two tokens are enough to improve parameter-efficient tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20209–20217, 2025. doi: 10.1609/aaai.v39i19.34226. URL https://ojs.aaai.org/index.php/AAAI/article/view/34226.

Sungbin Shin, Wonpyo Park, Jaeho Lee, and Namhoon Lee. Rethinking pruning large language models: Benefits and pitfalls of reconstruction error minimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1182–1191. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.68. URL https://aclanthology.org/2024.emnlp-main.68/.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech–ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. Pmlr, 2022.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023. doi: 10.1038/s41597-022-01721-8.

Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12135–12144. IEEE, 2022.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Andrea Tupini, Yu Wang, Matt Mazzola, Swadheen Shukla, Lars Liden, Jianfeng Gao, Matthew P. Lungren, Tristan Naumann, Sheng Wang, and Hoifung Poon. BiomedCLIP: A multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs, 2023.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11953–11962, 2022.

Andrey Zhmoginov, Jihwan Lee, and Mark Sandler. Projectable models: One-shot generation of small specialized transformers from large ones. In *ICML 2024 Workshop on Efficient Systems for Foundation Models II (ES-FoMo II)*, 2024. URL https://openreview.net/forum?id=F7mGtT3fQh.

APPENDIX OVERVIEW

- **A Methodological Details**: Remix maps for MLPs, gated MLPs (SwiGLU), and attention projections; composite distillation objective; donor discovery via successive halving; attention safeguard criteria; exactness of weight folding; and theoretical approximation bounds.
- **B Experimental Details and Complete Results**: dataset specifications and source model configurations; baseline implementations and training protocols; statistical significance tests; comprehensive per-dataset breakdowns; data-efficiency scaling; and per-task gains against LoRA.
- **C Extended Analyses and Ablation Studies**: cross-task donor sharing and domain clustering across thresholds and rank metrics; sensitivity to donor budget (K) and calibration size (N); directional recombination versus channel rescaling; non-uniqueness of donor subspaces; attention remixing failure diagnostics; component ablation protocols; donor discovery dynamics; alternative pruning criteria; and update parameterizations under unlimited data.

All experiments use adaptation seed 17. Unless stated otherwise, accuracy is top-1 test accuracy (%) of a frozen backbone with a freshly trained linear classifier, averaged with equal weight over tasks.

A METHOD DETAILS

A.1 SPARSE REMIX MAPS

Consider a source layer with d channels and a compact layer with $m < d$ channels. Channel selection assigns compact channel j an anchor index s_j; let $S = [e_{s_1}, \dots, e_{s_m}] \in \{0,1\}^{d\times m}$. A remix map is

$$R = S\,\mathrm{diag}(g) + C, \qquad R_{:,j} = g_j e_{s_j} + \sum_{r\in\mathcal{N}_j} \eta_{jr} e_r, \tag{A.1}$$

where g_j is the anchor scale (s_j in Eq. equation 2 of the main paper), η_{jr} are donor coefficients, and $\mathcal{N}_j$ is the donor set of size at most K. MLP donors are drawn from channels removed by pruning. The map has at most $m(K{+}1)$ active coefficients, against dm for a dense projection. Initializing $g = \mathbf{1}$ and $C = 0$ gives $R = S$, which reproduces the pruned model exactly.

A.2 MLP CHANNELS

An MLP with input projection $U \in \mathbb{R}^{d\times p}$, bias b_U, activation σ, and output projection $D \in \mathbb{R}^{p\times d}$ computes $D\,\sigma(Ux + b_U) + b_D$. Hidden channel i couples row i of U, entry i of b_U, and column i of D, so remixing applies one map to all three:

$$U_R = R^\top U + E_U, \qquad b_R = R^\top b_U + E_b, \qquad D_R = DR + E_D. \tag{A.2}$$

The combination acts before the nonlinearity, so a remixed channel is a new function of the input rather than a sum of activated source outputs.

Reconstruction offsets. The pruned starting model may already contain reconstruction corrections, with weights $\bar{U}, \bar{b}_U, \bar{D}$ that differ from the source slices. We keep these corrections as fixed offsets, $E_U = \bar{U} - S^\top U$, $E_b = \bar{b}_U - S^\top b_U$, $E_D = \bar{D} - DS$, so that $R = S$ recovers $(\bar{U}, \bar{b}_U, \bar{D})$ exactly. The output bias $\bar{b}_D$ stays fixed. Without corrections the offsets vanish and Eq. equation A.2 reduces to $W_S = \mathbf{M}W_T$ with $\mathbf{M} = R^\top$, the form given in Section 3.5 of the main paper.

Gated MLPs. In gated MLPs (e.g., SwiGLU), channel i couples rows of both the gate and the up projections with a column of the down projection. Both producer projections use the same map R, each with its own fixed offset, and the down projection uses $DR + E_D$.

A.3 ATTENTION CHANNELS

Queries and keys jointly determine attention scores, while values and the output projection determine the contribution to the residual stream. We therefore use one map R_{QK} for the query and key

projections and a second map R_{VO} for the value and output projections:

$$W_Q^R = R_{\mathrm{QK}}^\top W_Q + E_Q, \quad W_K^R = R_{\mathrm{QK}}^\top W_K + E_K,$$
$$W_V^R = R_{\mathrm{VO}}^\top W_V + E_V, \quad W_O^R = W_O R_{\mathrm{VO}} + E_O. \tag{A.3}$$

Query, key, and value biases follow their maps; the output bias is fixed. The remixed projections feed the compact model's unchanged positional transformations and softmax, so the order of nonlinear operations is preserved.

Donor eligibility. When pruning narrows each head (column pruning), a retained coordinate recruits donors from the same source head, which makes the maps block-diagonal across heads. When pruning removes whole heads, a coordinate of a retained head may recruit individual coordinates of discarded heads. An attention operator is remixed only if at least three discarded coordinates are available in each relevant bank; when fewer than K discarded coordinates exist, non-anchor retained coordinates of the same head pad the candidate set. All parameters outside the remix maps, including normalization layers, stay frozen.

A.4 Training Objective

The mixing coefficients and a calibration classifier are trained with

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \tau^2 \,\mathrm{KL}(p_T^\tau \,\|\, p_S^\tau) + \mathrm{NMSE}(z_S, z_T) + \tfrac{1}{|\mathcal{A}|} \textstyle\sum_{\ell \in \mathcal{A}} \mathrm{NMSE}\big(a_S^\ell, a_T^\ell\big) + \mathcal{L}_{\mathrm{reg}}, \tag{A.4}$$

with $\tau = 2$ and unit weights on the first four terms. This matches Eq. equation 3 of the main paper with $\lambda_1 = \lambda_2 = 1$: $\mathcal{L}_{\mathrm{rep}}$ consists of the final representation z and the post-attention outputs a^ℓ of the layers $\mathcal{A}$ whose attention maps are enabled. The normalized error divides the squared error by the teacher's energy, with a numerical floor. The teacher uses a fixed task classifier trained before remixing; teacher outputs are detached. For every map,

$$\mathcal{L}_{\mathrm{reg}}(R) = 10^{-4}\ \mathrm{MSE}(R - S,\, 0) + 10^{-3}\ \mathrm{MSE}(R^\top R,\, I). \tag{A.5}$$

Because $R - S = \mathrm{diag}(g - \mathbf{1}) + C$ on the anchor and donor entries, the first term is the penalty $\sum_j (|s_j - 1|^2 + \sum_r |\eta_{jr}|^2)$ of the main paper, normalized by the number of entries; the second keeps remixed channels from collapsing onto one another.

A.5 Donor Discovery and Coefficient Refinement

Candidate pools. Each MLP anchor starts with 32 candidate donors drawn from the pruned channels of its layer. In the primary framework formulated in Section 3.4 and Algorithm 1, candidates are initialized by ranking pruned channels by cosine feature similarity to the anchor c_j. In our exploration of candidate priors (evaluated extensively in Appendix C.7), we also examined alternative initialization heuristics, including a cross-task linear donor ranker trained on 53 activation, pairwise, and gradient features from independent tasks (for $N \le 512$) as well as uniform random sampling. As demonstrated by the ablation in Table C.3, successive halving reliably isolates top-performing donor combinations under all initialization policies, confirming that performance gains are driven by the downstream optimization objective rather than handcrafted candidate priors.

Scoring and successive halving. Candidates are fitted jointly under the full objective and reduced as $32 \to 16 \to 8 \to K$. The main DINOv3 runs use five epochs per stage (twenty search epochs); the non-DINO and data-efficiency runs use ten search epochs. For donor r of anchor j, let $F_{rr} = \frac{1}{T}\sum_t (\partial \mathcal{L}_t / \partial \eta_{jr})^2$ accumulate squared minibatch gradients over the stage, evaluated at the stage checkpoint with the best calibration-holdout accuracy. The score adds the normalized sensitivity $\eta_{jr}^2 F_{rr}$ and the normalized magnitude $|\eta_{jr}|$ with weight 0.1, normalizing each within the anchor's candidate set. Greedy selection then subtracts 0.1 times the largest absolute gradient correlation $|F_{rq}|/\sqrt{F_{rr}F_{qq}}$ with any donor q already selected for the same anchor. Attention operators use the sensitivity and magnitude terms only, and adapt their candidate counts to the number of eligible coordinates.

Coefficient refinement. After discovery, coefficients are refined end to end for twenty epochs (main DINOv3 runs) or ten epochs (other runs). After each update, only the K largest-magnitude eligible donor coefficients per anchor are kept. The donor budget is fixed, but donor identities may change: a zeroed coefficient still receives gradients and can re-enter. Anchor indices never change.

Data use. The calibration set is class-balanced where possible. A quarter of it ($N/4$ images) is held out for checkpoint selection and the attention safeguard; coefficients are fitted on the remainder. Test data are never used for any selection.

A.6 Attention Safeguard

Changing queries and keys changes attention routing, so a poor combination can disrupt a whole layer (Appendix C.4). After the first search epoch, the main DINOv3 runs check the mean normalized attention-output error on calibration-training batches and the accuracy on the calibration holdout. If the error exceeds 5 and holdout accuracy falls by more than three points relative to the starting model, the initialization is restored, attention maps are disabled in the layers above the error threshold, and the remaining maps continue training. Across the 12-task ablation suite, this rule triggered in 84 of the 720 runs. The non-DINO and data-efficiency runs use a global variant: attention remixing is disabled when the error exceeds twice its initial value or 4.5 and the holdout accuracy drops by more than three points, leaving MLP-only remixing. Neither rule uses test accuracy.

A.7 Weight Folding Is Exact

During calibration, a remixed MLP block evaluates $f_R(x) = (DR + E_D)\,\sigma\big((R^\top U + E_U)x + R^\top b_U + E_b\big) + \bar{b}_D$. The exported block stores D_R, U_R, b_R, and $\bar{b}_D$ from Eq. equation A.2 and evaluates the same expression, so folding introduces no approximation beyond floating-point rounding. The same holds for attention, since the remixed projections are applied before all attention operations both during calibration and after export. Every run records an export check that compares intermediate and final outputs of the calibration-time and folded models within numerical tolerance. The folded model has the same depth, widths, operators, and parameter count as the pruned model; the source bank, maps, and candidate machinery are discarded.

A.8 Why a Few Donors Can Suffice

Component absorption. For a linear pair, split the intermediate axis into retained and removed parts, $A = [A_S\ A_{\mathcal{R}}]$ and $B = [B_S; B_{\mathcal{R}}]$. If the removed components are partly explained by the retained ones, $A_{\mathcal{R}} = A_S T + E$, then

$$AB = A_S(B_S + TB_{\mathcal{R}}) + EB_{\mathcal{R}}, \qquad \big\|AB - A_S(B_S + TB_{\mathcal{R}})\big\|_F \le \|E\|_F\,\|B_{\mathcal{R}}\|_2. \tag{A.6}$$

Pruning keeps only $A_S B_S$; absorbing $TB_{\mathcal{R}}$ into the retained weights recovers the part of the removed contribution that the retained channels can express.

Dense versus few-donor maps. For $F(R) = (DR + E_D)(R^\top U + E_U)$, a dense map $R_\star$, a few-donor map R_K, and $\Delta = R_\star - R_K$, adding and subtracting $(DR_K + E_D)(R_\star^\top U + E_U)$ gives

$$\|F(R_\star) - F(R_K)\|_F \le \big(\|D\|_2\|R_\star^\top U + E_U\|_2 + \|DR_K + E_D\|_2\|U\|_2\big)\|\Delta\|_F. \tag{A.7}$$

For an L_σ-Lipschitz activation, the same argument applied to one MLP block yields

$$\|f_{R_\star}(x) - f_{R_K}(x)\|_2 \le \|\Delta\|_2\Big[\|D\|_2\,\|h_{R_\star}(x)\|_2 + L_\sigma\|DR_K + E_D\|_2\,\|Ux + b_U\|_2\Big], \tag{A.8}$$

with $h_R(x) = \sigma\big((R^\top U + E_U)x + R^\top b_U + E_b\big)$. The cost of sparsifying a useful map therefore scales with the map error and the norms of the surrounding operators. These bounds motivate learning few-donor maps under the task objective; they do not guarantee that a small-K approximation exists or that the source network's nonlinear function is recovered exactly.

A.9 Active Coefficients and Training Cost

Parameter counts in Table 1 are the final active coefficients (anchor scales and donor coefficients) and, for other methods, their trained or fitted parameters. They do not measure training memory: the implementation allocates dense coefficient storage during search and keeps the frozen source bank in memory. For example, at $K{=}5$ and $N{=}512$ across the 12-task ablation suite, the median active graph counts are about 69K to 371K from S/10M to L/150M, while the dense training allocations are about 11M to 167M scalar slots. None of this is present after folding.

B Experimental Details and Complete Results

B.1 Datasets

Table B.1 lists the sixteen datasets. Training counts are the images available for the final linear classifier; calibration subsets are drawn from them. Caltech-256 includes the clutter class (257 labels), FGVC-Aircraft trains on its combined training and validation split, and DTD uses its 1,880-image training split, which caps the nominal 2,048-image calibration budget at 1,880. Images are resized to 224×224 with each source model's preprocessing; the MedMNIST tasks use their 224-pixel release.

Table B.1: **Datasets.** Domain groups are those used in Section 5 of the main paper and Appendix C.1.

Domain	Dataset	Classes	Training	Test
General	CIFAR-10 (Krizhevsky, 2009)	10	50,000	10,000
	CIFAR-100 (Krizhevsky, 2009)	100	50,000	10,000
	Caltech-256 (Griffin et al., 2007)	257	24,485	6,122
Fine-grained	FGVC-Aircraft (Maji et al., 2013)	100	6,667	3,333
	CUB-200-2011 (Wah et al., 2011)	200	5,994	5,794
	Stanford Cars (Krause et al., 2013)	196	8,144	8,041
	Oxford-IIIT Pets (Parkhi et al., 2012)	37	3,680	3,669
Texture/scene	DTD (Cimpoi et al., 2014)	47	1,880	1,880
	MIT Indoor-67 (Quattoni & Torralba, 2009)	67	5,360	1,340
Medical	BloodMNIST (Yang et al., 2023)	8	11,959	3,421
	OrganAMNIST (Yang et al., 2023)	11	34,561	17,778
	PneumoniaMNIST (Yang et al., 2023)	2	4,708	624
Remote sensing	EuroSAT (Helber et al., 2019)	10	16,200	5,400
	RESISC45 (Cheng et al., 2017)	45	18,897	6,300
Synthetic	CLEVR Count (Johnson et al., 2017)	8	63,000	15,000
	smallNORB Elevation (LeCun et al., 2004)	9	24,300	12,150

B.2 Source Models and Compression

Table B.2 lists the source models and compact budgets. For each task and budget, the compact architecture comes from task-guided structured pruning (CarveVFM), which searches layerwise MLP widths and attention configurations under the budget, scores candidates on 512 calibration images. The compact model keeps every transformer layer and the residual width. The same compressed checkpoint is the starting point for every method in a given task and setting, including across calibration budgets in the data-efficiency study.

Table B.2: **Source models and compact budgets** (approximate parameter counts, millions).

Family	Source model	Source	Compact
DINOv3 (Siméoni et al., 2025)	ViT-S/16	22	10
	ViT-S+/16 (data-efficiency study)	29	14
	ViT-B/16	87	35
	ViT-L/16	304	150
	ViT-H+/16	841	300
MetaCLIP 2 (Chuang et al., 2025)	ViT-B/16	86	35
SigLIP 2 (Tschannen et al., 2025)	ViT-B/16	93	35
I-JEPA (Assran et al., 2023)	ViT-H/14	631	250
MAE (He et al., 2022)	ViT-B/16	86	35
AugReg (Steiner et al., 2022)	ViT-B/16	86	35
BiomedCLIP (Zhang et al., 2023)	ViT-B/16	86	35

B.3 BASELINES

All baselines start from the same compressed checkpoint, use the same calibration images and hold-out split, and are evaluated with the same fresh linear classifier (Appendix B.4). Iterative baselines train for twenty epochs with AdamW, cosine learning-rate decay (5% warmup, minimum ratio 0.05), and weight decay 10^{-4}. AdaPrune and block reconstruction minimize their reconstruction losses; the other iterative baselines use the teacher-guided objective of Appendix A.4 (cross-entropy, logit distillation, and final-feature matching). Table B.3 lists what each method updates.

Table B.3: **Baseline configurations.** LR: peak learning rate.

Method	What is trained or fitted	LR
B+LN	Biases and LayerNorm parameters	3×10^{-4}
AdaPrune	Output projections, trained to match teacher block outputs (local reconstruction)	3×10^{-4}
Ridge	Closed-form layerwise ridge reconstruction ($\lambda = 10^{-3}$); no gradient training	–
Neuron merging	Data-free merge of each removed MLP neuron into its most similar survivor (Kim et al., 2020); replaces the MLP compensation on the same pruning mask; not applicable to gated H+ MLPs	–
SSF	Per-feature scale and shift (Lian et al., 2022)	3×10^{-4}
SVFT	Sparse coefficients in the singular-vector basis of each weight (Lingam et al., 2024)	3×10^{-4}
AdaptFormer	Parallel MLP adapters, bottleneck 4 (Chen et al., 2022)	3×10^{-4}
LoRA ($r = 2, 4, 8$)	Low-rank updates of MLP down and attention output projections, all layers (Hu et al., 2022)	3×10^{-4}
DoRA ($r = 8$)	As LoRA, with magnitude–direction decomposition (Liu et al., 2024)	3×10^{-4}
PiSSA ($r = 8$)	As LoRA, initialized from principal singular components (Meng et al., 2024)	3×10^{-4}
Block recon.	All backbone weights, trained to match teacher block outputs	10^{-5}
Full fine-tune	All backbone weights (weight decay 10^{-2})	10^{-5}
Dense remix	Dense channel maps on the MLPs and eligible attention operators (Zhmoginov et al., 2024)	10^{-4}

B.4 TRAINING PROTOCOLS

Classifier. After adaptation, the backbone is frozen and a fresh linear classifier is trained for ten epochs on the full task training set (AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, seed 90,017, unstandardized features); the final epoch is reported. The same classifier protocol is applied to every method, so every gain in Table 1 compares backbones under identical evaluation conditions. For the DINOv3 ViT-H+ scale, both the compressed baseline and *Channel-Remix* use a 40-epoch classifier trained on standardized representations to ensure stable feature convergence. The uncompressed references are linear probes of the source models; for ViT-H+ it is a forty-epoch SGD probe (mean 88.2%), the same reference as in Figure 4. Source classifiers used by the teacher and the linear probes use full task data; the calibration budget N limits backbone adaptation only.

Ensuring fair comparisons. Because different methods update different parameter sets, from per-feature scales (SSF) to full backbone weights (fine-tuning), we tuned the learning rate for each method individually. Methods that update lightweight adapters or projections (LoRA, DoRA, PiSSA, AdaptFormer, SSF, SVFT, AdaPrune, B+LN) use a learning rate of 3×10^{-4}, while methods that update all backbone weights (block reconstruction, full fine-tuning) use 10^{-5}, and dense remix uses 10^{-4}. These rates were selected to give each baseline its best performance.

We also verified that the number of training epochs does not disadvantage any baseline. *Channel-Remix* uses a two-phase schedule of 20 search + 20 refine epochs. All iterative baselines were trained for both 20 and 40 epochs, and we report the better result, ensuring that no baseline is penalized by fewer gradient updates. *Channel-Remix* optimizes all maps with AdamW, learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 1. Equal epoch counts do not imply equal computation: *Channel-Remix*'s search phase evaluates candidate donors and is less expensive per epoch than a full backward pass over all parameters.

Parameter accounting. The trainable-parameter column of Table 1 includes the classifier and averages the same 144 model–task pairs for all methods (ViT-H+ excluded, where comparable accounting is incomplete). For *Channel-Remix* it counts final active coefficients, excluding temporary search allocations and frozen teacher weights. Excluding classifiers, $K{=}5$ averages 179K coefficients against 411K for LoRA-8.

Table B.4: **Paired tests behind Table 1.** For each setting, *Channel-Remix* (K=5) minus a baseline over the 16 tasks: mean difference (pp), tasks where *Channel-Remix* is higher, and the two-sided Wilcoxon signed-rank p-value.

	vs. PiSSA-8			vs. LoRA-8			vs. Full FT			vs. Block rec.		
Setting	Δ	Wins	p	Δ	Wins	p	Δ	Wins	p	Δ	Wins	p
ViT-S	+1.4	16/16	$<10^{-4}$	+2.0	16/16	$<10^{-4}$	+1.7	16/16	$<10^{-4}$	+2.0	16/16	$<10^{-4}$
ViT-S	+0.3	13/16	0.058	+1.2	14/16	0.005	+2.7	16/16	$<10^{-4}$	+3.0	16/16	$<10^{-4}$
ViT-L	+0.3	10/16	0.274	+0.7	11/16	0.130	+1.2	12/16	0.029	+3.9	12/16	0.074
ViT-H+	+2.2	12/16	0.013	+2.6	13/16	0.005	+8.8	13/16	0.005	+2.2	12/16	0.011
MetaCLIP 2	+3.6	16/16	$<10^{-4}$	+3.6	16/16	$<10^{-4}$	+1.9	10/16	0.029	+2.1	13/16	0.011
SigLIP 2	−0.5	8/16	0.562	+0.5	11/16	0.298	−1.4	3/16	0.058	−2.0	3/16	0.011
I-JEPA	+0.2	12/16	0.175	+1.5	15/16	$<10^{-4}$	+1.0	12/16	0.009	+0.6	11/16	0.058
MAE	+1.7	13/16	0.025	+2.3	15/16	0.001	+1.6	11/16	0.051	+1.2	10/16	0.083
AugReg	+1.1	13/16	0.011	+1.3	13/16	0.008	+0.4	9/16	0.404	+0.8	11/16	0.144
BiomedCLIP	+1.6	16/16	$<10^{-4}$	+1.9	15/16	$<10^{-4}$	+1.6	12/16	0.029	+0.2	12/16	0.061
Pooled, all	+1.2	129/160	$<10^{-12}$	+1.8	139/160	$<10^{-18}$	+1.9	114/160	$<10^{-9}$	+1.4	116/160	$<10^{-7}$
Pooled, matched	+1.3	78/96	$<10^{-7}$	+1.9	85/96	$<10^{-12}$	+0.8	57/96	0.009	+0.5	60/96	0.024

B.5 Statistical Significance

With one adaptation seed, we treat tasks as the unit of replication and use paired Wilcoxon signed-rank tests on per-task differences (Table B.4). *Channel-Remix* is higher than PiSSA-8 in 78 of 96 task–setting pairs ($p < 10^{-7}$) and than LoRA-8 in 85 of 96 ($p < 10^{-12}$). The advantage over full fine-tuning and block reconstruction is smaller (57 and 60 of 96 pairs). SigLIP 2 is the main exception: there, *Channel-Remix* trails PiSSA-8 slightly and loses to full fine-tuning and block reconstruction on 13 of 16 tasks, consistent with the discussion of broader updates in Section 4.2 of the main paper. Pooled tests treat task–setting pairs as independent although the tasks repeat across settings, so the per-setting rows are the more conservative evidence.

B.6 Per-Dataset Results

Tables B.5, B.6, B.7 give the per-dataset accuracies behind Table 1. The Mean column reproduces the compressed accuracy of Table 1, and the difference between a method's mean and the compressed mean reproduces its entry in Table 1. Abbreviations: C10/C100 CIFAR-10/100, Cal Caltech-256, Air FGVC-Aircraft, Ind MIT Indoor-67, OrgA OrganAMNIST, Pneu PneumoniaMNIST, Euro EuroSAT, RES RESISC45, sNORB smallNORB Elevation.

B.7 Data Efficiency

Tables B.8 give the per-dataset results behind Figure 4. Both methods start from the same compressed checkpoint, prepared with 512 images, at every calibration budget, so the budgets differ only in the data used for adaptation. The 3–4× ratios in the main text compare nominal calibration budgets at which the mean accuracies of the two methods meet: *Channel-Remix* with 512 images against LoRA-4 with 1,536 (ViT-B) or 2,048 images (ViT-S+ and ViT-H+), i.e., one third to one quarter as many calibration images. On ViT-S+, the match is within 0.07 points.

B.8 Dataset-Level Comparison with LoRA

Figure B.1 resolves the comparison with LoRA-4 at 512 images by dataset and source model. Among the nine settings with matched adaptation schedules (the controlled DINOv3 S+, B, and H+ runs and the six non-DINO families), *Channel-Remix* is higher in 132 of 144 model–dataset pairs.

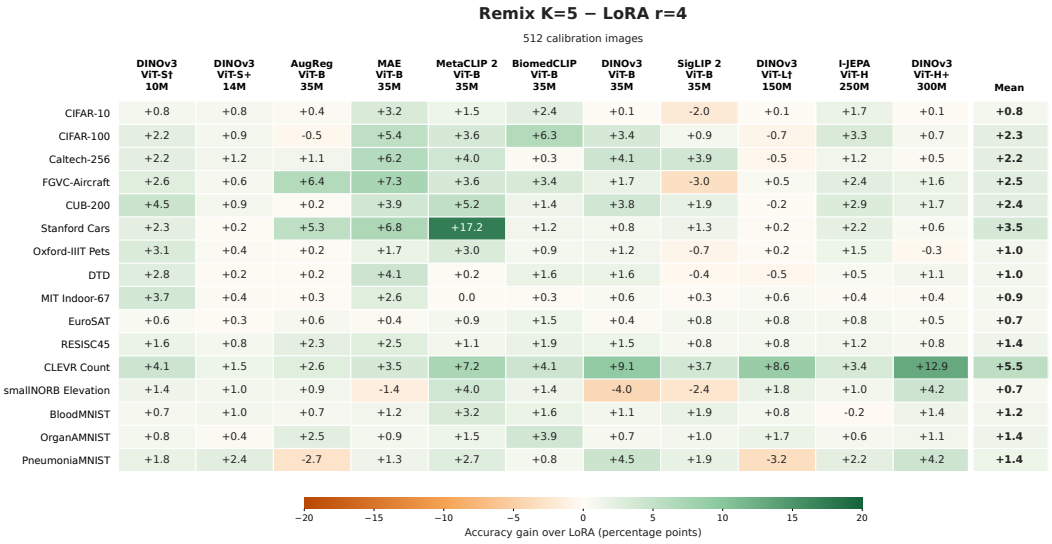


Figure B.1: **Accuracy of *Channel-Remix* (K=5) minus LoRA (r=4) by dataset and source model** (percentage points, 512 calibration images). The DINOv3 S+, B, and H+ columns use the controlled study of Figure 4; the other columns use the main-table runs. The dagger-marked ViT-S and ViT-L columns use forty *Channel-Remix* adaptation epochs versus twenty for LoRA; they are excluded from the 132/144 count.

Table B.5: **DINOv3 backbones** (ViT-S/16→10M, ViT-B/16→35M, ViT-L/16→150M, ViT-H+/16→300M): test accuracy (%) with 512 calibration images. Rows are grouped as Ref. (reference), Recovery (post-pruning recovery), PEFT (parameter-efficient adaptation), Full (full tuning) and Ours (*Channel-Remix*, K sources; shaded). Bold: best per column within each backbone.

		Natural									*Medical*			*Remote*		*Structured*		
	Method	C10	C100	Cal	Air	CUB	Cars	Pets	DTD	Ind	Blood	OrgA	Pneu	Euro	RES	CLEVR	sNORB	**Mean**
DINOv3 ViT-S/16 → 10M params																		
Ref.	Compressed (Comp.)	91.4	72.6	70.2	56.0	61.4	73.8	79.6	65.0	65.1	96.3	92.8	85.7	95.8	85.8	58.6	35.0	74.1
	Uncompressed LP	95.8	83.4	92.2	76.9	77.6	86.2	92.7	60.2	85.7	97.6	93.5	90.4	96.5	90.9	64.7	35.0	82.5
Recovery	B+LN	92.8	77.1	75.5	60.4	69.1	80.4	84.4	70.5	75.7	96.6	93.2	87.3	96.1	86.6	60.2	36.1	77.6
	AdaPrune	90.9	67.9	71.2	53.1	63.3	72.1	78.7	65.1	68.1	95.8	93.1	88.6	95.1	85.0	59.3	35.1	73.9
	Ridge	53.7	20.8	25.1	9.6	5.0	5.2	12.5	35.7	24.0	82.1	82.9	84.6	80.5	51.4	32.6	23.8	39.3
	Neuron merge.	41.3	23.5	28.2	6.3	5.2	4.0	10.7	29.3	23.7	87.0	86.1	82.1	63.9	60.0	42.8	22.1	38.5
PEFT	SSF	92.8	77.2	75.2	60.0	69.0	80.3	84.0	69.8	75.2	96.7	93.6	87.7	96.0	86.7	60.0	35.9	77.5
	SVFT	92.3	76.1	73.5	58.7	65.7	78.6	82.9	67.4	74.0	96.6	92.9	86.2	96.1	85.6	59.6	35.7	76.4
	AdaptFormer	92.1	75.5	72.7	57.5	64.8	77.3	82.5	66.4	71.3	96.5	92.9	85.6	96.0	85.5	59.0	34.8	75.7
	LoRA-2	92.7	76.7	75.1	59.4	67.4	79.3	83.0	68.8	75.0	96.4	93.5	86.7	95.9	86.2	59.5	35.5	76.9
	LoRA-4	92.7	76.9	75.6	59.9	68.3	79.8	83.5	69.4	75.2	96.7	93.6	86.9	96.0	86.4	59.3	35.5	77.2
	LoRA-8	92.7	77.1	75.8	60.2	69.0	79.8	83.9	70.3	74.9	96.6	93.5	87.3	95.9	86.4	59.1	35.5	77.4
	DoRA-8	92.7	77.3	75.5	59.9	69.2	80.1	84.3	70.6	75.0	96.7	93.6	86.5	95.9	86.5	59.3	36.4	77.5
	PiSSA-8	93.0	77.6	76.0	61.0	70.7	80.7	85.5	70.5	76.0	96.6	93.6	87.8	96.0	86.8	59.8	36.4	78.0
Full	Block recon.	93.0	77.7	75.5	60.3	68.9	80.5	84.7	69.4	75.4	96.5	93.5	86.7	95.5	86.6	59.7	35.5	77.5
	Full fine-tune	92.9	77.5	75.9	60.1	70.0	80.8	84.6	70.6	75.4	96.6	93.8	87.2	95.9	86.6	59.7	35.6	77.7
	Dense remix	93.0	78.0	76.6	60.9	73.0	81.8	86.3	70.4	76.0	**97.6**	**95.2**	87.2	96.4	87.8	**65.6**	30.7	78.5
Ours	K=3	**93.5**	78.9	**78.0**	62.1	72.6	81.8	85.1	**72.3**	78.7	97.2	94.3	**88.6**	96.4	88.1	62.6	36.1	79.2
	K=5	93.5	79.1	77.8	**62.6**	72.8	82.1	86.5	72.2	**79.0**	97.4	94.4	**88.6**	96.6	88.0	63.4	**36.8**	79.4
	K=7	93.4	**79.1**	**78.0**	62.4	**73.2**	**82.3**	**86.6**	72.2	78.9	97.3	94.4	**88.6**	**96.7**	**88.4**	63.9	36.4	**79.5**
DINOv3 ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	94.3	66.1	66.4	68.6	67.2	79.6	84.2	71.4	74.1	95.6	93.8	81.9	97.0	86.6	60.6	36.6	76.5
	Uncompressed LP	97.7	87.7	95.9	83.8	82.4	88.5	95.0	62.6	89.4	97.9	93.7	90.5	97.0	93.1	65.7	38.2	85.0
Recovery	B+LN	86.3	79.1	77.9	67.4	72.0	85.1	87.0	74.7	67.7	96.6	93.9	79.6	97.2	89.3	58.0	24.6	77.3
	AdaPrune	91.5	62.3	67.7	68.4	68.6	79.1	82.8	68.0	74.4	97.3	94.4	80.1	96.4	86.3	63.7	38.3	76.2
	Ridge	69.4	26.8	22.1	11.7	9.8	3.5	17.5	40.7	30.1	86.3	90.0	80.8	77.0	63.0	37.0	16.3	42.6
	Neuron merge.	51.8	38.2	42.0	7.6	10.1	7.1	26.6	44.7	41.3	87.4	88.3	68.3	73.5	78.7	39.4	25.1	45.6
PEFT	SSF	95.9	79.0	76.8	75.1	78.7	89.8	90.7	73.6	84.9	96.6	93.8	79.3	97.2	88.6	57.6	38.7	81.0
	SVFT	95.5	75.3	70.3	74.0	76.3	89.3	89.4	72.7	82.8	96.3	93.7	79.5	97.3	87.4	57.7	37.5	79.7
	AdaptFormer	95.5	76.1	70.8	73.7	74.8	88.7	89.8	72.6	81.1	96.1	94.0	80.8	97.1	87.4	57.0	35.6	79.4
	LoRA-2	95.8	79.0	77.7	75.3	77.5	90.1	91.4	73.9	84.7	96.3	94.1	80.8	97.1	88.3	58.7	37.3	81.0
	LoRA-4	96.0	79.5	78.7	75.4	78.6	90.3	91.8	73.9	84.7	96.3	94.0	81.2	97.1	89.1	57.3	37.3	81.3
	LoRA-8	96.0	80.0	79.5	75.1	79.2	90.6	92.0	73.6	84.9	96.3	93.8	81.9	97.2	89.3	58.7	38.5	81.7
	DoRA-8	95.9	79.7	78.2	74.9	79.3	90.5	92.0	73.2	84.4	96.7	93.9	81.6	97.2	89.3	59.5	38.1	81.5
	PiSSA-8	96.1	80.0	78.1	75.2	80.5	90.6	92.2	73.7	85.1	97.0	93.7	**85.1**	97.2	89.1	63.9	**43.3**	82.6
Full	Block recon.	92.0	80.0	76.8	71.3	75.0	88.2	91.2	74.3	78.6	97.1	94.0	79.6	97.2	89.3	65.0	28.4	79.9
	Full fine-tune	93.5	80.4	78.0	71.1	77.1	87.1	91.2	74.1	77.5	97.0	94.0	81.6	97.1	89.3	64.8	29.6	80.2
	Dense remix	90.7	73.9	72.2	67.8	72.5	85.2	85.6	61.9	73.1	96.4	95.3	82.2	96.4	86.7	37.3	18.2	74.7
Ours	K=3	**96.3**	82.1	81.3	76.1	81.9	**91.3**	**93.0**	**75.2**	84.9	97.3	95.1	81.7	97.5	90.0	61.9	28.7	82.1
	K=5	**96.3**	82.1	**82.5**	**76.2**	**82.2**	90.4	92.7	74.9	**85.7**	**97.5**	95.0	83.2	97.4	**90.6**	**66.4**	33.3	**82.9**
	K=7	96.1	**82.8**	81.2	76.1	82.1	90.9	92.8	73.8	84.7	97.4	**95.5**	83.7	**97.6**	90.4	62.0	24.8	82.0
DINOv3 ViT-L/16 → 150M params																		
Ref.	Compressed (Comp.)	97.5	88.8	93.2	80.8	88.5	93.7	95.6	81.8	90.7	95.6	93.7	86.1	97.2	93.6	58.9	37.9	85.9
	Uncompressed LP	98.9	92.3	97.2	84.8	90.6	94.4	96.3	82.7	90.1	98.3	92.9	92.6	97.3	94.0	68.1	35.8	87.9
Recovery	B+LN	97.8	89.9	94.6	80.8	89.1	93.8	95.9	82.5	90.7	97.1	93.4	86.4	97.5	93.7	71.0	37.3	87.0
	AdaPrune	97.2	88.4	93.3	80.1	87.7	93.7	95.6	81.7	89.9	96.1	93.5	82.7	96.7	93.5	63.5	36.0	85.6
	Ridge	48.9	38.4	23.2	7.8	8.4	15.4	19.4	24.5	38.4	82.8	85.1	78.7	77.8	63.8	27.0	27.1	41.7
	Neuron merge.	69.0	47.8	32.1	12.2	8.1	6.6	9.5	21.9	17.4	85.4	83.4	83.5	81.0	19.5	41.8	30.8	40.6
PEFT	SSF	98.5	90.4	94.8	80.9	89.2	93.7	96.1	**82.9**	90.8	97.0	93.9	87.3	97.8	93.5	69.4	39.6	87.2
	SVFT	98.6	90.1	93.9	80.9	88.8	93.7	95.7	82.3	90.9	96.4	93.8	87.7	97.1	93.5	64.6	39.2	86.7
	AdaptFormer	98.4	90.5	94.4	80.9	89.0	93.8	95.8	82.4	90.2	96.6	93.8	87.0	97.3	93.4	60.2	38.3	86.4
	LoRA-2	98.7	90.5	94.8	80.6	89.2	93.9	96.1	82.3	90.4	96.9	93.7	86.9	97.3	93.6	66.8	39.8	87.0
	LoRA-4	98.8	90.5	94.8	80.8	89.2	93.9	96.1	82.5	90.6	96.9	93.8	86.9	97.4	93.6	66.4	39.6	87.0
	LoRA-8	98.8	90.6	94.8	81.0	89.2	93.8	96.0	82.6	90.5	96.9	93.7	86.5	97.3	93.5	67.5	39.8	87.0
	DoRA-8	98.8	90.5	94.9	80.8	89.1	93.8	96.0	82.6	90.7	96.9	93.8	87.3	97.2	93.7	69.9	40.3	87.3
	PiSSA-8	**98.9**	90.7	**95.0**	80.7	89.2	93.7	95.9	82.6	90.4	97.3	94.3	87.3	97.6	93.8	70.5	39.4	87.3
Full	Block recon.	43.6	89.5	94.8	80.5	**89.4**	93.8	96.0	82.4	90.9	97.3	94.1	87.8	97.8	93.7	71.6	38.0	83.8
	Full fine-tune	98.7	90.1	94.9	80.7	89.3	93.8	96.0	82.7	90.7	97.5	94.3	79.8	97.8	93.7	71.2	33.0	86.5
	Dense remix	21.8	2.1	74.0	65.0	76.2	89.3	89.0	49.5	27.1	57.7	81.1	77.2	64.7	83.7	34.2	16.6	56.8
Ours	K=3	98.8	**91.0**	94.4	81.0	89.2	94.0	95.9	82.3	90.4	**98.0**	95.2	**88.9**	98.0	94.1	74.0	40.3	**87.8**
	K=5	98.9	89.9	94.3	**81.3**	89.1	**94.1**	**96.2**	82.0	**91.2**	97.7	95.5	83.7	**98.2**	94.4	75.1	**41.4**	87.7
	K=7	98.7	89.8	94.2	80.8	88.9	94.0	95.7	82.2	90.9	97.8	**95.6**	82.9	98.1	**94.9**	**78.0**	40.2	87.7
DINOv3 ViT-H+/16 → 300M params																		
Ref.	Compressed (Comp.)	98.9	89.0	91.0	79.6	85.7	93.9	96.2	81.8	89.6	98.5	94.9	89.6	97.7	93.7	68.1	36.1	86.5
	Uncompressed LP	99.5	94.5	97.3	83.5	90.6	94.3	96.5	82.4	91.9	98.1	92.6	89.7	97.6	95.1	75.5	31.8	88.2
Recovery	B+LN	36.5	91.4	94.3	72.7	86.6	92.4	96.4	80.8	90.5	93.9	83.8	82.5	97.5	93.4	71.5	38.6	81.4
	AdaPrune	98.8	89.5	91.5	69.7	85.2	92.5	95.9	81.2	89.4	93.7	93.9	84.3	97.5	93.9	65.5	36.5	84.9
	Ridge	37.8	37.5	14.3	4.3	5.8	2.0	9.9	38.3	26.3	59.5	70.4	78.4	81.4	38.1	29.1	13.6	34.2
	Neuron merge	29.1	46.7	19.8	2.8	6.0	1.5	8.5	34.2	26.0	61.4	69.1	76.1	77.7	44.5	38.2	15.5	34.8
PEFT	SSF	**99.4**	91.2	93.6	72.5	86.2	92.1	96.3	80.1	90.3	93.3	92.9	83.5	97.5	93.9	69.2	37.9	85.6
	SVFT	99.2	90.1	91.4	71.0	85.3	92.5	96.1	80.6	89.6	92.5	92.8	81.2	97.5	93.6	67.3	36.7	84.8
	AdaptFormer	99.2	90.1	92.4	71.8	85.2	92.2	96.2	79.9	90.1	92.3	92.7	81.1	97.5	93.7	66.8	36.1	84.8
	LoRA-2	99.3	91.1	93.8	72.0	85.8	92.3	96.5	80.6	90.2	93.1	92.8	82.4	97.6	94.1	67.7	37.1	85.4
	LoRA-4	99.3	91.2	94.0	72.3	85.8	92.3	96.5	81.0	90.1	93.3	92.8	82.7	97.6	94.2	68.0	37.4	85.5
	LoRA-8	99.2	91.4	94.1	72.3	86.8	92.3	96.4	80.6	90.2	93.5	92.9	83.8	97.6	94.3	69.6	36.7	85.7
	DoRA-8	99.3	91.1	93.9	72.0	85.8	92.2	96.4	80.5	90.1	93.4	92.8	82.1	97.5	94.2	71.4	37.9	85.7
	PiSSA-8	**99.4**	91.7	**94.4**	72.0	87.1	92.4	96.3	81.0	90.4	93.9	93.1	83.0	97.7	94.2	73.7	38.2	86.1
Full	Block recon.	99.3	91.8	94.2	71.8	87.3	92.4	96.3	80.2	90.4	94.2	93.1	81.6	97.8	94.1	74.3	38.7	86.1
	Full fine-tune	42.7	**91.9**	94.2	71.3	87.3	**92.5**	96.3	80.6	90.0	94.4	86.4	81.4	97.8	94.2	33.8	38.2	79.6
	Dense remix	36.5	17.5	68.4	2.8	55.9	82.9	4.8	26.0	5.4	80.6	82.5	74.2	73.6	41.4	35.8	19.2	44.2
Ours	K=3	99.2	90.4	93.8	**80.5**	**88.0**	91.6	**96.6**	82.4	90.3	98.7	**95.4**	90.7	97.9	94.7	76.8	41.4	88.0
	K=5	99.2	90.8	94.2	80.1	87.5	91.4	96.5	**82.7**	**90.7**	98.7	**95.4**	90.7	98.1	95.0	80.9	41.5	88.3
	K=7	99.1	90.6	94.1	80.5	87.8	92.5	96.5	82.3	90.5	**98.9**	94.9	**91.0**	**98.3**	**95.1**	**82.6**	**42.0**	**88.5**

Table B.6: **Vision–language backbones** (MetaCLIP 2, SigLIP 2 and BiomedCLIP ViT-B/16, each compressed to 35M): test accuracy (%) with 512 calibration images. Rows are grouped as Ref. (reference), Recovery (post-pruning recovery), PEFT (parameter-efficient adaptation), Full (full tuning) and Ours (*Channel-Remix*, K sources; shaded). Bold: best per column within each backbone

		Natural									*Medical*			*Remote*		*Structured*		
	Method	C10	C100	Cal	Air	CUB	Cars	Pets	DTD	Ind	Blood	OrgA	Pneu	Euro	RES	CLEVR	sNORB	**Mean**
MetaCLIP 2 ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	90.7	68.6	65.6	39.6	50.4	16.5	72.6	67.4	72.9	93.4	91.7	84.1	96.4	87.4	57.7	35.5	68.2
Ref.	Uncompressed LP	97.4	85.3	95.6	69.6	79.9	91.6	92.0	80.1	92.0	95.8	88.8	85.4	96.4	93.3	69.1	41.5	84.6
Recovery	B+LN	92.0	71.7	68.0	43.6	56.2	31.5	78.0	70.6	76.5	93.7	92.0	88.1	96.3	88.5	57.9	36.5	71.3
Recovery	AdaPrune	86.2	64.1	55.2	23.6	37.1	3.8	57.6	57.1	63.4	93.3	92.1	84.5	95.4	82.4	58.0	29.5	61.5
Recovery	Ridge	65.5	48.3	31.2	7.7	5.3	2.6	17.7	35.5	34.5	72.2	90.4	84.8	94.1	75.7	46.6	24.9	46.1
Recovery	Neuron merge.	52.6	20.8	27.4	7.7	5.6	4.0	12.0	38.4	30.6	84.2	85.4	81.2	88.4	63.7	38.4	33.0	42.1
PEFT	SSF	91.9	71.6	67.9	43.4	55.9	27.6	78.0	70.0	76.2	93.0	91.5	87.3	96.2	88.1	56.9	36.1	70.7
PEFT	SVFT	91.7	71.1	67.4	43.4	56.1	26.9	77.8	70.4	75.6	93.0	91.9	85.3	96.4	88.5	55.9	35.8	70.4
PEFT	AdaptFormer	91.9	71.8	68.1	42.4	55.5	25.8	77.4	70.4	76.4	93.0	91.7	86.1	96.4	88.1	56.4	34.3	70.4
PEFT	LoRA-2	92.0	71.9	68.1	44.5	56.6	29.4	77.8	72.1	76.5	93.0	91.8	86.5	96.4	88.7	56.7	35.0	71.1
PEFT	LoRA-4	91.8	71.9	68.5	44.5	57.1	31.2	77.9	72.0	76.2	93.0	91.9	87.0	96.7	88.7	57.3	35.7	71.3
PEFT	LoRA-8	92.0	72.3	68.5	45.1	57.2	32.1	78.3	71.8	75.8	93.1	91.6	85.4	96.6	88.5	57.6	35.8	71.4
PEFT	DoRA-8	91.9	71.0	67.8	45.4	56.2	32.6	78.9	71.0	76.3	93.3	91.3	87.0	96.4	88.1	58.2	36.4	71.4
PEFT	PiSSA-8	92.8	72.0	67.8	44.0	56.4	30.8	79.0	70.2	75.7	92.9	91.9	87.5	96.7	87.8	58.8	38.4	71.4
Full	Block recon.	**93.6**	74.7	69.9	47.6	59.9	38.6	80.7	**72.8**	**78.0**	94.9	93.1	83.3	97.0	89.3	61.8	31.5	72.9
Full	Full fine-tune	93.3	74.9	70.8	**48.2**	60.8	40.0	81.2	72.7	77.6	95.3	93.5	83.3	97.1	88.9	61.5	29.7	73.1
Full	Dense remix	58.7	59.9	51.6	32.6	47.0	5.5	29.1	50.1	47.5	91.7	89.4	81.4	89.2	61.2	47.1	22.8	54.0
Ours	$K=3$	93.5	**75.6**	**72.5**	45.8	61.8	47.4	**81.6**	72.7	76.6	95.9	93.6	89.6	97.3	89.3	63.0	**40.9**	74.8
Ours	$K=5$	93.2	75.5	72.4	48.1	**62.3**	**48.4**	80.9	72.2	76.2	96.1	93.4	89.7	**97.5**	**89.7**	64.5	39.7	**75.0**
Ours	$K=7$	92.8	75.2	72.1	47.5	62.2	47.6	81.2	72.3	76.0	**96.4**	**93.6**	**90.1**	97.5	89.3	**64.9**	39.6	74.9
SigLIP 2 ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	88.2	63.7	63.9	47.9	58.8	74.7	79.9	70.6	73.1	93.2	90.1	85.1	94.4	87.0	60.5	35.3	72.9
Ref.	Uncompressed LP	96.6	84.2	97.4	78.3	82.4	94.9	94.8	81.7	93.4	94.6	89.1	86.1	95.8	94.9	70.0	36.2	85.6
Recovery	B+LN	90.6	69.4	70.6	54.8	62.7	78.7	83.3	72.3	77.1	94.5	91.2	87.3	95.4	89.0	67.3	39.7	76.5
Recovery	AdaPrune	83.9	58.7	52.8	34.2	35.2	49.8	67.0	53.1	55.1	93.1	89.3	83.0	94.9	78.4	57.2	36.2	63.9
Recovery	Ridge	67.2	24.3	13.0	1.9	1.7	2.3	14.8	33.0	17.7	67.5	73.7	77.9	80.7	58.2	40.5	20.9	37.2
Recovery	Neuron merge.	42.5	7.4	4.5	3.2	1.2	4.1	5.5	5.3	3.0	58.0	74.6	70.8	72.5	23.6	28.2	25.0	26.8
PEFT	SSF	90.4	69.0	69.8	54.4	63.0	78.6	83.3	72.0	77.5	94.2	90.3	85.7	95.0	88.9	65.3	39.6	76.1
PEFT	SVFT	90.1	68.4	69.5	54.2	63.5	79.2	83.5	72.8	77.4	94.3	90.4	85.7	94.9	89.4	64.8	40.1	76.1
PEFT	AdaptFormer	90.0	68.8	70.5	55.7	63.3	79.7	83.3	72.3	76.6	94.3	90.0	85.4	95.0	89.1	63.2	39.1	76.0
PEFT	LoRA-2	90.3	68.6	71.1	54.3	63.5	78.9	83.7	72.2	78.4	94.4	89.8	87.0	95.1	89.2	63.1	38.7	76.1
PEFT	LoRA-4	90.4	68.8	70.5	53.8	63.4	79.0	83.8	72.7	78.4	94.3	90.2	86.2	94.8	89.5	64.1	39.4	76.2
PEFT	LoRA-8	90.1	68.6	70.3	54.2	63.4	79.5	84.1	72.6	77.8	94.5	90.5	86.1	95.0	89.5	65.3	39.6	76.3
PEFT	DoRA-8	90.3	68.6	70.0	54.3	63.7	79.7	83.8	72.8	78.5	94.3	90.2	86.9	94.9	89.5	65.4	40.8	76.5
PEFT	PiSSA-8	91.2	70.2	72.2	54.2	64.4	79.3	84.7	73.8	78.2	94.9	91.2	86.7	95.4	90.2	68.6	**42.4**	77.3
Full	Block recon.	**91.4**	71.7	**77.0**	58.5	**68.4**	**83.4**	**87.3**	**74.8**	**81.6**	95.6	83.5	87.8	**95.9**	**91.3**	**73.4**	38.9	**78.8**
Full	Full fine-tune	91.4	**71.8**	76.6	**58.6**	68.3	83.0	87.0	74.7	81.3	95.5	82.6	**88.9**	95.8	91.1	73.1	32.4	78.3
Full	Dense remix	46.0	27.2	31.2	5.0	26.9	1.1	29.2	41.8	21.1	91.3	74.9	81.6	85.7	63.2	43.6	17.2	42.9
Ours	$K=3$	90.3	70.4	74.2	52.8	66.0	81.2	83.9	73.9	78.1	95.8	**91.6**	87.2	95.6	90.6	69.3	40.7	77.6
Ours	$K=5$	88.4	69.7	74.4	50.8	65.3	80.3	83.1	72.3	78.7	**96.2**	91.2	88.1	95.6	90.2	67.8	37.0	76.8
Ours	$K=7$	87.9	68.9	73.1	49.5	64.9	80.0	83.1	73.0	75.4	94.9	91.3	86.7	95.6	89.2	68.3	36.6	76.2
BiomedCLIP ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	63.9	13.5	49.8	19.0	21.8	9.8	35.5	48.3	42.6	93.7	86.8	92.9	90.2	77.2	54.4	21.9	51.3
Ref.	Uncompressed LP	64.2	42.6	50.6	20.0	21.1	10.3	37.1	48.4	42.8	93.4	86.7	94.2	89.2	76.8	53.2	21.7	53.3
Recovery	B+LN	65.4	4.4	49.6	21.4	22.6	11.1	38.7	49.8	42.5	94.7	87.3	93.1	90.4	78.5	57.3	23.1	51.9
Recovery	AdaPrune	64.4	16.5	47.5	18.8	22.2	9.6	36.9	47.7	41.3	92.8	88.0	94.2	90.4	77.0	53.5	22.4	51.5
Recovery	Ridge	64.3	14.9	46.2	7.2	20.2	5.7	38.9	45.1	41.1	93.4	82.1	90.7	87.7	75.5	47.5	20.1	48.8
Recovery	Neuron merge.	51.3	11.0	30.1	12.4	12.6	4.5	19.7	35.5	24.3	85.4	86.6	85.6	81.1	63.3	44.6	21.7	41.9
PEFT	SSF	65.7	5.2	49.3	21.0	23.4	11.2	39.1	49.9	44.1	94.2	87.4	92.8	90.8	78.4	57.0	23.5	52.1
PEFT	SVFT	65.7	15.9	48.9	20.2	22.8	10.8	37.7	49.1	45.1	93.9	86.5	93.1	90.5	77.3	56.2	22.9	52.3
PEFT	AdaptFormer	66.4	17.1	50.0	20.9	23.5	11.2	38.5	50.5	44.6	93.8	87.0	92.9	90.7	78.2	57.0	22.9	52.8
PEFT	LoRA-2	65.5	16.5	49.6	20.0	23.0	10.6	39.5	49.1	44.9	94.0	86.8	92.9	90.8	78.1	56.5	23.7	52.6
PEFT	LoRA-4	65.7	16.8	49.5	20.6	23.2	10.9	39.5	50.0	44.2	94.0	87.0	93.3	90.8	77.8	56.6	23.4	52.7
PEFT	LoRA-8	65.6	17.5	49.6	20.9	23.2	11.0	39.2	50.0	**45.1**	94.1	86.9	93.3	90.7	77.8	56.5	23.6	52.8
PEFT	DoRA-8	66.2	17.8	49.1	20.9	23.4	11.5	39.2	50.4	44.6	93.8	87.2	93.6	90.7	78.6	57.0	23.6	53.0
PEFT	PiSSA-8	66.5	19.2	49.8	20.3	23.2	11.1	38.8	50.1	44.5	94.6	87.8	93.8	91.3	78.6	58.0	23.6	53.2
Full	Block recon.	68.8	**29.8**	50.1	22.1	23.7	11.2	39.7	51.3	44.4	95.0	89.7	91.8	92.3	79.5	60.5	23.6	54.6
Full	Full fine-tune	67.4	3.9	**50.6**	23.9	24.5	11.7	39.8	**51.7**	45.1	95.7	90.5	91.2	92.2	79.5	59.7	23.7	53.2
Full	Dense remix	**69.3**	7.1	43.5	9.0	25.0	6.7	**42.2**	46.2	41.1	**96.6**	81.4	84.9	83.7	75.3	40.7	18.2	48.2
Ours	$K=3$	67.6	21.4	50.5	22.7	24.5	12.0	39.9	50.6	44.3	95.7	89.9	93.6	91.9	80.0	59.2	24.5	54.3
Ours	$K=5$	68.1	23.1	49.9	24.0	24.6	12.2	40.4	51.6	44.5	95.6	90.9	94.1	92.3	79.7	60.7	24.7	54.8
Ours	$K=7$	68.5	23.6	50.3	**24.3**	**25.2**	**12.7**	41.4	51.1	44.0	96.3	**91.9**	**94.7**	**93.0**	**80.6**	**61.2**	**25.3**	**55.3**

Table B.7: **Self-supervised and supervised backbones** (I-JEPA ViT-H/14→250M, MAE ViT-B/16→35M, AugReg ViT-B/16→35M): test accuracy (%) with 512 calibration images. All methods: 20 adaptation epochs. Rows are grouped as Ref. (reference), Recovery (post-pruning recovery), PEFT (parameter-efficient adaptation), Full (full tuning) and Ours (*Channel-Remix*, K sources; shaded). Bold: best per column within each backbone.

		Natural									*Medical*			*Remote*		*Structured*		
	Method	C10	C100	Cal	Air	CUB	Cars	Pets	DTD	Ind	Blood	OrgA	Pneu	Euro	RES	CLEVR	sNORB	**Mean**
I-JEPA ViT-H/14 → 250M params																		
Ref.	Compressed (Comp.)	90.6	71.2	80.3	27.2	39.4	23.6	86.7	64.0	72.7	97.7	92.9	88.1	96.5	85.5	63.4	40.7	70.0
Ref.	Uncompressed LP	91.7	74.1	83.5	27.0	37.8	21.2	87.3	63.4	72.7	98.1	92.8	86.5	96.3	87.4	64.7	39.7	70.3
Recovery	B+LN	92.2	74.1	81.3	39.2	43.8	29.0	87.7	64.7	73.1	97.7	93.2	87.7	96.4	85.5	67.2	40.7	72.1
Recovery	AdaPrune	89.7	66.8	78.7	26.3	36.2	20.7	84.5	62.7	70.5	96.8	92.7	87.8	96.5	84.3	62.5	41.2	68.6
Recovery	Ridge	86.8	63.6	71.4	27.2	30.1	19.1	81.8	61.3	70.2	94.5	92.5	88.3	96.3	80.7	55.1	36.1	65.9
Recovery	Neuron merge.	27.4	1.6	4.7	1.0	0.4	0.4	2.8	2.7	1.9	88.2	27.0	62.7	24.4	5.2	25.6	15.2	18.2
PEFT	SSF	92.8	74.5	80.9	40.9	**45.9**	32.4	87.6	64.5	73.1	98.0	93.3	87.2	96.6	87.3	68.7	41.7	72.8
PEFT	SVFT	92.1	73.1	81.0	37.3	43.5	31.0	87.0	64.5	73.2	97.8	93.1	88.1	96.4	86.8	67.8	42.5	72.2
PEFT	AdaptFormer	92.0	73.4	81.1	41.8	45.3	32.3	87.0	64.4	73.0	97.9	93.0	87.5	96.2	87.0	68.2	41.8	72.6
PEFT	LoRA-2	91.8	72.8	80.9	39.8	41.8	29.3	86.8	64.5	73.1	97.9	92.9	87.5	96.3	87.4	68.0	42.2	72.1
PEFT	LoRA-4	91.8	72.7	81.1	40.9	41.9	30.1	86.9	63.9	73.1	98.0	92.9	87.8	96.3	87.3	68.3	42.4	72.2
PEFT	LoRA-8	91.6	73.2	80.8	41.6	41.8	30.7	86.9	64.0	72.8	97.9	93.0	87.5	96.4	87.6	68.0	42.3	72.3
PEFT	DoRA-8	91.8	72.9	80.9	**45.5**	44.6	34.8	87.1	64.0	73.3	97.7	93.1	88.0	96.4	87.4	66.9	42.3	72.9
PEFT	PiSSA-8	92.5	74.1	80.9	45.0	45.0	**35.1**	88.1	64.2	73.1	**98.2**	93.4	89.4	96.8	87.3	71.2	42.3	73.5
Full	Block recon.	92.0	73.3	81.1	43.6	44.2	33.7	88.0	64.4	73.3	98.0	93.8	90.4	96.4	87.5	70.2	41.6	73.2
Full	Full fine-tune	92.8	74.0	81.5	41.8	44.1	31.6	87.8	64.5	72.2	98.0	**94.0**	**91.7**	96.3	84.7	70.4	39.8	72.8
Full	Dense remix	10.0	1.0	2.4	1.0	0.5	0.5	2.7	2.1	1.4	19.6	29.5	62.5	26.7	4.6	12.7	13.2	11.9
Ours	K=3	93.3	74.6	81.4	41.3	44.0	31.1	87.8	64.6	72.4	98.1	93.4	89.7	97.1	88.4	71.2	43.2	73.2
Ours	K=5	93.5	**76.0**	**82.3**	43.4	44.8	32.3	**88.4**	64.4	**73.4**	97.8	93.6	90.1	**97.1**	**88.6**	71.7	**43.3**	73.8
Ours	K=7	**94.3**	75.7	81.7	43.4	45.3	32.5	88.2	**65.0**	**73.4**	98.0	93.9	89.4	97.0	88.2	**72.1**	43.2	**73.8**
MAE ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	75.7	50.4	49.3	15.6	16.4	9.2	49.9	42.0	45.0	92.3	89.8	87.5	92.4	77.7	63.2	35.1	55.7
Ref.	Uncompressed LP	82.2	57.4	70.7	18.8	22.1	11.3	62.5	53.5	54.4	91.6	90.2	87.7	92.6	81.2	63.0	36.3	61.0
Recovery	B+LN	77.3	52.8	53.8	27.4	31.5	14.0	72.9	50.7	51.6	92.0	90.2	89.4	92.2	76.5	51.7	38.1	60.1
Recovery	AdaPrune	73.5	41.5	40.1	14.4	15.8	8.7	32.9	41.5	41.3	92.6	89.6	88.3	91.2	75.6	60.7	34.8	52.7
Recovery	Ridge	70.3	41.5	34.1	12.4	12.0	5.6	24.2	37.8	36.6	91.1	89.0	87.7	89.2	74.2	52.1	30.1	49.2
Recovery	Neuron merge.	43.9	21.1	11.3	8.6	5.5	4.3	11.4	26.2	24.1	85.4	75.3	76.1	74.4	51.2	38.3	22.2	36.2
PEFT	SSF	76.9	52.3	53.1	25.5	29.3	13.9	72.4	49.8	51.1	93.6	89.5	88.3	92.6	75.8	50.6	38.7	59.6
PEFT	SVFT	76.4	51.1	50.0	20.6	25.4	12.0	67.3	47.1	48.1	92.3	89.3	88.3	92.2	76.0	51.6	37.4	57.8
PEFT	AdaptFormer	75.9	50.6	48.6	19.9	26.7	11.0	69.3	46.6	47.1	91.6	89.3	87.8	92.0	74.7	57.2	35.1	57.7
PEFT	LoRA-2	76.3	51.6	52.4	29.5	33.5	14.8	73.5	49.9	51.6	92.8	89.5	88.1	92.4	75.5	50.8	36.3	59.9
PEFT	LoRA-4	76.5	51.8	53.7	32.2	34.6	15.8	74.4	51.1	52.6	93.1	89.5	87.8	92.5	75.9	52.2	37.2	60.7
PEFT	LoRA-8	76.4	52.7	54.9	34.6	35.7	16.5	75.3	51.4	53.4	93.5	89.5	88.1	92.7	76.3	54.2	38.0	61.4
PEFT	DoRA-8	76.7	52.7	55.1	33.6	35.6	16.1	74.8	51.8	53.1	93.2	89.5	88.8	92.7	76.4	53.5	39.1	61.4
PEFT	PiSSA-8	78.3	53.9	55.6	35.7	34.2	16.7	72.2	49.9	52.8	94.2	90.1	88.9	93.0	77.6	59.7	**40.1**	62.1
Full	Block recon.	79.0	54.2	56.8	37.9	34.2	15.9	75.1	50.6	52.2	94.5	89.9	89.6	92.9	78.4	61.1	38.4	62.6
Full	Full fine-tune	78.9	54.2	56.1	36.0	33.3	15.8	73.9	50.9	51.7	93.9	89.9	**90.5**	**93.3**	78.5	62.8	35.9	62.2
Full	Dense remix	45.9	37.9	39.2	23.6	31.0	11.5	55.5	37.2	38.4	94.7	88.4	83.2	80.2	66.3	34.2	21.7	49.3
Ours	K=3	79.5	57.3	59.0	39.4	38.2	**22.7**	75.9	54.6	55.0	94.5	**90.5**	88.8	93.0	79.2	56.3	37.4	63.8
Ours	K=5	**79.7**	57.2	59.9	**39.5**	38.5	22.5	**76.1**	**55.2**	**55.2**	94.3	90.4	89.1	92.9	78.4	55.7	35.8	63.8
Ours	K=7	78.2	**57.7**	**59.9**	39.3	**38.7**	22.3	76.0	54.2	54.9	**94.7**	90.3	89.6	92.9	**79.5**	**62.9**	33.9	**64.1**
AugReg ViT-B/16 → 35M params																		
Ref.	Compressed (Comp.)	95.9	79.8	82.8	53.6	84.3	60.7	92.0	72.6	82.8	97.5	92.7	85.7	96.7	90.4	63.3	41.1	79.5
Ref.	Uncompressed LP	97.4	86.6	93.3	54.1	87.1	62.9	93.0	76.3	87.0	97.9	91.0	83.8	97.1	90.3	63.2	38.9	81.2
Recovery	B+LN	95.7	80.0	83.6	52.5	84.7	58.6	92.1	74.0	81.6	97.7	92.0	85.7	96.4	88.7	63.7	39.9	79.2
Recovery	AdaPrune	93.2	75.9	75.5	49.7	80.4	50.5	89.3	69.5	77.1	97.6	93.5	86.9	96.7	89.1	62.0	40.2	76.7
Recovery	Ridge	90.6	69.4	63.9	42.7	73.7	30.9	81.9	61.4	71.1	95.1	90.2	83.0	96.3	85.5	53.1	36.9	70.4
Recovery	Neuron merge.	30.9	4.7	6.5	1.4	1.6	0.6	3.9	6.2	2.6	30.3	43.4	85.3	51.8	16.4	25.9	17.4	20.5
PEFT	SSF	95.5	79.8	84.3	52.1	84.8	58.7	92.0	73.8	81.5	97.6	91.9	85.3	96.6	89.9	63.0	40.0	79.2
PEFT	SVFT	95.9	80.1	84.5	51.8	85.1	59.2	92.5	73.2	82.7	97.5	91.8	85.4	96.7	89.4	62.1	40.4	79.3
PEFT	AdaptFormer	95.9	80.6	83.4	51.5	85.0	59.2	92.0	**74.4**	82.2	97.7	91.8	85.7	96.7	89.5	62.2	40.4	79.3
PEFT	LoRA-2	96.0	80.9	84.7	53.2	85.1	60.6	92.3	73.7	82.3	97.5	92.5	85.7	96.6	89.1	62.5	40.4	79.6
PEFT	LoRA-4	95.8	81.1	83.8	53.0	85.2	61.3	92.3	74.0	82.5	97.5	91.9	86.2	96.5	89.5	63.0	40.4	79.6
PEFT	LoRA-8	95.9	80.9	84.8	53.6	85.1	61.4	92.3	73.6	82.8	97.5	91.7	84.9	96.7	89.1	63.5	40.6	79.6
PEFT	DoRA-8	95.6	81.1	84.5	53.0	84.9	59.8	92.5	73.6	81.9	97.5	92.8	**87.5**	96.4	90.1	62.7	40.5	79.7
PEFT	PiSSA-8	96.1	80.8	84.4	53.5	85.3	60.7	92.5	73.5	82.3	**98.2**	93.1	85.9	96.9	91.1	64.2	38.9	79.8
Full	Block recon.	96.1	82.0	**85.6**	54.2	85.5	61.6	**92.7**	73.7	82.2	98.1	93.3	85.3	97.0	90.8	64.8	39.0	80.1
Full	Full fine-tune	96.4	**82.4**	85.2	55.5	85.5	64.0	92.3	73.8	82.7	98.0	92.9	86.4	97.2	91.0	**66.6**	39.1	80.6
Full	Dense remix	87.9	59.6	38.3	51.1	72.2	54.0	82.2	55.3	38.4	97.5	91.8	84.0	95.7	85.8	56.8	20.0	66.9
Ours	K=3	**96.4**	81.3	84.9	57.8	**85.5**	65.5	92.6	73.8	**83.1**	97.4	94.0	86.4	97.1	91.2	64.7	40.7	80.8
Ours	K=5	96.2	80.6	84.9	**59.3**	85.4	**66.7**	92.6	74.3	82.8	98.2	**94.4**	83.5	97.2	**91.8**	65.6	**41.3**	**80.9**
Ours	K=7	96.0	80.4	83.9	58.5	84.6	66.1	92.6	73.9	81.9	97.7	93.7	83.3	**97.2**	91.5	65.5	40.6	80.4

Table B.8: **Data efficiency on DINOv3 backbones.** Test accuracy (%) of *Channel-Remix* (K=5) and LoRA (r=4) for each calibration budget N, with matched 20 adaptation and 10 classifier epochs. Comp. and Uncomp. are linear probes of the compressed and uncompressed backbones. DTD provides 1,880 images at the nominal 2,048 budget.

		Channel-Remix (K=5)					LoRA (r=4)					
Dataset	Comp.	256	512	1,024	1,536	2,048	256	512	1,024	1,536	2,048	Uncomp.
DINOv3 ViT-S+ (29M → 14M)												
CIFAR-10	94.5	95.6	95.6	96.0	96.2	96.0	94.9	94.8	95.3	95.2	95.3	96.8
CIFAR-100	78.2	80.7	81.6	82.0	82.3	82.3	80.2	80.7	81.4	81.3	81.5	85.7
Caltech-256	81.1	83.1	83.7	84.5	85.0	84.7	82.2	82.5	83.4	83.5	83.9	93.5
FGVC-Aircraft	69.2	70.1	70.6	71.4	71.6	72.2	70.1	70.0	70.7	71.3	71.0	78.5
CUB-200	76.7	78.6	79.0	79.3	79.8	79.6	77.8	78.0	78.6	78.8	78.7	86.3
Stanford Cars	87.3	87.8	87.9	88.6	88.8	89.2	87.5	87.7	88.0	88.0	88.2	92.4
Oxford-IIIT Pets	91.1	92.0	92.3	92.6	92.4	92.7	91.8	91.9	92.2	91.9	92.2	94.6
DTD	74.1	75.6	75.5	76.9	77.7	77.4	74.7	75.4	76.0	76.0	76.3	79.8
MIT Indoor-67	80.9	82.1	82.2	82.2	83.1	82.7	81.7	81.8	81.6	81.7	82.2	88.2
BloodMNIST	95.0	96.3	96.7	97.0	97.0	97.2	95.1	95.7	96.3	96.6	96.6	97.2
OrganAMNIST	94.0	93.9	94.3	94.3	94.2	94.4	94.0	93.9	94.0	94.0	94.0	92.4
PneumoniaMNIST	86.2	87.2	88.0	88.3	88.8	88.8	85.9	85.6	86.2	85.9	85.6	87.8
EuroSAT	96.9	97.0	97.1	97.1	97.3	97.4	97.0	96.9	97.1	97.0	97.0	96.2
RESISC45	88.3	89.8	89.7	90.5	90.9	91.0	88.4	88.9	89.3	90.3	90.4	91.0
CLEVR Count	66.3	66.2	67.0	68.0	68.2	70.4	65.9	65.5	66.1	66.2	67.1	64.8
smallNORB Elev.	33.9	34.3	35.1	37.6	39.4	39.7	34.0	34.1	35.9	36.1	37.3	34.5
Mean	80.9	81.9	82.3	82.9	83.3	83.5	81.3	81.5	82.0	82.1	82.3	85.0
DINOv3 ViT-B (86M → 35M)												
CIFAR-10	94.4	96.2	96.1	96.3	96.7	96.9	95.5	96.0	95.9	96.2	96.3	97.8
CIFAR-100	66.1	81.5	83.0	83.9	84.1	84.2	78.1	79.5	80.9	81.8	81.7	88.0
Caltech-256	66.4	81.0	82.7	84.6	85.1	85.2	75.6	78.7	80.4	81.3	82.0	95.9
FGVC-Aircraft	68.6	76.5	77.2	77.6	78.5	79.2	74.6	75.4	75.5	76.0	76.3	83.9
CUB-200	67.4	81.0	82.4	82.7	83.1	83.6	76.8	78.6	80.3	81.2	82.1	88.9
Stanford Cars	79.7	90.9	91.1	91.6	91.7	91.6	89.6	90.3	90.9	91.0	91.0	94.0
Oxford-IIIT Pets	84.3	92.5	93.0	93.7	93.5	93.8	90.7	91.8	92.5	92.4	92.8	96.4
DTD	71.4	74.6	75.6	76.8	75.6	77.9	72.4	73.9	74.6	75.4	76.3	81.9
MIT Indoor-67	74.0	85.1	85.3	86.3	86.3	86.0	83.8	84.7	84.9	85.5	85.6	89.3
BloodMNIST	95.6	97.0	97.4	97.7	98.0	98.1	96.3	96.3	97.0	97.0	97.1	97.3
OrganAMNIST	93.8	95.0	94.7	94.7	94.8	94.9	94.2	94.0	93.9	94.1	93.9	93.2
PneumoniaMNIST	81.7	81.6	85.7	85.4	86.5	84.9	81.6	81.2	80.6	81.9	83.3	86.9
EuroSAT	97.0	97.4	97.4	97.9	97.8	97.9	97.3	97.1	97.2	97.5	97.5	97.0
RESISC45	86.6	89.2	90.6	91.1	91.1	91.3	88.1	89.1	89.6	90.3	90.3	93.1
CLEVR Count	60.6	61.0	66.4	78.6	81.4	81.2	57.5	57.3	60.2	65.0	69.4	65.7
smallNORB Elev.	36.7	26.9	33.3	34.0	30.7	30.8	38.1	37.3	40.6	40.3	42.5	38.2
Mean	76.5	81.7	83.2	84.6	84.7	84.8	80.6	81.3	82.2	82.9	83.6	86.7
DINOv3 ViT-H+ (841M → 300M)												
CIFAR-10	99.0	99.3	99.3	99.4	99.4	99.4	99.3	99.3	99.3	99.3	99.4	99.5
CIFAR-100	89.5	91.7	92.0	92.1	92.1	92.2	91.2	91.4	91.7	91.9	92.2	94.5
Caltech-256	90.7	94.1	94.5	94.8	95.1	95.1	93.8	94.1	94.1	94.6	94.8	97.3
FGVC-Aircraft	71.8	72.9	75.5	78.1	79.7	80.3	73.1	73.9	75.1	76.1	76.3	83.5
CUB-200	84.8	88.0	88.0	88.0	88.3	88.1	86.1	86.3	87.3	87.4	87.2	90.6
Stanford Cars	92.8	93.1	93.0	93.1	93.1	93.5	92.5	92.5	92.4	92.3	92.3	94.3
Oxford-IIIT Pets	96.1	96.3	96.1	96.4	96.6	96.8	96.4	96.4	96.6	96.6	96.6	96.5
DTD	80.4	81.3	82.1	82.1	83.3	83.5	80.9	81.1	81.9	81.8	82.0	82.4
MIT Indoor-67	89.7	90.3	90.6	90.2	90.4	89.9	90.3	90.1	90.1	90.1	90.3	91.9
BloodMNIST	93.6	97.0	97.0	97.6	97.7	97.9	95.8	95.7	96.5	96.6	97.0	98.1
OrganAMNIST	93.3	94.3	94.2	94.3	94.8	94.7	93.0	93.1	92.7	92.9	92.8	92.6
PneumoniaMNIST	80.9	83.3	87.0	84.1	84.0	85.4	82.7	82.9	84.1	84.1	83.3	89.7
EuroSAT	97.7	97.9	98.1	97.9	98.1	98.1	97.5	97.6	97.5	97.6	97.5	97.6
RESISC45	93.7	94.7	95.0	95.1	95.1	95.1	94.1	94.2	94.0	94.0	94.4	95.1
CLEVR Count	68.1	73.6	80.9	86.9	92.0	94.1	67.0	68.0	73.8	82.3	86.0	75.5
smallNORB Elev.	36.1	38.1	41.5	44.9	47.4	51.6	35.8	37.4	38.6	39.4	41.0	31.8
Mean	84.9	86.6	87.8	88.4	89.2	89.7	85.6	85.9	86.6	87.3	87.7	88.2

Table C.1: **Domain-structured donor sharing** (K=5, N=1,024, random candidates). *Top:* Jaccard over the top 5/10/20% donors (chance 2.6/5.3/11.1%) and Spearman ρ of donor energy. Dom.: domains (of six) where same-domain sharing is higher; p: permutation test over domain labels; p_{adj}: same test after regressing out mask similarity. *Bottom:* top-10% Jaccard per domain; Other averages pairs with one task inside and one outside the domain.

	ViT-S+/14M					ViT-B/35M					ViT-H+/300M				
Measure	Same	Other	Dom.	p	p_{adj}	Same	Other	Dom.	p	p_{adj}	Same	Other	Dom.	p	p_{adj}
Jaccard, top 5%	3.38	3.19	3/6	0.182	0.123	4.63	3.84	6/6	$<10^{-3}$	0.003	3.21	2.97	6/6	$<10^{-4}$	0.012
Jaccard, top 10%	6.27	5.88	4/6	0.018	0.033	7.82	6.87	6/6	$<10^{-4}$	$<10^{-3}$	6.03	5.78	6/6	0.001	0.030
Jaccard, top 20%	12.21	11.98	4/6	0.114	0.177	14.12	13.16	5/6	$<10^{-3}$	0.003	12.13	11.80	6/6	$<10^{-4}$	0.003
Spearman ρ	0.047	0.038	5/6	0.061	0.113	0.103	0.076	6/6	$<10^{-3}$	0.008	0.041	0.035	6/6	0.007	0.009
General	6.54	5.97				7.08	6.81				6.07	5.86			
Fine-grained	5.70	5.71				8.54	7.00				6.07	5.79			
Texture/scene	7.65	5.93				8.15	6.98				6.32	5.84			
Medical	6.84	6.04				7.37	6.84				5.87	5.69			
Remote sensing	6.01	6.02				7.53	6.85				6.11	5.75			
Synthetic	6.07	5.65				7.03	6.68				5.72	5.70			

C Extended Analyses and Ablation Studies

The *12-task ablation suite* used throughout this appendix contains the general, fine-grained, texture/scene and medical tasks, i.e., all benchmark tasks except EuroSAT, RESISC45, CLEVR Count and smallNORB Elevation.

C.1 Cross-Task Donor Sharing and Domain Clustering

Measure. For task t, layer ℓ and pruned MLP channel i, the donor energy $\epsilon_{t\ell i} = \sum_j \eta_{ji}^2$ is the summed squared coefficient that channel i contributes to all compact channels. For a pair of tasks, we restrict to channels pruned in both, take each task's top 10% by donor energy, compute their Jaccard index, and average over layers. The restriction removes the trivial effect of different pruning masks; two random sets of this size give $q/(2-q) = 5.3\%$ for $q = 0.1$. We use the K=5, N=1,024 runs of the data-efficiency study, whose candidates are uniformly random. Each task then assigns some coefficient to most pruned channels, so the overlap of full donor sets is uninformative, and we compare instead the channels each task relies on most.

Robustness. Table C.1 varies the threshold and replaces the Jaccard index with the Spearman correlation of donor energy over all commonly pruned channels. On ViT-B/35M and ViT-H+/300M, same-domain pairs share more donors under every measure ($p \leq 0.007$), and the effect stays significant after regressing out pruning-mask similarity. This adjustment matters because same-domain tasks also keep more similar channels (mask Jaccard 0.63 vs. 0.57 on ViT-B/35M). On ViT-S+/14M the ordering holds for every measure but is significant only for the top-10% Jaccard. The lower part of the table breaks sharing down by domain.

Shared banks under a candidate prior. With candidate-ranker priors ($N \leq 512$), the tasks of the 12-task suite converge on a common set of donors (Figure C.1). Each task uses about 88 of roughly 970 pruned channels per layer, the twelve ViT-S/10M models together use 32% of the MLP channels (48% expected for independent sets of the same size), and pairwise overlap is six times its chance value. With random candidates, the same statistics equal chance. The per-channel recipes stay task-specific: the donors assigned to an anchor retained by two tasks overlap by 4.6%, against 17.7% for the layer-level donor sets. Since the ranker does not reduce accuracy (Appendix C.7), one stored subset of about a third of the MLP channels can serve all twelve tasks in this setting.

C.2 Sensitivity to Donor Budget and Calibration Size

Figure C.2 and Table C.2 sweep $K \in \{3, 5, 7\}$ and $N \in \{256, 512, 1024, 2048\}$ on the 12-task suite for five DINOv3 settings (720 runs). Candidates come from ranker priors at $N \leq 512$ and from uniform sampling at $N \geq 1{,}024$. At $N \leq 512$ the three budgets are statistically indistinguishable

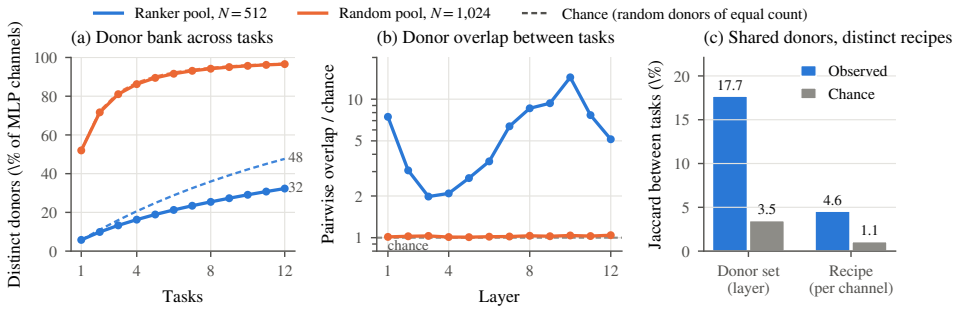


Figure C.1: **A cross-task candidate prior induces a shared donor bank** (twelve ViT-S/10M models, K=5). Dashed lines and gray bars: random donors of equal count from each task's own pruned channels. (a) Distinct MLP donors used by the first n tasks, averaged over layers and 400 task orders; the random-pool curve coincides with chance. (b) Pairwise donor-set overlap relative to chance. (c) With the ranker: overlap of layer-level donor sets vs. of donors assigned to the same anchor in two tasks.

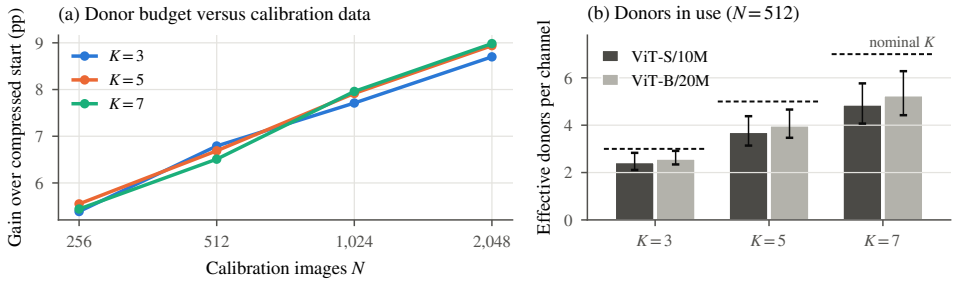


Figure C.2: **Calibration data matters more than the donor budget.** (a) Mean gain over the compressed model (twelve tasks, five DINOv3 settings, 60 pairs per point). (b) Effective donors per compact MLP channel at N=512 (bars: mean; whiskers: interquartile range; dashed: nominal budget).

(paired differences at most 0.28 points, Wilcoxon $p \geq 0.12$). At $N \geq 1{,}024$, $K = 5$ improves on $K = 3$ by 0.2 points (40 of 60 pairs, $p \leq 0.004$) and $K = 7$ adds nothing further. Increasing N from 256 to 2,048 raises the gain over the compressed model by 3.3–3.5 points for every K. The extra slots are used: the effective number of donors per channel (the exponential entropy of the squared-coefficient shares) is 2.4, 3.7 and 4.9 for $K = 3, 5, 7$ on ViT-S/10M. Calibration data rather than the donor budget limits the gain, which is why K needs little tuning (Section 4.2), and the gains hold under both candidate-discovery regimes.

C.3 Recombination Versus Rescaling

A remixed channel changes its scale through the anchor gain and its direction through its donors. To separate the two, we compare a refitted classifier, per-channel MLP gains without donors, and a dense MLP remix on a ViT-S/6M model with full task data from the 12-task suite (Figure C.3). Gains raise mean accuracy from 43.5% to 49.5%, while mixing reaches 63.9% and beats gains on all twelve tasks (Wilcoxon $p = 5 \times 10^{-4}$). Sparse K=3 donors show the same ordering (Table 2(a)): anchor gains alone reach 31.7%, three random donors 47.6%, and learned donors 48.9%. Most of the recovery thus comes from changing what compact channels compute. These studies use full task data and the best of ten evaluated epochs, so they compare the capacity of the parameterizations, not their data efficiency.

Table C.2: **Donor budget and calibration sweep.** Mean accuracy (%) on the 12-task suite; each column uses a fixed K for all tasks.

		N=256			N=512			N=1,024			N=2,048		
Setting	Start	K=3	5	7	K=3	5	7	K=3	5	7	K=3	5	7
S/10M	75.84	80.92	80.81	81.21	81.94	82.16	82.21	82.52	82.84	82.97	83.22	83.44	83.76
B/20M	53.60	64.70	65.45	66.02	68.83	68.79	68.53	71.44	72.14	72.03	74.06	74.73	75.09
B/35M	78.60	85.80	85.69	85.87	86.35	86.55	86.42	87.32	87.62	87.67	87.93	87.93	87.86
L/100M	83.91	86.97	87.24	85.74	87.69	87.23	86.89	88.17	87.74	88.21	88.90	89.09	89.04
L/150M	90.50	90.99	91.00	90.82	91.59	91.16	90.96	91.56	91.68	91.37	91.84	91.96	91.62

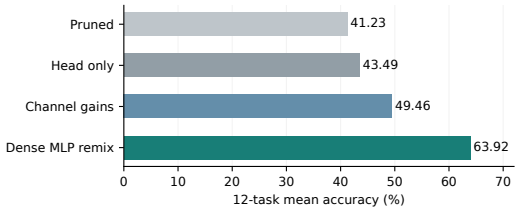


Figure C.3: **Rescaling versus recombination.** Mean accuracy on the 12-task suite (ViT-S/6M, full task data) of the compressed model (Pruned), a refitted classifier (Head only), learned per-channel MLP gains (Channel gains), and a dense MLP remix.

C.4 Attention Remixing Diagnostic

The safeguard of Appendix A.6 is motivated by a diagnostic on CIFAR-100 with a DINOv3 ViT-L/100M model (K=3, N=512). From a compressed start of 80.77%, MLP-only remixing reaches 84.96%, whereas unguarded joint remixing of MLPs and attention (cross- and within-head) reaches only 80.80%, four points lower. Cross-head attention remixing is responsible (80.75% for MLP plus cross-head attention only), while within-head remixing is nearly harmless (84.80%). The failing configurations show much larger first-epoch attention-output errors, which the safeguard detects without test data.

C.5 Computational Cost

We report the cost of backbone adaptation only; the linear classifier is shared by all methods, and all methods start from the same compressed model. As a reference we use the smallest setting in which every method has the same twenty-epoch schedule: AugReg ViT-B/16 compressed to 35M, with 512 calibration images.

Training time. From per-epoch timestamps in the training logs (one GPU of an NVIDIA RTX PRO 6000 node shared with other jobs, so times are approximate), the median adaptation time per task is 1.9 minutes for *Channel-Remix* (K=5), including donor search and refinement. The baselines take 1.3–2.8 minutes (e.g., 1.9 for full fine-tuning, 2.2 for PiSSA-8, 2.7 for LoRA-8), and SVFT 4.8 minutes. With 512 images, adaptation is therefore a matter of minutes for every method, and *Channel-Remix* is not slower than the adapter baselines.

Training memory. *Channel-Remix* keeps the frozen source weights (86M parameters, about 0.34 GB in fp32) and allocates dense coefficient storage during search (median 35M slots, about 0.4 GB with AdamW moments), although only 112K coefficients remain active. Full fine-tuning of the 35M compact backbone needs about 0.56 GB for weights, gradients, and optimizer moments. These are estimates from parameter counts; peak GPU memory was not logged, and at 512 images activations dominate for all methods.

Inference. After folding, *Channel-Remix* has exactly the architecture, parameter count, and FLOPs of the compressed model. The same holds for B+LN, SSF, and the LoRA family after

Table C.3: **Complete donor-discovery comparison.** Test accuracy (%) on held-out MIT-67 (ViT-S/6M, K=3, one seed; checkpoints chosen on a calibration holdout). Ranker pools come from a ranker trained on Aircraft and Caltech-256.

Procedure	N=512	1,024	2,048	Mean
Top-K refinement only, all pruned	29.6	34.3	39.9	34.6
Top-K refinement only, random pool	26.3	33.4	40.8	33.5
Top-K refinement only, ranker pool	28.0	33.9	39.6	33.8
One-shot $32 \to K$, local loss, ranker pool	22.0	31.6	40.4	31.4
Halving, local loss, random pool	25.7	32.2	42.5	33.5
Halving, local loss, ranker pool	26.6	34.9	40.0	33.8
Halving, full objective, random pool	30.0	33.4	41.9	35.1
Halving, full objective, ranker pool	27.3	32.2	41.4	33.6
Halving, full objective, ranker pool, no diversity penalty	–	33.7	–	–
Reference: top-K refinement only, K=5	30.9	35.0	42.2	36.0

merging, and for full fine-tuning and block reconstruction. AdaptFormer keeps its adapters at inference. Deployment cost is therefore set by the compression budget (Table 1), not by the adaptation method.

C.6 Ablation Setup

All component ablations in Table 2 use a DINOv3 ViT-S/16 backbone compressed to 6M parameters, K=3 donors per anchor channel, and learnable anchor gains.

Full task data (Table 2(a)). Models are trained for 10 epochs on the full training splits of FGVC-Aircraft, Caltech-256 and MIT Indoor-67, and every arm reports its best evaluated checkpoint. *W/o donors* disables donor recruitment and optimizes only the anchor gains s_j. *W/o support learning* fixes the donors before training, either uniformly at random from the pruned pool or as the three discarded channels with the largest calibration-task gradients, and optimizes only the mixing coefficients and anchor gains. *W/o teacher guidance* removes the representation and logit distillation losses ($\mathcal{L}_{\text{rep}}$, $\mathcal{L}_{\text{KD}}$) and trains with $\mathcal{L}_{\text{task}}$ alone.

Constrained calibration (Table 2(b)). We use MIT Indoor-67, which is held out from ranker pre-training, with $N \in \{512, 1024, 2048\}$; a stratified 25% of the calibration budget ($N/4$) is held out for checkpoint selection. Each arm changes one design choice. Successive halving ($32 \to 16 \to 8 \to K$) is compared with one-shot reduction from 32 candidates to K, both with ranker pools and a local block-reconstruction loss. The full objective (Eq. equation 3) is compared with a local layer-wise reconstruction loss, both with successive halving and uniform random candidates. Finally, the discovery stage is compared with directly optimizing top-K donor coefficients over all pruned channels. With 1,340 MIT Indoor-67 test images, the standard error between two individual runs is about 1.8 points, and about 1.0 point for the three-budget mean.

C.7 Complete Donor-Discovery Comparison

Table C.3 lists every discovery procedure behind Table 2(b). Successive halving is robust to the candidate pool: ranker candidates are within 0.5–2.7 points of uniform random ones across budgets. The cross-task ranker predicts optimizer-selected edges three times more often than random initialization (9.0% vs. 2.9%) and promotes domain-structured donor sharing (Appendix C.1), but successive halving under the task objective is the main driver of accuracy. Evaluating the gradient-diversity penalty at N=1,024 shows that candidate pruning is already effective, because pruned directions are nearly orthogonal (gradient redundancy 0.37–0.55). *Channel-Remix* thus finds good donor subspaces without rigid candidate pre-filtering.

Table C.4: ***Channel-Remix* after other structured pruning criteria.** Mean accuracy (%) on the 12-task suite, N=512. Start: pruned model with a fresh linear probe; K: donors per channel.

	Magnitude start				Wanda-style start			
Setting	Start	K=3	K=5	K=7	Start	K=3	K=5	K=7
S/10M	28.4	33.5	34.4	33.9	40.8	56.0	58.0	58.9
B/20M	29.9	36.8	39.5	39.2	29.2	39.3	40.7	40.3
B/35M	33.9	44.7	45.3	45.6	36.5	55.0	54.0	54.8
L/100M	24.8	44.1	44.3	44.1	33.1	56.8	57.1	55.3
L/150M	31.1	52.4	52.1	52.0	40.4	80.0	76.8	71.9
H+/300M	31.5	55.5	54.5	53.5	49.3	83.2	80.3	79.1

C.8 Other Pruning Criteria

Channel-Remix needs only the partition into retained and discarded channels, so it can follow any structured pruning criterion. Table C.4 starts from structured magnitude and activation-weighted (Wanda-style) channel pruning at the budgets of the 12-task suite, with 512 calibration images. *Channel-Remix* improves every start in all six settings; at K=5 the average gain is 15.1 points from magnitude and 22.9 points from Wanda-style pruning. The final accuracies remain well below those from the task-guided starts of the main experiments (Table C.2), so the gains are large partly because these starts are weak. Remixing extends what can be learned within a chosen architecture; it does not make channel selection irrelevant.