DON'T JUST PRUNE-AND-FINE-TUNE: OPTIMAL INFORMATION MIGRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured pruning typically follows a simple recipe: rank, remove, and fine-tune. We argue that this conflates structural removal with functional discard. A structural unit may no longer warrant an independent place in the compressed network, while its learned contribution need not be discarded with it. Two compressed models can therefore have the same final architecture yet retain very different amounts of pretrained functionality, depending on how these contributions are handled before deletion. We reformulate structured pruning as a *migration-before-deletion* process and propose Optimal Information Migration (OIM). Information Preservation Priority (IPP) selects retained targets, while OIM uses optimal transport to migrate pretrained contributions onto those targets before deletion. We further introduce DeepInversion-based BatchNorm recalibration (DI-BN) to correct compression-induced statistical mismatch without real training data or updates to learnable parameters. Across CNNs and a vision Transformer on CIFAR-10/100 and ImageNet-1K, OIM consistently preserves substantially more pre-fine-tuning accuracy than direct pruning at the same compression budget and final architecture. On ResNet-50/ImageNet-1K at 30.26% FLOPs reduction, Top-1 accuracy improves from 15.79% after direct pruning to 42.15% with OIM and further to 52.55% with DI-BN.

1 INTRODUCTION

Modern vision models often need to be compressed after training to meet the memory and computational constraints of deployment. Structured pruning removes entire filters, channels, or other parameter groups, yielding compact models with lower computational cost that can run efficiently on standard hardware (Li et al., 2017; Fang et al., 2023). A large body of work has therefore developed increasingly sophisticated criteria for deciding which structural units to remove, using signals such as filter magnitude, next-layer response reconstruction, geometric redundancy, loss sensitivity, and structural dependencies (Li et al., 2017; Luo et al., 2017; He et al., 2019; Molchanov et al., 2019; Fang et al., 2023). Yet the dominant pruning recipe has barely changed: importance ranking, structural removal, and model fine-tuning (Kim et al., 2020; Theus et al., 2024). We revisit the second stage and examine what happens when the selected structures are removed.

Our key observation is that a structural unit may no longer warrant an independent position in the compressed model, yet the function it has learned need not disappear with it. Direct removal eliminates not only the unit itself but also its aligned downstream connections, thereby removing its contribution to subsequent computation (Kim et al., 2020). Fine-tuning is then typically used to recover the resulting accuracy loss, usually requiring access to the original training data (Theus et al., 2024; Wang et al., 2025). If that contribution is first migrated to the retained structures, the same set of units can be removed without discarding all of the functionality they carried. The final architecture can therefore be identical while retaining more of the pretrained functionality. This motivates a question beyond conventional importance ranking: if a structural unit must eventually be removed, can its useful functional contribution be migrated to the retained structures before deletion?

Motivated by this perspective, we reformulate structured pruning as a *migration-before-deletion* process and introduce Information Preservation Priority (IPP) and Optimal Information Migration (OIM). Here, *information* refers to the functional contribution learned by a structural unit in the pretrained network. IPP determines which structural units to retain as the target support, while OIM

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

uses optimal transport (Peyré & Cuturi, 2019) to redistribute the contributions of units selected for removal onto this retained support. This shifts the focus from identifying what is dispensable to preserving as much pretrained functionality as possible under a fixed compression budget.

Migration alone, however, does not address all effects of structural compression. Structural compression can also alter the activation statistics propagated through a network (Saikumar & Varghese, 2025; Wang et al., 2025). In particular, BatchNorm layers retain running statistics estimated before compression (Ioffe & Szegedy, 2015), which may no longer match the activations of the compressed model (Saikumar & Varghese, 2025). We therefore introduce DeepInversion-based BatchNorm recalibration (DI-BN), which builds on DeepInversion (Yin et al., 2020) to adapt the BatchNorm statistics of the compressed model without access to the original training data or updates to its learnable parameters. Together, OIM and DI-BN form a post-training compression pipeline that requires neither access to the original training data nor parameter fine-tuning.

Our experiments focus primarily on performance before fine-tuning. We compare direct pruning, OIM, and OIM+DI-BN across multiple models and compression budgets to measure the immediate effects of information migration and statistical recalibration. We then compare IPP with ℓ_1, ℓ_2, and FPGM (He et al., 2019) under the same OIM mechanism to isolate the role of target-support selection. We further evaluate the compressed models after post-compression fine-tuning and analyze intermediate representations and BatchNorm statistics to examine the mechanisms underlying OIM and DI-BN.

Our contributions are threefold:

- *We rethink structured pruning as migration before deletion.* We separate the removal of a structural unit from the loss of the function learned by that unit, making functional preservation an explicit objective before deletion.
- *We introduce IPP and OIM for target-support selection and information migration.* IPP prioritizes positions whose removal without migration would cause a large downstream disturbance, while OIM uses optimal transport to migrate the downstream contributions of units selected for removal into the retained support before dependency-consistent structural deletion.
- *We build a post-training compression pipeline that requires neither the original training data nor parameter fine-tuning.* DI-BN recalibrates the BatchNorm statistics of the compressed model using synthetic inputs, and we analyze the pipeline through performance before fine-tuning, support-selection ablations, post-compression fine-tuning, and internal activation statistics.

2 Background

2.1 Structured Pruning

Structured pruning mainly addresses two questions: which structures to remove, and how to remove dependent structures consistently. Filter-pruning criteria commonly use either weights or activations to assess importance (He et al., 2020; Lin et al., 2020). Weight-based methods evaluate filters directly from pretrained parameters: Li et al. (2017) use filter ℓ_1-norms, while FPGM assesses redundancy through proximity to the geometric median in weight space (He et al., 2019). Activation-based methods use data-dependent network responses. ThiNet selects filters by minimizing reconstruction error in the next layer (Luo et al., 2017), whereas HRank measures importance through the rank of output feature maps (Lin et al., 2020).

A channel cannot always be removed independently. Residual connections and grouped or depthwise convolutions introduce dependencies that require coordinated changes across operations (Liu et al., 2021; Fang et al., 2023). Gate Decorator uses architecture-specific grouping (You et al., 2019), Group Fisher automatically identifies coupled channels through layer grouping (Liu et al., 2021), and DepGraph traces pruning dependencies through the computation graph to identify parameter dimensions that must be removed together (Fang et al., 2023).

A complementary line of work seeks to preserve the contributions of units selected for removal. Neuron Merging matches each pruned neuron to a similar retained neuron and incorporates its con-

tribution into the corresponding downstream weights (Kim et al., 2020). UDFC and LBYL use linear combinations of retained units to approximate the contributions of pruned units (Bai et al., 2023; Lee et al., 2025), while AutoDFP jointly searches pruning and reconstruction strategies (Li et al., 2026). These methods broaden the role of pruning from selecting structures to preserving their learned functionality within the remaining network.

2.2 Optimal Transport

Discrete Optimal Transport (OT) finds a nonnegative transport plan that minimizes the total transport cost subject to prescribed source and target masses (Peyré & Cuturi, 2019). Each entry specifies the mass assigned from a source unit to a target unit, allowing their correspondences to be determined jointly under marginal constraints.

The resulting coupling can represent many-to-many correspondences between source and target supports. OTFusion uses this formulation to align neurons across independently trained networks before parameter fusion (Singh & Jaggi, 2020). Intra-Fusion extends this approach to structured pruning, using an importance criterion to select target units and OT to aggregate the original pruning group into a compact representation (Theus et al., 2024).

2.3 Activation Statistics Repair

Parameter interpolation, model fusion, and structural compression can alter internal activation statistics even when structural or neuronal correspondences have been established (Jordan et al., 2023; Wang et al., 2025). REPAIR identifies progressive activation variance collapse in aligned model interpolation and restores channel statistics through rescaling and shifting (Jordan et al., 2023). In pruning, REFLOW links signal collapse to a mismatch between post-pruning activations and the BatchNorm running statistics inherited from the dense model. It recalibrates these statistics using a small set of real training samples while keeping learnable parameters fixed (Saikumar & Varghese, 2025).

Such calibration requires access to representative data, which may be unavailable in post-training deployment because of privacy, licensing, or storage constraints (Yin et al., 2020; Wang et al., 2025). DeepInversion synthesizes inputs from a frozen pretrained model using its stored BatchNorm statistics as priors on intermediate feature distributions (Yin et al., 2020). These synthetic inputs provide a basis for statistical recalibration without access to the original training set.

3 Method

We reformulate structured pruning as a migration-before-deletion process. Given a pretrained network and a target compression budget, our method proceeds in three stages. First, IPP identifies the units that should remain as the target support, prioritizing those whose removal without migration is expected to cause larger downstream disturbance. Second, OIM treats the original units as transport sources and the retained support as transport targets, and uses optimal transport to determine how the downstream contributions of units selected for removal should be redistributed before dependency-consistent structural deletion. Finally, because migration and structural removal can alter internal activation statistics, DI-BN synthesizes calibration inputs from the frozen dense model and re-estimates the BatchNorm running statistics of the compressed network. The resulting pipeline requires neither access to the original training data nor parameter fine-tuning. Figure 1 illustrates the central migration-before-deletion process.

3.1 Information Preservation Priority

Information migration first requires deciding which filters should remain as migration targets. A natural principle is to prioritize filters whose removal without migration could strongly perturb downstream computations. We assess this risk through an upper bound on the resulting perturbation.

We build on the adjacent-layer replacement analysis of Tofigh et al. (2026). Consider the p-th filter in convolutional layer l, denoted by $F^l_{[:,:,:,p]}$, and let $\widehat{F}^l_{[:,:,:,p]}$ denote an alternative representation of this filter after its contribution has been preserved elsewhere. The perturbation $\mathcal{E}_p$ induced at the

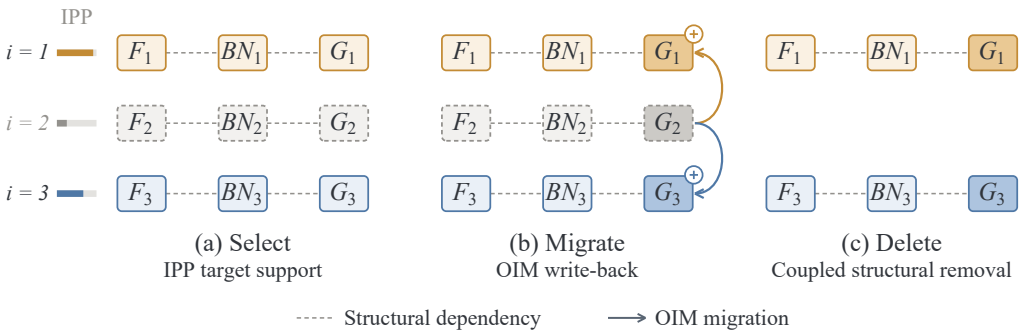

Figure 1: Migration before deletion: select targets, migrate contributions, and remove coupled structures. Each row shows an output filter F_i, its aligned BatchNorm entries BN_i, and the corresponding downstream input-weight slice G_i; dashed lines denote structural dependencies, not forward connections. (a) IPP selects channels 1 and 3 as the retained target support. (b) OIM adds weighted contributions from G_2 to the existing G_1 and G_3 according to the transport plan. The retained output filters and BatchNorm entries remain unchanged. (c) Channel 2 and its structurally coupled parameters are removed only after migration. Darker fills highlight input-weight slices involved in migration; circled plus signs denote additive updates.

next layer is bounded by

$$\mathcal{E}_p \leq \left\| F^{l+1}_{[:,:,p,:]} \right\|_2^2 \left\| F^{l}_{[:,:,:,p]} - \widehat{F}^{l}_{[:,:,:,p]} \right\|_2^2 \Lambda_{l-1}, \tag{1}$$

where Λ_{l-1} depends on the preceding inputs and parameters and is shared by candidate filters within the same layer. The bound combines the discrepancy between the original and preserved filter representations with the magnitude of the corresponding input weights in the next convolutional layer.

To determine which filters should retain their original structural positions, we consider the limiting case in which the contribution of filter p is not migrated at all, represented by $\widehat{F}^{l}_{[:,:,:,p]} = 0$. Defining $X_p^l := \|F^l_{[:,:,:,p]}\|_2^2$ and $Y_p^{l+1} := \|F^{l+1}_{[:,:,p,:]}\|_2^2$, the bound becomes

$$\mathcal{E}_p^0 \leq X_p^l Y_p^{l+1} \Lambda_{l-1}. \tag{2}$$

Here, X_p^l measures the magnitude of the filter itself, while Y_p^{l+1} measures the magnitude of its corresponding input weights in the next layer. Their product determines the filter-dependent part of the no-migration error bound.

Following the conservative criterion of Tofigh et al. (2026), we assign low preservation priority only when both factors are small. We therefore define IPP as

$$\mathrm{IPP}_l(p) = \max\left\{X_p^l, Y_p^{l+1}\right\}. \tag{3}$$

A small IPP implies that both factors are small, yielding a small upper bound in the no-migration case. Such a filter receives lower priority for retaining its original position. Conversely, a large IPP indicates that at least one factor is large, so the criterion gives the filter higher retention priority.

For layer l with H_l candidate filters, we rank the filters by IPP under the prescribed compression budget. Let $\mathcal{P}_l$ contain the indices selected for removal. The remaining filters form the target support $\mathcal{T}_l = \{1, \ldots, H_l\} \setminus \mathcal{P}_l$, with $m_l = |\mathcal{T}_l|$. These retained positions serve as the targets for the subsequent information migration.

3.2 Optimal Information Migration

Given the target support selected by IPP, OIM determines how to migrate the contributions of filters selected for removal before deleting their structural carriers. We formulate the source–target assignment as an optimal transport problem, using all original filters, indexed by $\mathcal{S}_l = \{1, \ldots, H_l\}$, as sources and the retained filters $\mathcal{T}_l = \{q_1, \ldots, q_{m_l}\}$ as targets. The resulting transport plan specifies the coefficients used to migrate contributions from $\mathcal{P}_l$ into the retained downstream input weights.

Migration Cost. Removing a filter can trigger coordinated changes at multiple positions in the computation graph. We use DepGraph to identify the corresponding dependency group (Fang et al., 2023). This group captures the parameter dimensions coupled to the same channel index, providing the structural context for comparing source and target channels. It also specifies the complete set of coupled dimensions removed after migration.

Let $\theta^l_{r,i}$ denote the slice associated with channel i in the r-th aligned tensor, and let R_l be the number of such tensors. We concatenate these slices as $z^l_i = \mathrm{Concat}_{r=1}^{R_l}(\mathrm{vec}(\theta^l_{r,i}))$ and normalize the descriptor as $\widehat{z}^l_i = z^l_i/(\|z^l_i\|_2 + \epsilon)$, where vec flattens each slice and $\epsilon > 0$ ensures numerical stability. The migration cost from source i to target q_j is

$$C^l_{ij} = \left\| \widehat{z}^l_i - \widehat{z}^l_{q_j} \right\|_1 . \tag{4}$$

A smaller cost indicates greater similarity between the channels under this dependency-aligned representation. Collecting these costs gives $C^l \in \mathbb{R}^{H_l \times m_l}$, which defines the source–target costs for the transport problem.

Optimal Transport Plan. We assign uniform probability masses $a^l_i = 1/H_l$ to the sources and $b^l_j = 1/m_l$ to the targets. The transport plan $T^l \in \mathbb{R}^{H_l \times m_l}_{\geq 0}$ specifies the mass assigned from each source i to each target q_j. OIM jointly determines these assignments by solving the discrete OT problem (Peyré & Cuturi, 2019):

$$\begin{aligned} T^{l*} &= \arg\min_{T^l \geq 0} \left\langle T^l, C^l \right\rangle_F , \\ \text{s.t.} \quad T^l \mathbf{1}_{m_l} &= a^l, \qquad (T^l)^\top \mathbf{1}_{H_l} = b^l. \end{aligned} \tag{5}$$

Here, a^l and b^l collect the source and target masses, $\mathbf{1}_k$ is the length-k all-ones vector, and $\langle T^l, C^l \rangle_F = \sum_{i,j} T^l_{ij} C^l_{ij}$ is the total transport cost. The constraints fix the row and column sums to the prescribed masses.

To turn the transported masses into migration coefficients, we normalize each row as $M^l_{ij} = T^{l*}_{ij}/a^l_i$. Since $\sum_j M^l_{ij} = 1$, these coefficients specify how each source's contribution is distributed across the retained targets.

Information Migration and Structural Removal. The transport plan is computed over all original filters; the parameter update uses the rows corresponding to $\mathcal{P}_l$ to migrate the contributions selected for removal. Let $\mathcal{D}_l$ denote the downstream layers whose input-channel dimensions belong to the dependency group. For each $d \in \mathcal{D}_l$, let $\mathbf{G}^{l\to d}_i$ denote the input-weight slice associated with filter i. OIM updates each retained target as

$$\mathbf{G}^{l\to d}_{q_j} \leftarrow \mathbf{G}^{l\to d}_{q_j} + \sum_{p \in \mathcal{P}_l} M^l_{pj}\, \mathbf{G}^{l\to d}_p, \qquad d \in \mathcal{D}_l. \tag{6}$$

The existing input weights at q_j remain as the base, and the weighted contributions from the removed channels are added to them. When several downstream layers share the dependency, the same coefficients are applied to each corresponding weight tensor. The retained output filters and their aligned BatchNorm entries remain unchanged during migration.

After these updates, DepGraph removes the filters in $\mathcal{P}_l$ together with their aligned BatchNorm channels, downstream input channels, and other coupled dimensions. Migration thus places the discarded input-weight contributions into the retained coordinates before structural deletion. This stage uses only pretrained parameters and structural dependencies, requiring neither training data nor parameter optimization.

3.3 DeepInversion-based BatchNorm Recalibration

After pruning, the compressed network produces new activation distributions, while its BatchNorm layers retain running statistics estimated for the dense model. This mismatch can impair normalization (Saikumar & Varghese, 2025). We use DI-BN to adapt these statistics without accessing the original training data.

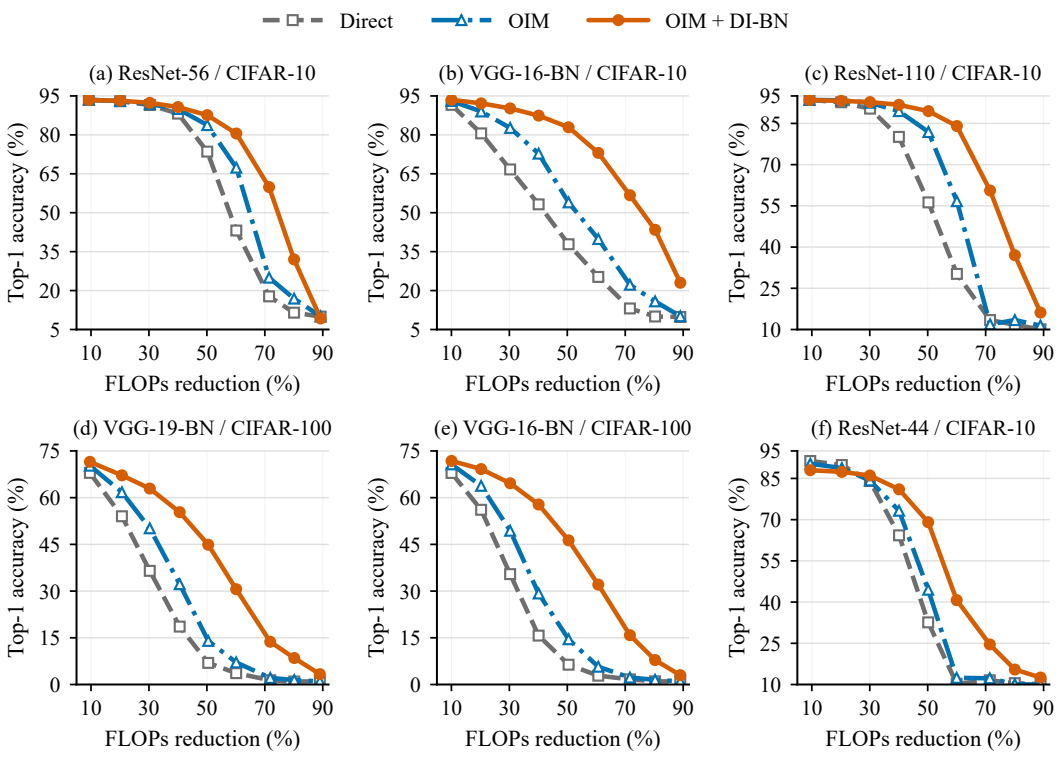

Figure 2: Pre-fine-tuning Top-1 accuracy on CIFAR-10/100.

Following DeepInversion (Yin et al., 2020), we synthesize calibration inputs by optimizing random noise against the frozen dense model. The objective combines a classification loss for prescribed class targets, an image prior, and a feature-statistics loss that matches the means and variances induced by the synthetic inputs to the running statistics stored in the dense model's BatchNorm layers. Only the inputs are optimized; the model parameters remain fixed throughout synthesis.

We then reset the compressed model's BatchNorm running statistics and pass the synthetic calibration inputs through it to re-estimate the running means and variances (Wang et al., 2025). All learnable parameters remain fixed during recalibration. DI-BN thus adapts the normalization statistics to the compressed model using synthetic inputs, completing the compression pipeline without original training data or parameter fine-tuning.

4 Experiments

We evaluate ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2015), WideResNet (Zagoruyko & Komodakis, 2016), MobileNetV2 (Sandler et al., 2018), and DeiT-T (Touvron et al., 2021) on CIFAR-10/100 (Krizhevsky, 2009) and ImageNet-1K (Russakovsky et al., 2015). We focus on pre-fine-tuning accuracy and IPP ablations, with compression requiring neither original training data nor parameter fine-tuning. For BatchNorm-based models, DI-BN recalibrates running statistics using 256 synthetic images generated from the corresponding frozen dense model. We also evaluate accuracy after 100 epochs of knowledge-distillation fine-tuning (Hinton et al., 2015) and visualize spatial responses after compression.

4.1 Preserving Performance Before Fine-Tuning

We compare Direct, OIM, and OIM + DI-BN before fine-tuning. At each compression budget, all variants share the same IPP-selected target support and final architecture, isolating the effects of information migration and subsequent BatchNorm recalibration.

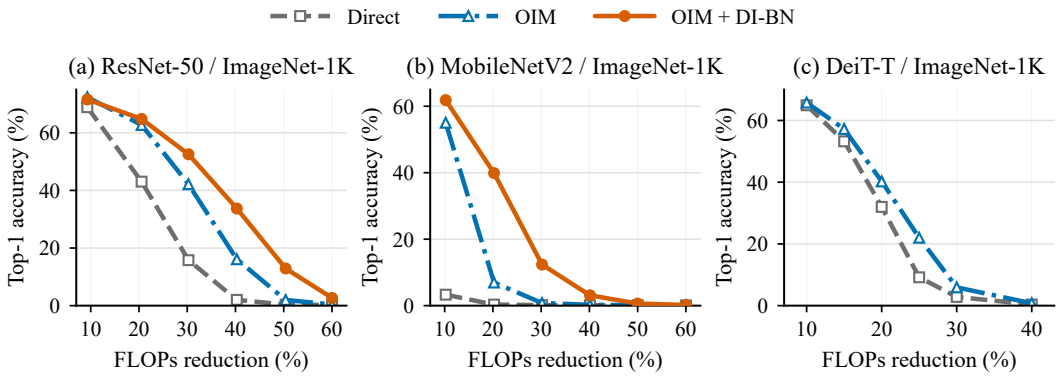

Figure 3: Pre-fine-tuning Top-1 accuracy on ImageNet-1K. DeiT-T uses LayerNorm, so DI-BN is not applicable.

Figures 2 and 3 show that OIM preserves more accuracy than direct deletion across the evaluated CIFAR and ImageNet settings, with additional recovery from DI-BN in BatchNorm-based models. On ResNet-50/ImageNet-1K at 30.26% FLOPs reduction, OIM raises Top-1 accuracy from 15.79% to 42.15%, and DI-BN further improves it to 52.55%. These gains are obtained before any parameter fine-tuning.

The benefit also extends to DeiT-T, which uses LayerNorm and is evaluated without DI-BN. At 25% FLOPs reduction, OIM improves Top-1 accuracy from 9.18% to 22.01%, demonstrating the value of migration for Transformer multilayer perceptron hidden units as well as convolutional channels.

4.2 Target-Support Selection Matters

We isolate the effect of target-support selection by replacing IPP with ℓ_1 (Li et al., 2017), ℓ_2, or FPGM (He et al., 2019) while keeping the OIM pipeline fixed. To compare the criteria over the

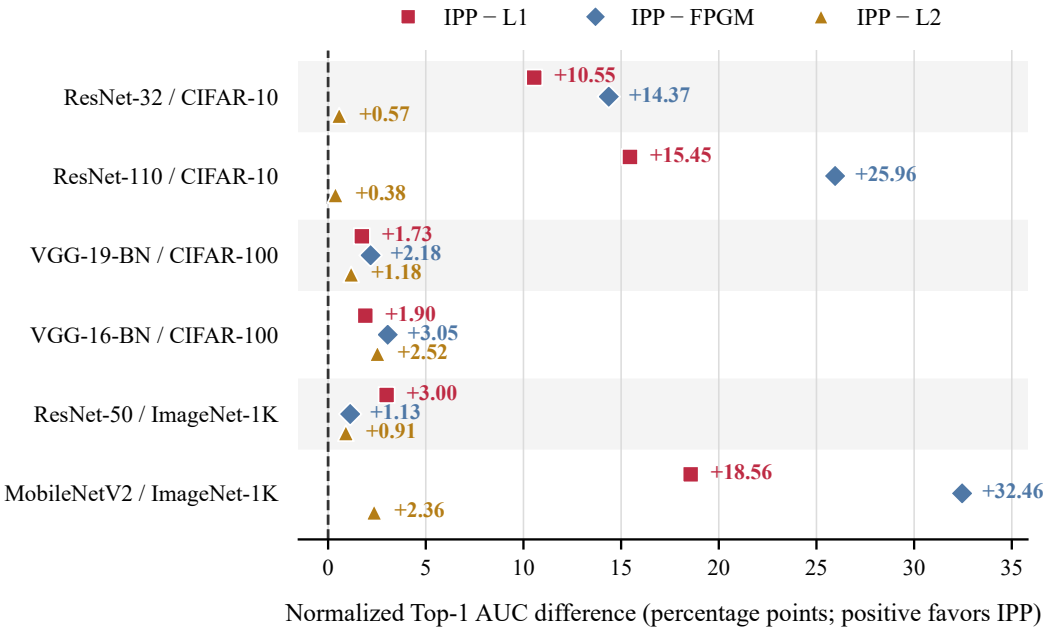

Figure 4: Effect of target-support selection: normalized AUC differences, AUC(IPP) − AUC(baseline), for pre-fine-tuning accuracy–compression curves.

entire compression trajectory rather than at a single operating point, we report the normalized area under the pre-fine-tuning accuracy–compression curve (AUC), using only operating points whose realized computation reductions differ by less than 1% across criteria.

As shown in Figure 4, IPP outperforms all three alternatives across all six model–dataset settings. On ResNet-110, IPP improves over ℓ_1, ℓ_2, and FPGM by 15.45, 0.38, and 25.96 normalized AUC points, respectively. The advantage is also pronounced on MobileNetV2/ImageNet-1K, where IPP exceeds ℓ_1, ℓ_2, and FPGM by 18.56, 2.36, and 32.46 points, respectively.

These results show that migration quality depends not only on how information is transported, but also on where it is preserved. By accounting for both a unit's own weights and its downstream dependence, IPP identifies a target support that is better aligned with information migration.

Table 1: Fine-tuned Top-1 accuracy on CIFAR and ImageNet-1K.

Model / Dataset	Method	Base Acc. (%)	Top-1 Acc. (%)	ΔAcc. (pp)	FLOPs ↓ (%)
ResNet-56 CIFAR-10	DTP (Li et al., 2023)	93.36	93.46	+0.10	49.75
	LIAP (Tofigh et al., 2026)	93.26	93.54	+0.28	50.50
	DepGraph (Fang et al., 2023)	93.53	93.77	+0.24	52.61
	Ours	93.53	**93.85**	+0.32	52.48
	FP-AGL (Kim & Kim, 2023; Li et al., 2025)	93.59	93.49	−0.10	60.92
	DepGraph (Fang et al., 2023)	93.53	93.64	+0.11	61.09
	SFP-D (Li et al., 2025)	93.26	93.29	+0.03	63.31
	Ours	93.53	**93.80**	+0.27	61.04
ResNet-110 CIFAR-10	HRank (Lin et al., 2020)	93.50	93.36	−0.14	58.20
	REAF (Zhang et al., 2023)	93.37	93.37	0.00	58.32
	LIAP (Tofigh et al., 2026)	93.50	93.91	+0.41	60.00
	Ours	93.68	**93.98**	+0.30	60.07
	HRank (Lin et al., 2020)	93.50	92.65	−0.85	68.60
	BNP (Liu et al., 2022)	93.54	93.04	−0.50	69.21
	LIAP (Tofigh et al., 2026)	93.50	93.28	−0.22	69.40
	Ours	93.68	**93.57**	−0.11	69.50
VGG-19-BN CIFAR-100	DepGraph (Fang et al., 2023; Chen et al., 2026)	73.50	73.30	−0.20	33.33
	OBC (Jacobian + EP) (Chen et al., 2026)	73.50	73.84	+0.34	33.33
	Network Slimming (Liu et al., 2017)	73.26	73.48	+0.22	37.10
	Ours	73.50	**74.17**	+0.67	33.63
	Network Slimming (4 passes) (Liu et al., 2017)	73.26	69.41	−3.85	83.00
	OBC (Jacobian + EP) (Chen et al., 2026)	73.50	**72.27**	−1.23	83.50
	Electrostatic force (p) (Ferdi et al., 2026)	74.59	69.00	−5.59	85.40
	Ours	73.50	70.27	−3.23	83.43
ResNet-50 ImageNet-1K	CCP (Peng et al., 2019)	76.15	75.50	−0.65	48.80
	Isomorphic Pruning (Fang et al., 2024)	76.13	**75.91**	−0.22	50.12
	WHC (Chen et al., 2023)	76.13	75.33	−0.80	53.50
	Ours	76.14	75.53	−0.61	50.40
	CLR-RNF-0.44 (Lin et al., 2023)	76.01	72.67	−3.34	70.07
	ThiNet-30 (Luo et al., 2017)	72.88	68.42	−4.46	71.50
	GAL-1-joint (Lin et al., 2019)	76.15	69.31	−6.84	72.86
	Ours	76.14	**73.04**	−3.10	71.48

4.3 Compatibility with Fine-Tuning

We next examine whether the functionality preserved before fine-tuning provides a strong initialization for subsequent optimization. Starting from the IPP + OIM + DI-BN checkpoints, we apply 100 epochs of knowledge-distillation fine-tuning (Hinton et al., 2015), using the corresponding frozen dense model as teacher. The objective combines ground-truth cross-entropy and T^2-scaled KL distillation with equal weights, using a temperature of $T = 4$. We compare with structured pruning methods at similar compression budgets.

Table 1 shows that the compressed models retain strong performance after optimization. On ResNet-56/CIFAR-10, our method reaches 93.85% and 93.80% Top-1 accuracy at 52.48% and 61.04% FLOPs reduction, respectively, exceeding its dense baseline by 0.32 and 0.27 percentage points. On ResNet-110, it achieves 93.98% at 60.07% reduction and retains 93.57% near 70% reduction. The result extends to CIFAR-100: on VGG-19-BN, our method reaches 74.17% at 33.63% FLOPs reduction.

At ImageNet scale, our method achieves 75.53% Top-1 accuracy at 50.40% FLOPs reduction and 73.04% at 71.48% reduction. The latter exceeds the listed high-compression baselines at comparable budgets. These results show that OIM-compressed models provide effective initializations for fine-tuning and retain competitive final accuracy after optimization.

4.4 What Does OIM Preserve?

The accuracy gains in Section 4.1 suggest that OIM preserves pretrained responses that direct deletion disrupts. We visualize this effect using class-discriminative spatial responses with Class Activation Mapping (CAM) (Zhou et al., 2016).

Figure 5 compares Dense, Direct, OIM, and OIM + DI-BN on WideResNet-40-4 (WRN-40-4) trained on CIFAR-10 at 49.78% FLOPs reduction. We visualize signed CAMs for the ground-truth class, using a shared color scale across models for each input to compare response magnitudes. In these representative examples, direct deletion shifts or attenuates the dense model's responses, whereas OIM retains spatial patterns closer to those of the dense model. DI-BN further improves this alignment.

These visualizations suggest that OIM preserves aspects of both where the network responds and how strongly those regions contribute to the target-class prediction, providing a spatial interpretation of its pre-fine-tuning accuracy gains.

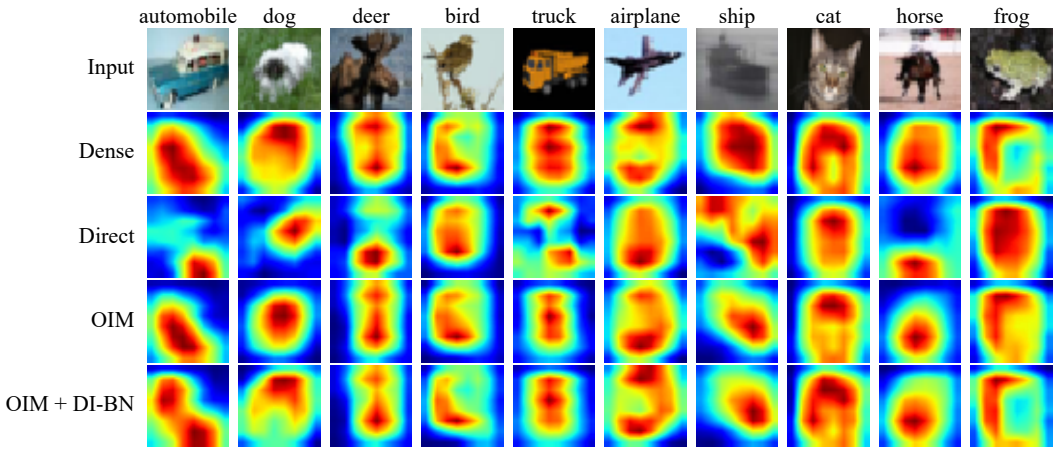

Figure 5: True-class signed CAMs on CIFAR-10 with WRN-40-4 at 49.78% FLOPs reduction. OIM better preserves the dense model's spatial response patterns, with further alignment after DI-BN.

5 Conclusion

We rethink structured pruning as a migration-before-deletion process, combining IPP for target selection with OIM for migrating pretrained contributions to retained positions. Across the evaluated architectures and compression budgets, information migration improves pre-fine-tuning accuracy over direct deletion at the same final architecture, showing that the compression procedure matters beyond the structure it produces. Target-selection ablations demonstrate the importance of choosing where contributions are preserved, while spatial visualizations suggest that migration retains response patterns disrupted by direct deletion. DI-BN provides complementary recovery by adapting BatchNorm statistics without original training data or parameter fine-tuning. The resulting models also remain effective starting points for subsequent fine-tuning. Together, these findings support a broader view of structured pruning: deciding that a unit is structurally dispensable need not make its learned contribution expendable. Structural deletion can instead become the final step of constructing what should survive.

AI USE STATEMENT

We used generative AI tools to assist with drafting and language editing of the manuscript, including improvements to clarity, organization, grammar, and concision. All AI-assisted content was critically reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work, including its text, claims, citations, figures, and conclusions.

REFERENCES

Shipeng Bai, Jun Chen, Xintian Shen, Yixuan Qian, and Yong Liu. Unified data-free compression: Pruning and quantization without fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5876–5885, 2023.

Shaowu Chen, Weize Sun, and Lei Huang. WHC: Weighted hybrid criterion for filter pruning on convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023.

Shaowu Chen, Wei Ma, Binhua Huang, Qingyuan Wang, Guoxin Wang, Weize Sun, Lei Huang, and Deepu John. Optimal brain connection: Towards efficient structural pruning. *IEEE Transactions on Multimedia*, pp. 1–11, 2026.

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

Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. In *European Conference on Computer Vision*, pp. 232–250. Springer, 2024.

Abdesselam Ferdi, Abdelmalik Taleb-Ahmed, Amir Nakib, and Youcef Ferdi. Electrostatic force regularization for neural structured pruning. *Information Sciences*, 735:123045, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2019.

Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2009–2018, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: REnormalizing permuted activations for interpolation repair. In *International Conference on Learning Representations*, 2023.

Nam Joon Kim and Hyun Kim. FP-AGL: Filter pruning with adaptive gradient learning for accelerating deep convolutional neural networks. *IEEE Transactions on Multimedia*, 25:5279–5290, 2023.

Woojeong Kim, Suhyun Kim, Mincheol Park, and Geunseok Jeon. Neuron merging: Compensating for pruned neurons. In *Advances in Neural Information Processing Systems*, volume 33, pp. 585–595, 2020.

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

Keonho Lee, Minsoo Kim, and Dong-Wan Choi. Training-free restoration of pruned neural networks. *arXiv preprint arXiv:2502.08474*, 2025.

Guoqing Li, Rengang Li, Tuo Li, Chaoyao Shen, Xiaofeng Zou, Jiuyang Wang, Changhong Wang, and Nanjun Li. SFP: Similarity-based filter pruning for deep neural networks. *Information Sciences*, 689:121418, 2025.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations*, 2017.

Siqi Li, Jun Chen, Jingyang Xiang, Chengrui Zhu, Jiandang Yang, Xiaobin Wei, Yunliang Jiang, and Yong Liu. Automatic data-free pruning via channel similarity reconstruction. *Neurocomputing*, 661:131885, 2026.

Yunqiang Li, Jan C. van Gemert, Torsten Hoefler, Bert Moons, Evangelos Eleftheriou, and Bram-Ernst Verhoef. Differentiable transportation pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16957–16967, 2023.

Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. HRank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1529–1538, 2020.

Mingbao Lin, Liujuan Cao, Yuxin Zhang, Ling Shao, Chia-Wen Lin, and Rongrong Ji. Pruning networks with cross-layer ranking & k-reciprocal nearest filters. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9139–9148, 2023.

Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. Towards optimal structured CNN pruning via generative adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2790–2799, 2019.

Huidong Liu, Fang Du, Lijuan Song, and Zhenhua Yu. Block-wisely supervised network pruning with knowledge distillation and Markov chain Monte Carlo. *Applied Sciences*, 12(21):10952, 2022.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group Fisher pruning for practical network compression. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7021–7032. PMLR, 2021.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2736–2744, 2017.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. ThiNet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5058–5066, 2017.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Hanyu Peng, Jiaxiang Wu, Shifeng Chen, and Junzhou Huang. Collaborative channel pruning for deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5113–5122. PMLR, 2019.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

Dhananjay Saikumar and Blesson Varghese. Signal collapse in one-shot pruning: When sparse models fail to distinguish neural representations. *arXiv preprint arXiv:2502.15790*, 2025.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055, 2020.

Alexander Theus, Olin Geimer, Friedrich Wicke, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Towards meta-pruning via optimal transport. In *International Conference on Learning Representations*, 2024.

S. Tofigh, M. Askarizadeh, M. Omair Ahmad, M. N. S. Swamy, and Kim Khoa Nguyen. Resource-efficient and layer interdependence-aware CNN pruning leveraging filter replacement. *IEEE Transactions on Neural Networks and Learning Systems*, 37(8):3998–4011, 2026.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021.

Dong Wang, Haris Šikić, Lothar Thiele, and Olga Saukh. Forget the data and fine-tuning! Just fold the network to compress. In *International Conference on Learning Representations*, 2025.

Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, pp. 87.1–87.12, 2016.

Xin Zhang, Weiying Xie, Yunsong Li, Kai Jiang, and Leyuan Fang. REAF: Remembering enhancement and entropy-based asymptotic forgetting for filter pruning. *IEEE Transactions on Image Processing*, 32:3912–3923, 2023.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.