PREDICTING IMP MASKS AT INITIALIZATION UNDER FINITE-TIME LINEARIZED DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Iterative magnitude pruning (IMP) selects sparse masks from weights obtained after training, whereas pruning at initialization selects masks before weight training. We formulate this difference as a **finite-time terminal-ranking problem**: can the magnitudes used by IMP be determined from initialization-side quantities? Under fixed-Jacobian squared-loss gradient descent, the finite-time response specifies terminal weights from the initialization, data, learning rate, and training horizon. Applying the same magnitude-pruning rule recovers one-shot and iterative linearized IMP masks when the top-k boundary is stable. We validate this correspondence in controlled and functional whole-network fixed-Jacobian models. On correlated-feature tasks, the predictor recovers teacher supports where the tested static initialization scores degrade, and finite-time ablations show that the converged solution can select different masks. Beyond the exact regime, initialization-side scores combined with architecture-based layer allocation produce competitive sparse retraining accuracy in several nonlinear LeNet and ResNet settings. Repeated IMP discoveries provide a reference for interpreting coordinate agreement under training randomness. A matrix-free implementation evaluates the predictor without explicit pseudoinverses. Together, these results connect posterior IMP selection to initialization-side trajectory prediction and distinguish exact mask recovery from functional transfer to nonlinear training.

1 INTRODUCTION

Neural network pruning identifies sparse subnetworks that retain useful predictive performance (Han et al., 2015; Blalock et al., 2020; Hoefler et al., 2021). The Lottery Ticket Hypothesis connects this search to initialization: a dense, randomly initialized network can contain a sparse subnetwork that trains effectively when reset to its original weights (Frankle & Carbin, 2019). Iterative magnitude pruning (IMP) discovers such masks through repeated training, magnitude pruning, and rewinding (Frankle & Carbin, 2019; Renda et al., 2020). Its selection criterion depends on terminal weights. This raises a question about the information needed for discovery: under specified training dynamics, can the mask selected after training be computed from initialization-side quantities?

Pruning at initialization (PaI) selects a mask before weight training. SNIP, GraSP, and SynFlow use connection sensitivity, gradient-flow preservation, and synaptic flow, respectively (Lee et al., 2019b; Wang et al., 2020; Tanaka et al., 2020). More recent methods explicitly consider trainability: ProsPr differentiates through early optimization steps, while NTK-SAP seeks to preserve neural tangent kernel spectra (Alizadeh et al., 2022; Wang et al., 2023). These developments motivate a trajectory-aware treatment of mask selection. We study a specific target within this perspective: the coordinate ranking produced by a prescribed finite-time IMP discovery process.

Mask identity and sparse trainability capture different properties. Initialization-time pruning can depend strongly on layerwise sparsity and can behave similarly after within-layer mask shuffling (Frankle et al., 2021; Su et al., 2020). Recent work further studies sparse topology, effective random masks, and parameter signs (Pham et al., 2023; Gadhikar et al., 2023; 2025; Oh et al., 2025). Meanwhile, Kumar et al. (2024) relate sparse model capacity to the information shared by a mask and the data. These findings motivate evaluating both which coordinates a predictor selects and whether the resulting subnetwork trains effectively under matched initialization and retraining protocols.

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.

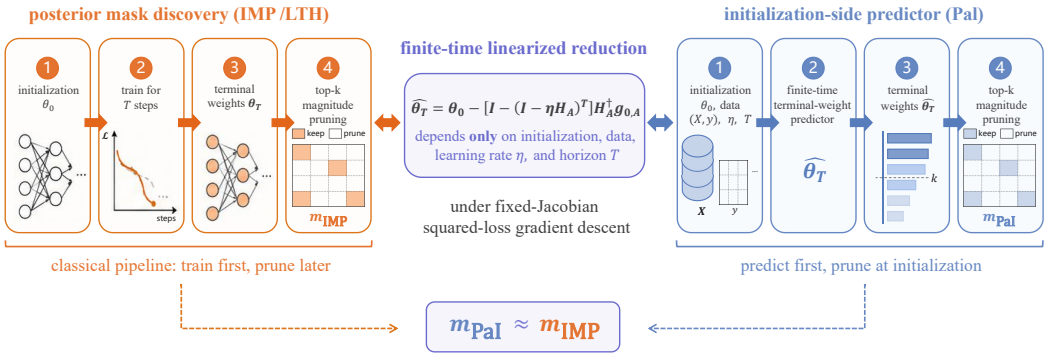

Figure 1: IMP prunes finite-time terminal weights; the predictor computes their ranking from initialization-side quantities. Under matched fixed-Jacobian squared-loss dynamics and stable top-k boundaries, one-shot and iterative masks agree. Nonlinear experiments test the scores with ERK layer allocation during whole-network sparse retraining from the original initialization.

Our starting point is fixed-Jacobian squared-loss dynamics, a setting connected to neural tangent kernels and linearized training (Jacot et al., 2018; Lee et al., 2019a). The known finite-time quadratic-GD response retains the learning rate and horizon through a spectral filter (Engl et al., 1996; Yao et al., 2007; Ali et al., 2019). We use this response to express IMP selection as initialization-side terminal ranking, as illustrated in Figure 1. Applying IMP's pruning rule gives an exact correspondence, which extends recursively when each round uses its own predicted active set. The formulation also exposes a discrete question beyond predicting terminal loss: whether a change in the trajectory changes which magnitudes cross the pruning boundary.

An initialization-side formulation is about when information is available, not about ignoring the training objective. The inputs, labels, initialization, and discovery schedule jointly specify the terminal-ranking target. Iterative discovery adds an important constraint: pruning alters the residual and the curvature seen by surviving coordinates, so a single ranking computed for the dense network is not the whole prediction problem. Each predicted active set must define the next round, without reading a mask obtained from trained weights. Ranking fidelity also differs from terminal-loss fidelity. Top-k selection is discontinuous at its boundary: exchanging two nearly equal magnitudes can change the selected subnetwork while barely affecting a continuous error measure. We therefore ask both whether a prescribed coordinate decision can be reproduced and whether the resulting mask supports useful sparse training. This separation provides a common framework for the exact linearized comparison and the nonlinear functional experiments.

We examine this ranking through correlated-feature support recovery and finite-time diagnostics, and validate recovery in functional whole-network fixed-Jacobian models. We then evaluate the scores in fully nonlinear train-prune-rewind-retrain experiments. For this extension, the predictor combines initialization-side scores with architecture-based ERK layer budgets (Evci et al., 2020); its masks are generated without trained weights or reference IMP masks. Sparse accuracy measures functionality, while repeated IMP discoveries contextualize coordinate agreement. The exact result concerns matched linearized dynamics, and the nonlinear experiments assess the usefulness of the resulting scores beyond those dynamics under the same sparse retraining protocol.

Our contributions are summarized as follows:

- We formulate posterior IMP mask selection as finite-time terminal-ranking prediction from initialization-side quantities, and establish when this representation reproduces both one-shot and iterative linearized IMP masks.
- We show that the predictor recovers data-dependent teacher supports in correlated-feature tasks where the tested static scores degrade, and use horizon, learning-rate, and loss diagnostics to examine which training details affect mask selection.
- We evaluate initialization-side masks through whole-network nonlinear sparse retraining and stochastic IMP self-consistency, demonstrating functional transfer in several LeNet and ResNet settings. A matrix-free implementation enables these evaluations without explicit Jacobians, Hessians, or pseudoinverses.

2 Background and Related Work

2.1 Sparse Subnetworks and IMP

Pruning criteria range from local second-order loss sensitivity (LeCun et al., 1989; Hassibi & Stork, 1992) to weight magnitudes used in compression pipelines (Han et al., 2015; 2016). The Lottery Ticket Hypothesis studies subnetworks together with their initial weights (Frankle & Carbin, 2019); subsequent work examines rewinding, optimization stability, transfer, and parameter signs (Renda et al., 2020; Frankle et al., 2020; Morcos et al., 2019; Zhou et al., 2019). Recent work aligns lottery-ticket masks to new initializations through weight symmetry (Adnan et al., 2025) and couples sparsity with flatness (Lee et al., 2025). These properties complement the sparse architecture itself (Liu et al., 2019). Our target is the mask selected by a specified IMP procedure. Recovering this mask and establishing that its associated ticket trains well are distinct questions, motivating separate measurements of coordinate agreement and sparse accuracy.

2.2 Pruning at Initialization

SNIP, GraSP, and SynFlow select connections using sensitivity or flow criteria (Lee et al., 2019b; Wang et al., 2020; Tanaka et al., 2020). ProsPr introduces foresight through meta-gradients over early training, and NTK-SAP aligns training dynamics through kernel-spectrum preservation (Alizadeh et al., 2022; Wang et al., 2023). DPaI optimizes differentiable masks under a node-path balance principle (Xiang et al., 2025), while recent work evaluates scalable second-order criteria for PaI (Navarrete et al., 2026). We instead target the terminal magnitude ranking of a prescribed finite-time trajectory, including its learning rate and horizon.

Layerwise allocation and sparse topology also affect PaI performance (Su et al., 2020; Frankle et al., 2021; Pham et al., 2023). Random sparse constructions can contain trainable subnetworks under appropriate width conditions (Gadhikar et al., 2023). Orthogonal initialization improves static sparse training (Nowak et al., 2024), while recent reparameterization work highlights the role of signs (Gadhikar et al., 2025). We therefore distinguish layer allocation from within-layer scoring; the nonlinear extension uses Erdős–Rényi–Kernel (ERK) allocation as in Evci et al. (2020). The information-theoretic account of Kumar et al. (2024) further motivates making data dependence explicit: our predictor uses both inputs and labels at initialization.

2.3 Linearized Training and Finite-Time Dynamics

Neural tangent kernel and lazy-training analyses characterize regimes in which neural networks follow approximately fixed-feature dynamics (Jacot et al., 2018; Lee et al., 2019a; Chizat et al., 2019). We take the fixed-Jacobian model as an explicit reference system. For squared loss, its finite-time response belongs to classical spectral filtering and early-stopping theory (Engl et al., 1996; Yao et al., 2007; Ali et al., 2019). This known response supplies the terminal weights. Our mask-level formulation then tracks their coordinate ordering, the pruning boundary, and the changing active subspace across IMP rounds. Finite-time filtering matters because convergence to a least-squares solution need not preserve the intermediate magnitude ranking.

3 Prerequisites

3.1 Masks and Iterative Magnitude Pruning

Let $\theta_0 \in \mathbb{R}^p$ be the initialization and $m \in \{0,1\}^p$ a mask with active set $A(m) = \{i : m_i = 1\}$. Write $\mathrm{TopK}_A(s, K)$ for the indices of the K largest scores in A, with a fixed deterministic tie rule. We study IMP with initialization rewinding (Frankle & Carbin, 2019; Renda et al., 2020): each round trains the retained coordinates from $\theta_{0,A}$, keeps the largest terminal magnitudes, and resets the survivors to their original values. For a prescribed sequence of keep counts,

$$A_{c+1}^{\mathrm{IMP}} = \mathrm{TopK}_{A_c^{\mathrm{IMP}}}\left(\left|\theta_{T,A_c^{\mathrm{IMP}}}^{\mathrm{trained}}\right|, K_{c+1}\right), \qquad A_0^{\mathrm{IMP}} = \{1,\dots,p\}. \tag{1}$$

One-shot pruning is a single round. For clarity, the notation treats all coordinates as prunable; permanently retained parameters remain in the training active set and are excluded from top-k selection. Layerwise pruning applies the same selection rule separately within each layer.

For descending scores $s_{(1)} \geq \cdots \geq s_{(|A|)}$, define the boundary gap $\Delta_{A,K}(s) = s_{(K)} - s_{(K+1)}$, where $1 \leq K < |A|$. A positive gap gives a unique selected set and makes selection stable to sufficiently small score perturbations. Matching masks requires matching the discovery protocol, including the initialization, objective, training horizon, and pruning rule.

3.2 Fixed-Jacobian Squared-Loss Dynamics

Stack the training outputs and targets into vectors $f_0 = f_{\theta_0}(X)$ and $y \in \mathbb{R}^n$, where n counts scalar output entries under the mean-squared-loss convention. Fix $J = \nabla_\theta f_{\theta_0}(X) \in \mathbb{R}^{n \times p}$ throughout discovery, giving the affine model $f_{\text{lin}}(\theta) = f_0 + J(\theta - \theta_0)$ (Jacot et al., 2018; Lee et al., 2019a). Let J_A contain its active columns. Setting inactive weights to zero gives the round-specific residual and objective

$$\begin{aligned} r_{0,A} &= f_0 - y - J_{A^c}\theta_{0,A^c}, \\ L_A(\theta_A) &= \frac{1}{2n}\left\|r_{0,A} + J_A(\theta_A - \theta_{0,A})\right\|_2^2. \end{aligned} \tag{2}$$

The residual correction accounts for zeroing pruned coordinates in the fixed affine model; it vanishes in the initial dense round. Define

$$g_{0,A} = \frac{1}{n}J_A^\top r_{0,A}, \qquad H_A = \frac{1}{n}J_A^\top J_A. \tag{3}$$

Full-batch gradient descent with constant learning rate η follows

$$\theta_{t+1,A} = \theta_{t,A} - \eta\left[g_{0,A} + H_A(\theta_{t,A} - \theta_{0,A})\right], \qquad t = 0, \ldots, T-1. \tag{4}$$

The usual stable regime is $0 < \eta < 2/\lambda_{\max}(H_A)$ when $H_A \neq 0$ (Yao et al., 2007). Each round starts with zero displacement from $\theta_{0,A}$ and updates both $r_{0,A}$ and H_A for its active set. These quantities define the finite-time reference whose terminal ranking the predictor will reproduce.

4 Finite-Time Mask Prediction

We compute the terminal magnitudes used by IMP from the initialization-side quadratic system, then apply the same pruning rule. The prediction target is a specified finite-time trajectory, rather than a converged fit with a potentially different magnitude ordering.

4.1 Initialization-Side Terminal Ranking

For the active-set quantities in Eq. equation 3, define the finite-time filter

$$q_T(\lambda) = \begin{cases} [1 - (1 - \eta\lambda)^T]/\lambda, & \lambda > 0, \\ \eta T, & \lambda = 0. \end{cases} \tag{5}$$

The classical quadratic-GD response (Engl et al., 1996; Yao et al., 2007) gives

$$\widehat{\theta}_{T,A} = \theta_{0,A} - q_T(H_A)g_{0,A} = \theta_{0,A} - [I - (I - \eta H_A)^T]H_A^\dagger g_{0,A}. \tag{6}$$

Here $H_A^\dagger$ is the Moore–Penrose pseudoinverse. The equality holds because $g_{0,A} \in \text{range}(H_A)$, so the gradient has no component in its nullspace. The Jacobian depends on the initialization and inputs; $g_{0,A}$ additionally incorporates the targets and the residual correction for previously pruned coordinates relative to the same initialization-centered affine model.

The scores $s_{A,i} = |\widehat{\theta}_{T,A,i}|$ retain the initial weights as well as the finite-time displacement. Curvature couples coordinates through H_A, while η and T control the spectral response. Under stable dynamics, the displacement approaches $-H_A^\dagger g_{0,A}$ as T grows, but this limit can have a different magnitude ranking from the finite-time vector. The discrete mask therefore depends on more than the limiting training loss or the quality of the converged least-squares fit.

Gradient components along eigendirections with $\eta\lambda T \ll 1$ receive an approximately ηT displacement factor, whereas $|1 - \eta\lambda|^T \ll 1$ gives a factor close to $1/\lambda$. Coordinate magnitudes mix these differently filtered components with the initialization. Changing the horizon can therefore alter their ordering while keeping the active set and initial gradient fixed. After pruning, restricting J changes

this mixture again through H_A and the residual correction. The predictor consequently carries both the finite-time filter and the predicted active set into each round, rather than reusing the dense ranking or substituting a converged fit. Section 5.5 tests these ranking consequences through controlled horizon and learning-rate ablations.

4.2 Recursive Recovery and Boundary Stability

Starting from the dense set, prediction follows its own pruning recursion:

$$\widehat{A}_0 = \{1, \dots, p\}, \qquad \widehat{A}_{c+1} = \mathrm{TopK}_{\widehat{A}_c}\big(|\widehat{\theta}_{T,\widehat{A}_c}|, K_{c+1}\big). \tag{7}$$

At every round, the predicted set determines $r_{0,\widehat{A}_c}$, $g_{0,\widehat{A}_c}$, and $H_{\widehat{A}_c}$. Thus subsequent predictions do not require reference IMP masks or trained weights from earlier pruning rounds.

Proposition 1 (Matched mask recovery). For fixed-Jacobian squared-loss full-batch GD initialized at $\theta_{0,A}$, Eq. equation 6 equals the terminal weight vector after T steps. With the same keep counts and pruning rule, a positive top-k boundary gap implies exact one-shot mask recovery. If these conditions hold at every round, prediction and initialization-rewinding linearized IMP have identical active sets throughout the complete train-prune-rewind discovery sequence.

Proof. Writing $\delta_{t,A} = \theta_{t,A} - \theta_{0,A}$, Eq. equation 4 yields

$$\delta_{T,A} = -\eta \sum_{t=0}^{T-1} (I - \eta H_A)^t g_{0,A} = -q_T(H_A) g_{0,A}. \tag{8}$$

Identical terminal vectors give identical magnitude scores and, under a positive boundary gap, a unique selected set. For the iterative statement, both procedures begin with the same dense set. Equality at round c gives the same residual, restricted curvature, and finite-time response, hence the same next set. Induction completes the argument. With identical deterministic tie-breaking, exact score equality also resolves boundary ties under the same coordinate ordering in both procedures.

The boundary gap also quantifies the tolerance to approximate evaluation. Let $\Delta_{A,K} = \Delta_{A,K}(|\widehat{\theta}_{T,A}|)$ as defined in Section 3. Then

$$\|\widetilde{\theta}_{T,A} - \widehat{\theta}_{T,A}\|_\infty < \tfrac{1}{2}\Delta_{A,K} \quad \Longrightarrow \quad \mathrm{TopK}_A(|\widetilde{\theta}_{T,A}|, K) = \mathrm{TopK}_A(|\widehat{\theta}_{T,A}|, K). \tag{9}$$

Indeed, taking absolute values changes each score by at most the weight error, so the gap between any retained and discarded coordinate remains positive. Applied round by round, this condition preserves the recursive masks as well. It links numerical accuracy to the actual selection boundary; a small relative ℓ_2 error alone is not a certificate of mask equality.

4.3 Matrix-Free Evaluation

The pseudoinverse in Eq. equation 6 is analytical notation. Equation equation 8 evaluates the same response using products $H_A v = J_A^\top (J_A v)/n$, implemented by Jacobian-vector and vector-Jacobian products (Paszke et al., 2019). Outputs can be processed in batches while accumulating the full-data product; this preserves the full-batch quadratic operator. Neither J_A nor H_A is materialized.

If one curvature product costs C_H, the recurrence costs $O(TC_H)$ per round and uses $O(p)$ working-vector storage, in addition to data and automatic-differentiation storage. An optional Lanczos evaluation (Golub & Loan, 2013) constructs an orthonormal basis Q_r of $\mathrm{span}\{g, Hg, \dots, H^{r-1}g\}$ and evaluates

$$q_T(H)g \;\approx\; \|g\|_2 Q_r q_T(B_r) e_1, \qquad B_r = Q_r^\top H Q_r. \tag{10}$$

This retains the target horizon T while using r curvature products. With full reorthogonalization, its cost is $O(rC_H + pr^2 + r^3)$ and its basis storage is $O(pr)$. Mask fidelity is measured against the full recurrence, with Eq. equation 9 giving a sufficient preservation condition.

4.4 TRANSFER TO NONLINEAR SPARSE TRAINING

For nonlinear experiments, the predictor still uses an initialization-centered, fixed-Jacobian squared-loss surrogate, with the model in evaluation mode during score computation. We combine its terminal scores with ERK layer allocation (Evci et al., 2020). For a weight tensor of shape $(d_{\ell,1}, \ldots, d_{\ell,m_\ell})$ and size $P_\ell = \prod_j d_{\ell,j}$, define

$$a_\ell = \frac{\sum_j d_{\ell,j}}{P_\ell}, \qquad \rho_\ell = \min(1, \epsilon a_\ell), \qquad \sum_\ell P_\ell \rho_\ell = K. \tag{11}$$

The desired counts $P_\ell \rho_\ell$ are apportioned to integer budgets summing to K, subject to the current active-layer capacities. Each layer retains its highest predicted magnitudes. This defines the nonlinear predictor: the budgets depend on architecture and the pruning schedule, while the within-layer scores depend on initialization-side data and dynamics.

After discovery, the mask is applied to the original initialization and trained with the nonlinear reference's sparse training protocol. All convolutional and linear weights are eligible for pruning; biases and normalization parameters remain dense. No trained checkpoint or IMP mask enters predictor discovery. The exact correspondence above applies to matched quadratic dynamics and pruning rules; the nonlinear comparison evaluates sparse functionality against cross-entropy IMP.

5 EXPERIMENTS

We test matched linearized IMP recovery, data-dependent mask structure, and nonlinear sparse functionality on MNIST (LeCun et al., 1998), CIFAR-10, and CIFAR-100 (Krizhevsky & Hinton, 2009), using MLPs, LeNet5, ResNets (He et al., 2016), and VGG11-BN (Simonyan & Zisserman, 2015).

For predicted and reference supports S, R, overlap is $O = |S \cap R|/|R|$ and Jaccard is $J = |S \cap R|/|S \cup R|$. References are linearized IMP, teacher support, or nonlinear IMP as specified. We report test top-1 accuracy (%), averaging repeated runs where available; figure captions specify variability.

5.1 RECOVERY IN FIXED-JACOBIAN MODELS

Controlled recovery. The binary CIFAR-10 MLP sweep spans seven depth/width configurations and pruning rates of 70, 80, 90, 95, and 98%. Across all 35 configurations, one-shot and iterative prediction give exactly equal masks ($O = J = 1$), with relative terminal-weight error at most 7.06×10^{-8}. This verifies finite-time evaluation and recursive active-set recovery against matched full-batch gradient descent on the same fixed-Jacobian squared-loss objective.

Whole-network functional validation. Table 1 includes all weight layers in the Jacobian and global pruning. Random-initialization MLPs use full datasets; dkwr denotes depth k and hidden width r. MNIST d3w1024 retains 91.66/91.04/90.04% accuracy at 70/90/95% pruning. Agreement above 99.99% therefore coexists with useful whole-network accuracy at high sparsity. The ResNet20/32 rows instead fix the Jacobian at supervised dense checkpoints, including the backbone in prediction and pruning. They test linearized fine-tuning at learned representations, not prediction from random initialization. Horizons and target conventions appear in Appendix A.

The two initialization regimes answer complementary questions. The MLP rows establish functional recovery before feature learning, while the ResNet rows test the active-set account around useful learned representations. In both cases, the reference is the corresponding affine model, so agreement is not confounded with switching from linearized to nonlinear training.

5.2 DATA-DEPENDENT SUPPORT UNDER CORRELATED FEATURES

An AR(1) linear teacher has 200 features, 2,000 training examples, and covariance $\Sigma_{ij} = \rho^{|i-j|}$. Prediction selects K coordinates using four squared-loss pruning rounds, $\eta = 0.02$, and $T = 600$. Three-seed sweeps vary correlation, sparsity, noise, or signal around $\rho = 0.9$, $K = 20$, zero noise, and unit signal, with teacher support used only as the evaluation reference.

Table 1: Whole-network fixed-Jacobian mask agreement with matched linearized IMP and affine-model accuracy (%). Triplets follow 70/90/95% pruning; single values apply to all three. MLPs use random initialization; ResNets use supervised warm starts. Unrounded overlap and Jaccard exceed 99.99% in these runs and are displayed to two decimals.

Dataset / model	Pruning	Dense accuracy	IMP / Predictor accuracy	Overlap	Jaccard
Whole-network fixed-Jacobian models: random initialization					
MNIST / MLP, d2w2048	70/90/95%	84.20	77.73 / 50.32 / 37.06	1.00	1.00
MNIST / MLP, d3w1024	70/90/95%	92.24	91.66 / 91.04 / 90.04	1.00	1.00
MNIST / MLP, d5w512	70/90/95%	88.21	87.54 / 85.55 / 84.54	1.00	1.00
CIFAR-10 / MLP, d3w1024	70/90/95%	48.31	47.92 / 43.25 / 38.42	1.00	1.00
CIFAR-10 / MLP, d4w768	70/90/95%	45.36	44.91 / 39.31 / 37.66	1.00	1.00
Whole-network linearized fine-tuning: supervised warm starts					
CIFAR-10 / ResNet20	70/90/95%	92.08	87.86 / 74.64 / 64.66	1.00	1.00
CIFAR-10 / ResNet32	70/90/95%	93.06	90.88 / 80.96 / 75.00	1.00	1.00

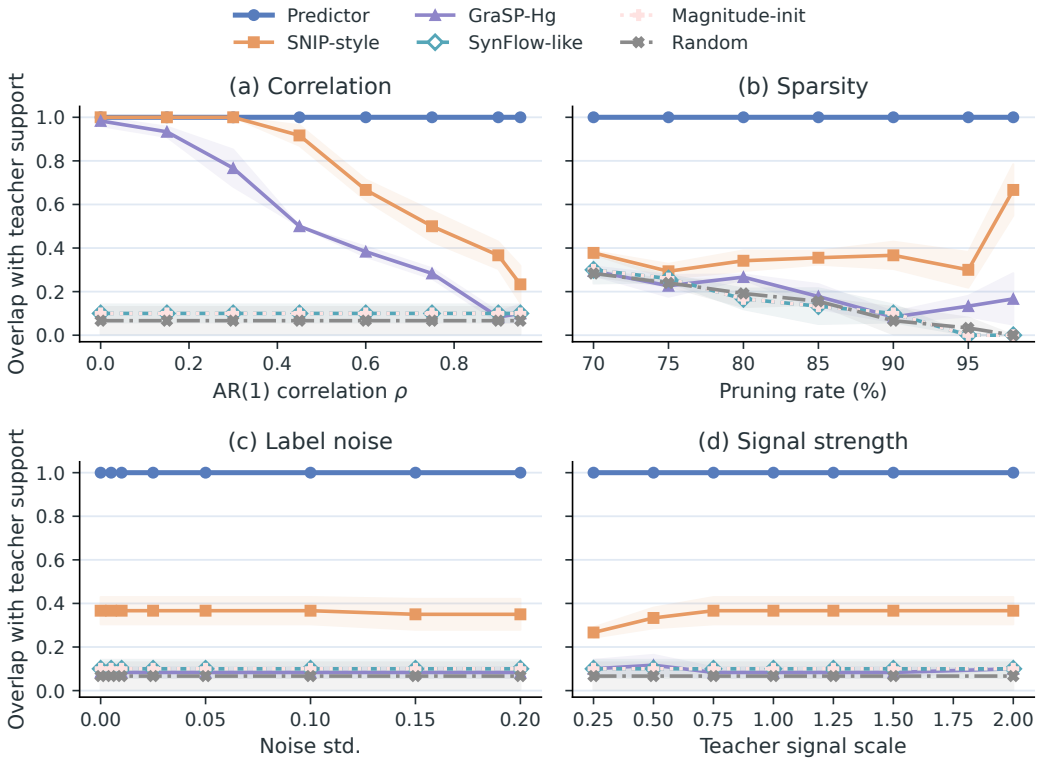

Figure 2: AR(1) teacher-support overlap (mean and population SD, three seeds). Panels vary (a) feature correlation, (b) pruning rate, (c) label noise, and (d) teacher signal strength around $\rho = 0.9$, 90% pruning, zero noise, and unit signal. Methods share data and keep counts; in (b), teacher support size changes with the pruning budget. Shading shows variability across seeds.

Figure 2 measures teacher-support recovery, not merely IMP agreement. The predictor achieves 1.00 throughout the recorded sweeps; at $\rho = 0.95$, SNIP-style sensitivity (Lee et al., 2019b) and GraSP-Hg (Wang et al., 2020) reach 0.233 and 0.083. Other controlled baselines are initialization magnitude, random pruning, and a SynFlow-like proxy (Tanaka et al., 2020) that reduces to initialization magnitude in this linear task. The finite-time response identifies label-dependent support despite correlation that degrades these static scores. This tests the complete trajectory score, not curvature in isolation from the horizon, residual, and initialization.

Because student weights begin with equal absolute magnitude, magnitude-only pruning has no teacher-dependent ordering. The predictor instead uses the label residual together with coupled

Table 2: Whole-network nonlinear sparse accuracy (%) after initialization rewinding. Triplets follow 70/90/95% pruning; dense accuracy is the unpruned reference. IMP and predictor masks use matched nonlinear retraining protocols. Values summarize the available runs.

Dataset / model	Pruning	Dense accuracy	IMP accuracy	Predictor accuracy
MNIST / LeNet5	70/90/95%	99.21	99.15 / 99.09 / 98.96	99.17 / 98.84 / 98.43
MNIST / LeNet5-Mini	70/90/95%	99.07	98.58 / 93.26 / 88.53	98.59 / 96.51 / 90.64
CIFAR-10 / LeNet5-CIFAR	70/90/95%	74.41	70.57 / 67.28 / 10.00	70.92 / 63.40 / 57.17
CIFAR-10 / ResNet20 (0.25×)	70/90/95%	81.98	76.74 / 65.85 / 59.84	76.12 / 65.35 / 58.16
CIFAR-10 / ResNet20	70/90/95%	92.45	90.53 / 87.10 / 84.40	90.73 / 87.50 / 83.52
CIFAR-10 / ResNet32	70/90/95%	93.34	91.72 / 89.22 / 86.50	91.51 / 89.32 / 86.36
CIFAR-10 / VGG11-BN	70/90/95%	92.51	92.37 / 91.67 / 90.63	91.76 / 90.72 / 89.47
CIFAR-100 / LeNet5-CIFAR	70/90/95%	39.51	35.18 / 29.70 / 1.00	36.98 / 28.00 / 20.78
CIFAR-100 / ResNet20	70/90/95%	67.27	65.29 / 55.45 / 48.46	65.13 / 55.72 / 47.11

Table 3: Nonlinear whole-network mask agreement at fixed initialization; triplets follow 70/90/95% pruning. IMP–IMP averages 45 pairs from ten discoveries; predictor–IMP compares one predicted mask with those ten. Pairs are descriptive, not independent accuracy replicates.

		IMP–IMP		Predictor–IMP	
Dataset / model	Pruning	Overlap	Jaccard	Overlap	Jaccard
MNIST / LeNet5	70/90/95%	0.491 / 0.239 / 0.173	0.325 / 0.136 / 0.095	0.524 / 0.287 / 0.170	0.355 / 0.168 / 0.093
CIFAR-10 / LeNet5-CIFAR	70/90/95%	0.321 / 0.151 / 0.115	0.191 / 0.082 / 0.061	0.336 / 0.172 / 0.122	0.202 / 0.094 / 0.065
CIFAR-10 / ResNet20	70/90/95%	0.341 / 0.131 / 0.077	0.205 / 0.070 / 0.040	0.326 / 0.130 / 0.083	0.195 / 0.069 / 0.043
CIFAR-100 / LeNet5-CIFAR	70/90/95%	0.343 / 0.152 / 0.112	0.207 / 0.082 / 0.060	0.401 / 0.170 / 0.128	0.251 / 0.093 / 0.069

feature geometry. Recovery across the noise and signal sweeps shows that this ordering remains informative over the tested perturbations, rather than only at one calibrated signal level.

5.3 Nonlinear Sparse Functionality and IMP Self-Consistency

Protocol. Whole-network IMP uses five train-prune-rewind rounds (Frankle & Carbin, 2019; Renda et al., 2020). The predictor uses 1,000 class-balanced examples, one-hot squared-loss targets, five rounds, and $T = 50$, with an initialization-fixed Jacobian and ERK layer budgets. Each mask is rewound and retrained with its reference's nonlinear cross-entropy optimizer, schedule, and seed on the full data. Learning rates and schedules appear in Appendix A.

Sparse accuracy. Table 2 shows functional transfer across model scales and datasets. On CIFAR-10 ResNet20 at 70/90% pruning, predictor accuracy is 90.73/87.50%, versus IMP's 90.53/87.10%. ResNet32 accuracy differs from IMP by at most 0.21 percentage points across all three pruning rates. At 95%, the predictor retains 98.43% on MNIST LeNet5 and 86.36% on CIFAR-10 ResNet32. The isolated chance-level CIFAR LeNet5 IMP runs do not establish stable predictor superiority. These comparisons support useful sparse tickets, not exact nonlinear coordinate recovery.

Coordinate agreement in context. Ten nonlinear IMP discoveries share initialization but vary training order (Table 3). At 90% pruning on ResNet20, IMP–IMP and predictor–IMP overlap are 0.131 and 0.130; Jaccard values are 0.070 and 0.069. MNIST LeNet5 overlaps are 0.239 and 0.287. A stochastic IMP mask is therefore not a unique coordinate target. Self-consistency contextualizes agreement, while paired retraining separately measures functionality; these descriptive mask comparisons are neither upper bounds nor statistical equivalence tests.

Separating allocation and ranking. Allocation-matched random masks use identical layerwise keep counts and uniformly sampled coordinates. At 70/90/95% pruning, three-seed LeNet5 mean overlaps with seed-matched IMP are 0.514/0.258/0.157 for the predictor and 0.399/0.188/0.123 for random masks. Allocation-only expectations are 0.400/0.190/0.123, computed as $K_{\mathrm{IMP}}^{-1} \sum_\ell k_\ell r_\ell / P_\ell$, with r_ℓ denoting IMP's layerwise keep count. The predictor thus en-

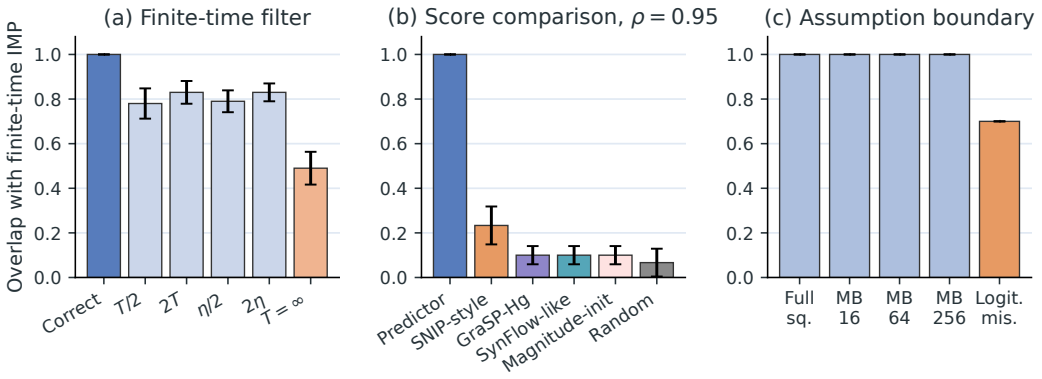

Figure 3: Mask-overlap diagnostics (mean and population SD). (a) Matched and mismatched finite-time predictions against IMP at $T = 30$ (five seeds). (b) Teacher-support comparison at $\rho = 0.95$ (three seeds). (c) Squared-loss batch variants and logistic-loss mismatch at $T = 600$ (three seeds).

riches coordinate agreement beyond allocation. Random-mask accuracies of 99.17/98.92/98.67% remain close to the predictor's (Table 2); this control does not show a general accuracy advantage from ranking alone under matched retraining and fixed layerwise keep counts.

5.4 Numerical Fidelity of Matrix-Free Prediction

On double-precision LeNet5 with 2, 4, 8, and 16 MNIST examples, $T = 30$, and $\eta\lambda_{\max} = 1$, matrix-free and explicit spectral evaluation differ by 1.49×10^{-16}–5.05×10^{-15} in relative terminal-weight error. All four 95%-pruned masks are bitwise equal ($O = J = 1$). These checks validate the operator implementation used in the whole-network experiments, without storing the Jacobian or forming a dense pseudoinverse, even when the quadratic operator is rank deficient.

5.5 Finite-Time and Loss Diagnostics

Figure 3(a) fixes data and initialization and compares predictions against IMP at $T = 30$. Across five seeds, matched overlap is 1.00; halving or doubling the horizon or learning rate gives 0.78–0.83, while the converged predictor gives 0.49 ± 0.07. Thus finite-time information changes the discrete support even with identical data. Panel (b) retains the high-correlation teacher comparison, complementing Figure 2 under the same strongly correlated teacher-feature setting.

Panel (c) tests loss and batching at $T = 600$. Full-batch and mini-batch squared-loss runs match the full-batch predictor's masks in this tested high-margin task. With shared sign-thresholded classification targets, agreement between logistic training and squared-loss prediction falls to 0.70. Observed batching robustness therefore does not remove dependence on the loss vector field.

6 Conclusion

We studied when the terminal magnitude ranking used by IMP can be determined from initialization-side quantities. The known finite-time quadratic-GD response provides a controlled reference: under matched fixed-Jacobian squared-loss dynamics and stable pruning boundaries, initialization-side prediction reproduces both one-shot and iterative IMP masks. This formulation makes the training horizon, active-set curvature, and selection boundary explicit in the mask-discovery problem.

The experiments connect this correspondence to data-dependent structure and sparse functionality. Correlated-feature tasks demonstrate teacher-support recovery, while horizon and loss diagnostics show why the specified dynamics matter. Whole-network fixed-Jacobian models provide functional validation, and nonlinear retraining evaluates useful tickets beyond the exact regime. Repeated IMP discoveries contextualize coordinate agreement, and matrix-free evaluation avoids explicit pseudoinverses. Together, these findings establish a conditional connection between posterior pruning and initialization-side prediction. Extending this account to feature-learning dynamics and identifying when ranking information improves allocation-matched sparse training are natural next steps.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing the manuscript, including improving English expression, clarity, and organization. We have reviewed the AI-assisted writing for clarity and consistency with the intended scientific content. We take responsibility for the final content of this work, including text produced with the aid of generative AI.

REFERENCES

Mohammed Adnan, Rohan Jain, Ekansh Sharma, Rahul Krishnan, and Yani Ioannou. Sparse training from random initialization: Aligning lottery ticket masks using weight symmetry. In *ICML*, 2025.

Alnur Ali, J. Zico Kolter, and Ryan J. Tibshirani. A continuous-time view of early stopping for least squares regression. In *AISTATS*, 2019.

Milad Alizadeh, Shyam A. Tailor, Luisa M. Zintgraf, Joost van Amersfoort, Sebastian Farquhar, Nicholas D. Lane, and Yarin Gal. Prospect pruning: Finding trainable weights at initialization using meta-gradients. In *ICLR*, 2022.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? In *MLSys*, 2020.

Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In *NeurIPS*, 2019.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and Its Applications*. Kluwer Academic Publishers, Dordrecht, 1996.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *ICML*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*, 2020.

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

Advait Gadhikar, Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Sign-In to the lottery: Reparameterizing sparse training. In *NeurIPS*, 2025.

Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *ICML*, 2023.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4 edition, 2013.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *NeurIPS*, 1992.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *JMLR*, 22 (241):1–124, 2021.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

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

Tanishq Kumar, Kevin Luo, and Mark Sellke. No free prune: Information-theoretic barriers to pruning at initialization. In *ICML*, 2024.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *NeurIPS*, 1989.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998.

Dongyeop Lee, Kwanhee Lee, Jinseok Chung, and Namhoon Lee. SAFE: Finding sparse and flat minima to improve pruning. In *ICML*, 2025.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*, 2019a.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *ICLR*, 2019b.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *ICLR*, 2019.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In *NeurIPS*, 2019.

Ivo Gollini Navarrete, Nicolas Mauricio Cuadrado, Martin Takáč, and Samuel Horváth. What scalable second-order information knows for pruning at initialization. In *CPAL (Proceedings Track)*, 2026.

Aleksandra Nowak, Łukasz Gniecki, Filip Szatkowski, and Jacek Tabor. Sparser, better, deeper, stronger: Improving static sparse training with exact orthogonal initialization. In *ICML*, 2024.

Junghun Oh, Sungyong Baik, and Kyoung Mu Lee. Find a winning sign: Sign is all we need to win the lottery. In *ICLR*, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Hoang Pham, The Anh Ta, Shiwei Liu, Lichuan Xiang, Dung Le, Hongkai Wen, and Long Tran-Thanh. Towards data-agnostic pruning at initialization: What makes a good sparse mask? In *NeurIPS*, 2023.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *ICLR*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Jingtong Su, Yihang Chen, Tianle Cai, Tianhao Wu, Ruiqi Gao, Liwei Wang, and Jason D. Lee. Sanity-checking pruning methods: Random tickets can win the jackpot. In *NeurIPS*, 2020.

Hidenori Tanaka, Daniel Kunin, Daniel Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS*, 2020.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *ICLR*, 2020.

Yite Wang, Dawei Li, and Ruoyu Sun. NTK-SAP: Improving neural network pruning by aligning training dynamics. In *ICLR*, 2023.

Lichuan Xiang, Quan Nguyen-Tri, Lan-Cuong Nguyen, Hoang Pham, Khoat Than, Long Tran-Thanh, and Hongkai Wen. DPaI: Differentiable pruning at initialization with node-path balance principle. In *ICLR*, 2025.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *NeurIPS*, 2019.

A APPENDIX

This appendix gives the implementation conventions used for the finite-time predictor. The algorithm is included only to make the active-set recursion explicit. The reference IMP mask is not used during prediction; it is used only for evaluation.

Algorithm 1 Finite-time mask prediction

Require: initialization θ_0, data (X, y), learning rate η, horizon T, pruning schedule $\{K_c\}_{c=1}^{C}$
Ensure: predicted active set A_C

$f_0 \leftarrow f_{\theta_0}(X)$; fix $J = \nabla_\theta f_{\theta_0}(X)$ at initialization
$A_0 \leftarrow \{1, \dots, p\}$
for $c = 0, \dots, C-1$ **do**
  $r_{0,A_c} \leftarrow f_0 - y - J_{A_c^c}\theta_{0,A_c^c}$
  $g_{0,A_c} \leftarrow J_{A_c}^\top r_{0,A_c}/n$; define $H_{A_c}v = J_{A_c}^\top(J_{A_c}v)/n$
  $u_0 \leftarrow 0$
  for $t = 0, \dots, T-1$ **do**
    $u_{t+1} \leftarrow u_t + \eta\big(g_{0,A_c} - H_{A_c}u_t\big)$
  end for
  $\widehat{\theta}_{T,A_c} \leftarrow \theta_{0,A_c} - u_T$
  $A_{c+1} \leftarrow \mathrm{TopK}_{A_c}\big(|\widehat{\theta}_{T,A_c}|, K_{c+1}\big)$
end for
return A_C

For one-shot pruning, $C = 1$. For iterative pruning, $C > 1$ and the residual, gradient, and curvature are updated using the original Jacobian restricted to the active set predicted at the previous round. This convention matches Section 4: the predictor starts from the dense active set and updates it by its own top-k decisions. The nonlinear predictor uses the layer budgets in Eq. equation 11 for within-layer selection; A_c^c denotes the complement of A_c.

A.1 NUMERICAL CONVENTIONS

All masks are unstructured parameter-level masks. Permanently retained coordinates enter the quadratic system but are excluded from ranking. Predicted and reference masks use the same keep counts and deterministic tie-breaking rule in exact-recovery comparisons. The matrix H_A can be singular. Since $g_{0,A}$ lies in its range, the nullspace does not contribute to the finite-time displacement. Matrix-free products $J_A^\top(J_A v)/n$ accumulate the full-data operator across batches. Equation equation 9 gives a sufficient condition for preserving the top-k boundary. Optional Lanczos evaluation retains the target horizon; terminal-weight error and mask agreement are evaluated separately.

A.2 PROTOCOL NOTES

Fixed-Jacobian models. The fixed-Jacobian experiments use linearized IMP with matched quadratic dynamics. MLPs use full training sets and 10,000 test examples. MNIST d2w2048 uses 100 steps with $\eta\widehat{\lambda}_{\max} = 1$; other MLPs use 300 steps with $\eta\widehat{\lambda}_{\max} = 1.5$, except CIFAR-10 d3w1024 uses 600. Warm-start ResNet20/32 use 5,000 training and 5,000 test examples, 30 discovery and 60 sparse fitting steps, $\eta\widehat{\lambda}_{\max} = 1$, and targets $y = f_0 + (\mathrm{onehot}(y_{\mathrm{class}}) - 1/C)$ with $C = 10$.

Teacher tasks. Initial weights are independently ± 0.01; the teacher has K nonzero coefficients with random signs. Horizon diagnostics use $\rho = 0.9$, 90% pruning, four rounds, $\eta = 0.02$, and reference $T = 30$.

Nonlinear training. Predictor discovery uses $\eta = 1/\widehat{\lambda}_{\max}$ from ten initialization-time power iterations. LeNet5 dense training, each IMP round, and retraining use 50 epochs, learning rate 0.05, tenfold decays at 30/40, and weight decay 10^{-4}. ResNet20 uses 200 epochs, learning rate 0.1, decays at 100/150, and weight decay 5×10^{-4}. Both use SGD, momentum 0.9, and batch size 128. Paired retraining shares initialization, optimizer, schedule, and seed. Repeated accuracy runs vary initialization; fixed-initialization IMP discoveries vary training order. Figure error bars use population SD.