PRUNE WHAT THE TEACHER CAN REPAIR: DISTILLATION-AWARE PRUNING FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper proposes a novel **D**istillation-**A**ware **P**runing (**DAP**) framework for addressing the teacher–student compatibility issue in large language models (LLMs). Unlike existing approaches to teacher–student mismatch, which typically assume a predefined student and subsequently adapt the distillation objective to bridge the resulting gap, we revisit the student construction problem through pruning, which provides a natural mechanism for preserving parameters and representations from its parent teacher. Motivated by this intuition, DAP therefore prunes for post-distillation quality rather than immediate post-pruning quality by preserving what the intended distillation process cannot recover and removing what it can repair. Across settings on Qwen3-8B and DeepSeek-R1-Distill-Llama-8B, multiple tasks, and both supervised and on-policy distillation, DAP consistently produces stronger post-distillation students than both independently constructed and conventionally pruned alternatives under matched parameter and training budgets.

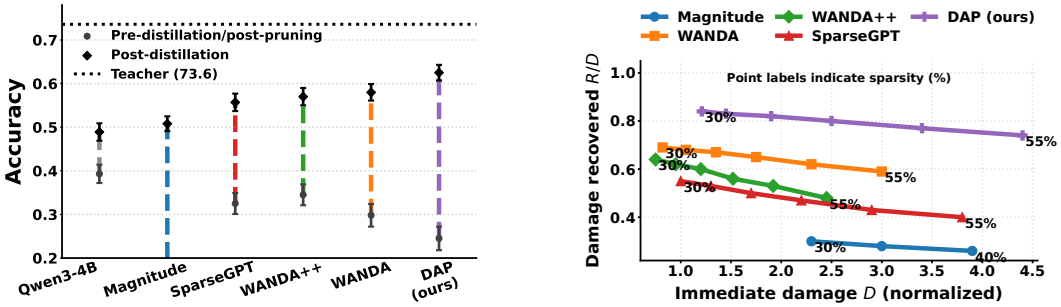
1 INTRODUCTION

Large language models (LLMs) have demonstrated impressive problem-solving capabilities across a wide range of domains, including coding (Nam et al., 2024), mathematical reasoning (Ahn et al., 2024), and planning (Wei et al., 2025). However, these advances have been accompanied by rapidly growing model sizes and computational costs, causing unavoidable challenges for low-latency inference and memory-efficient deployment, particularly in resource-constrained downstream scenarios (Yang et al., 2025b). Among various approaches to improving LLM efficiency, distillation has emerged as a widely adopted model compression technique, transferring knowledge from a high-capacity teacher to a smaller and more efficient student while preserving much of the teacher's performance (Hinton et al., 2015; Gu et al., 2024; Agarwal et al., 2024). Together with the proliferation of capable open-source LLMs (Grattafiori et al., 2024; Liu et al., 2024; Yang et al., 2025a), distillation provides a practical way of adapting powerful models to constrained deployment budgets. However, distillation effectiveness depends not only on the capacity of the student, but also on the compatibility between the teacher and student. Even under comparable model budgets, different teacher–student pairings can exhibit markedly different post-distillation performance (Zhang et al., 2025; Minegishi et al., 2026).

Teacher–student mismatch is a long-standing problem that has limited the effectiveness of knowledge distillation (Zhang et al., 2025; Cho & Hariharan, 2019). Existing approaches mitigate this mismatch by modifying the teacher to better suit a given student (Park & No, 2022; Li et al., 2024; Cao et al., 2026), selecting a more compatible student from a set of candidates (Dong et al., 2023; Lu et al., 2024; Chen et al., 2024), or adapting the distillation data to make knowledge transfer easier (Li et al., 2025; Liu et al., 2026b; Zhang et al., 2026). Although they deal with the problem at different levels, these approaches share a common premise: the student is first specified or selected, and distillation is then adapted around it. This leaves a more fundamental possibility largely unexplored—rather than asking how to better distill into a given student, can we construct the student itself from the teacher in a way that makes subsequent distillation more effective?

Model pruning provides a natural way to instantiate this idea. Instead of choosing an independently constructed smaller model, pruning derives a compact student directly from the dense teacher by removing parameters while retaining the important weights and much of the teacher's internal representational structure. The resulting sparse model can then be distilled to recover capabilities

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.



(a) MATH-500 (Hendrycks et al., 2021) accuracy before/after SFT distillation for pruned students (50% sparsity) and an independently trained Qwen3-4B student.

(b) Recoverability versus normalized immediate damage across different sparsity levels and pruning methods. Formal definitions are provided in Section 4.

Figure 1: **Pre-distillation quality does not determine post-distillation performance, with Qwen3-8B (Yang et al., 2025a) as the fixed distillation teacher.**

lost during pruning, as in existing prune–then–distill pipelines (Muralidharan et al., 2024; Sreenivas et al., 2024; Liu et al., 2026a). We call a student obtained by pruning a particular teacher a **descendant student**, and refer to the dense teacher from which it is pruned as its **parent teacher**. This construction couples student selection and distillation in a way that independently constructed students do not: the pruning decision determines not only what information is removed from the teacher, but also the state from which the teacher must subsequently recover that information through distillation. This raises a key question: *should a descendant student be selected by minimizing its immediate pruning damage, or by considering how recoverable that damage is under the intended teacher?*

Our empirical results in Fig. 1 support the latter view. Fig. 1a shows that students with similar parameter budgets can differ substantially after receiving the same distillation procedure, and that stronger post-pruning performance does not necessarily imply stronger post-distillation performance. Fig. 1b provides a clue to this discrepancy: pruning methods differ not only in the amount of damage they introduce, but also in the fraction of that damage that can subsequently be recovered. A pruning decision that appears worse immediately after compression may therefore become preferable once the intended distillation dynamics are taken into account.

We summarize our contributions as follows. First, we show that pruning decisions differ not only in the immediate damage they introduce, but also in how much of that damage subsequent distillation can repair. Second, we show that this recoverability depends on the downstream teacher and distillation procedure: changing either can change which pruning mask performs best. Third, we propose **DAP** (**D**istillation-**A**ware **P**runing), which selects pruning masks by estimating the damage expected to remain after distillation rather than immediate pruning damage alone. Using factorized curvature, DAP makes this criterion scalable without separately distilling candidate students. Across multiple parent models, supervised fine-tuning (SFT) and on-policy distillation (OPD), and both unstructured and semi-structured sparsity, DAP produces stronger post-distillation students than conventional pruning under matched sparsity and training budgets.

2 Related Work

LLM Distillation and NN Pruning Methods. Knowledge distillation (KD) (Hinton et al., 2015) is a widely used model compression technique that transfers capabilities from stronger teachers to smaller students. *Offline distillation* trains students on data generated by a different teacher, as exemplified by DeepSeek-R1 (Guo et al., 2025). *On-policy distillation* (Gu et al., 2024; Agarwal et al., 2024) instead trains on student-generated sequences to reduce train–inference mismatch while retaining an external teacher. More recently, *on-policy self-distillation* removes the external teacher by using different configurations of the same model as teacher and student, with OPSD (Zhao et al., 2026) as an early representative method. In another line of work, neural network pruning aims to identify and remove redundant parameters while preserving the performance of the original model.

Magnitude-based pruning, popularized by Han et al. (2015), has become a simple yet strong baseline and has been widely applied (Frankle & Carbin, 2018; Sun et al., 2024; Kolawole et al., 2024). Beyond magnitude, gradient-based methods estimate parameter importance using gradient information and have been extensively studied across model architectures (Das et al., 2023; Frantar & Alistarh, 2023; Yang et al., 2025c). More recently, graph-geometric approaches characterize neural networks through graph curvature and exploit such structural information for pruning (Tan et al., 2024; 2026). However, KD typically assumes a predefined student, while pruning optimizes immediate damage without considering post-distillation recovery. We bridge these two perspectives by selecting a student from the dense teacher that jointly minimizes pruning damage and maximizes distillation recoverability.

Bridging the Teacher–Student Gap. One line of work modifies the teacher while assuming a fixed student, for example by pruning the teacher or progressively adapting its capacity to ease knowledge transfer (Park & No, 2022; Li et al., 2024; Cao et al., 2026). A second line searches for students that are more suitable for distillation using compatibility metrics or architecture search (Dong et al., 2023; Lu et al., 2024), though such search is difficult to scale to modern LLMs. A third line improves teacher–student compatibility through data selection or synthesis (Li et al., 2025; Liu et al., 2026b; Zhang et al., 2026). More recently, pruning followed by distillation has become a practical recipe for LLM compression, where the pruned model serves as the student and the original model as the teacher (Muralidharan et al., 2024; Sreenivas et al., 2024; Bercovich et al., 2025; Liu et al., 2026a). However, existing methods still largely determine what to prune using pruning-specific heuristics or separate architecture search, whereas our method directly incorporates the subsequent distillation objective into the pruning criterion, making the pruning decision explicitly aware of the student's post-distillation performance.

3 Problem Formulation

Conventional pruning seeks a compressed model whose performance remains as close as possible to that of the dense model after pruning. Let π_{θ_T} denote a post-trained dense model, which we refer to as the *parent teacher*. Let z denote a model-wide binary pruning mask over the prunable parameters, and let $\mathcal{Z}_B$ denote the set of feasible masks satisfying a target compression budget B. Applying $z \in \mathcal{Z}_B$ to π_{θ_T} produces a *descendant student* $\pi_{\theta_T,z}$, whose parameters are obtained by pruning θ_T according to z, i.e., $\theta_T \odot z$, where $\odot$ denotes element-wise multiplication.

To formally measure behavior preservation under pruning, and abstracting away from the particular importance score or approximation used by a pruning method, the conventional pruning objective can therefore be written as

$$z^*_{\text{prune}} = \arg\min_{z \in \mathcal{Z}_B} L_{\text{ref},\theta_T}(\theta_T \odot z). \tag{1}$$

where L_{ref,θ_T} is defined as a reference objective using forward KL (Kim et al., 2026):

$$L_{\text{ref},\theta_T}(\theta) = \mathbb{E}_{\tau \sim \mathcal{D}_{\text{ref}}} D_{\text{KL}}\left(\pi_{\theta_T}(\cdot \mid \tau) \,\|\, \pi_\theta(\cdot \mid \tau)\right). \tag{2}$$

In Eq. 2, $\mathcal{D}_{\text{ref}}$ is an independent calibration set whose prompts are drawn from the target-task distribution. The dense parent π_{θ_T} defines the reference output distribution. That is, conventional pruning prefers the mask whose descendant deviates least from the dense parent after pruning. This objective, however, does not account for what happens when pruning is followed by distillation.

In the setting studied in this work, the descendant student $\pi_{\theta_T,z}$ is subsequently distilled from a parent teacher π_{θ_T} using a prescribed distillation algorithm $\Phi(\theta_T \odot z, \theta_T)$, which includes SFT (Hinton et al., 2015) and OPD (Agarwal et al., 2024). In $\Phi(\theta_T \odot z, \theta_T)$, the student is initialized with the pruned teacher parameters ($\theta_T \odot z$), and the pruning mask (z) is kept fixed throughout distillation. The behavior of the resulting student is therefore described by π_{θ_Φ}.

If the final model is evaluated only *after* this distillation procedure, the relevant pruning objective is instead

$$z^*_{\text{post}} = \arg\min_{z \in \mathcal{Z}_B} L_{\text{ref},\theta_T}\left(\Phi(\theta_T \odot z, \theta_T)\right). \tag{3}$$

The formulation above describes our default descendant setting, in which the dense parent is also the downstream distillation teacher. In Sec. 5.4, we additionally consider a cross-teacher extension

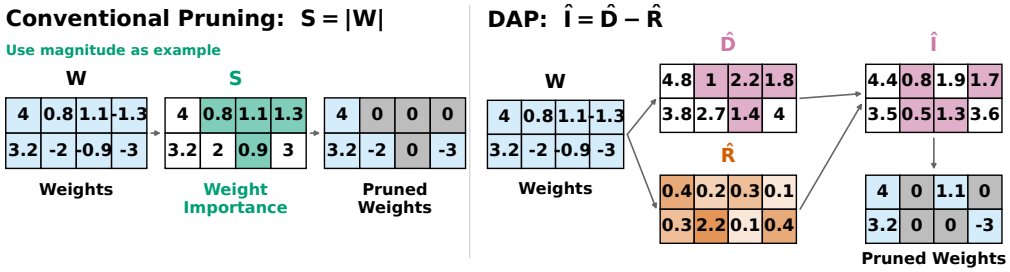


Figure 2: Illustration of DAP, compared with conventional pruning. Conventional pruning ranks candidates using saliency scores and removes the least important ones. In contrast, DAP considers both the predicted immediate pruning damage $\widehat{D}_i$ and the predicted recovery $\widehat{R}_i$, and ranks candidates by the irrecoverable damage $\widehat{I}_i = \widehat{D}_i - \widehat{R}_i$. Thus, DAP may prune a candidate with larger immediate damage if that damage is more recoverable.

in which the second argument of Φ is replaced by an intended teacher θ_Q, while $L_{\mathrm{ref},\theta_{\mathrm{T}}}$ remains anchored at the dense parent.

The distinction between Eqs. 1 and 3 is central to our setting. The mask that causes the least damage immediately after pruning need not perform best after distillation: different pruning perturbations may differ not only in their initial severity, but also in how effectively the intended distillation procedure can repair them. Consequently, the preferred pruning mask can depend on the intended distillation teacher, the distillation algorithm, and the resulting training dynamics.

Our goal is therefore to identify pruning decisions by their *post-distillation* impact rather than by their immediate pruning damage alone. Fig. 2 illustrates DAP and how it differs from conventional pruning methods. However, directly solving Eq. 3 would require running the prescribed distillation procedure for each candidate pruning decision. In the following sections, we develop DAP as a tractable approximation to this objective by explicitly modeling both the damage introduced by pruning and its subsequent recoverability through distillation.

4 Distillation-Aware Pruning

We begin by defining the quantity that an ideal DAP rule would minimize. Consider removing a component g from the dense parent by setting its parameters to zero. Let z_g be the corresponding binary mask, with zeros at the removed parameter entries and ones elsewhere. This removal induces the parameter perturbation $\delta_g = (z_g - \mathbf{1}) \odot \theta_T$. Thus, the pruned parameters are $z_g \odot \theta_T = \theta_T + \delta_g$. During subsequent distillation, z_g remains fixed, so only the surviving parameters are updated.

Definition 1 (Model Damage) *Given a post-trained dense model π_{θ_T} and a fixed pruning mask z_g, the **immediate pruning damage** is calculated by*

$$D_g = L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T + \delta_g). \tag{4}$$

*Then, the **remaining post-distillation damage (irrecoverable damage)** is defined as*

$$I_g = L_{\mathrm{ref},\theta_{\mathrm{T}}}(\Phi(\theta_T + \delta_g, \theta_T)) - L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T), \tag{5}$$

which is the remaining pruning damage after the intended distillation procedure, which updates the remaining parameters, and $L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T) = 0$ by construction.

Definition 2 (Recoverability) *Given a post-trained dense model π_{θ_T} and a fixed pruning mask z_g, let D_g and I_g be the corresponding terms. The **recoverability** is defined as*

$$R_g = D_g - I_g, \tag{6}$$

which represents the amount recovered through intended distillation.

Finally, we rewrite I_g as follows, as the DAP objective:

$$I_g = \underbrace{D_g}_{\text{immediate pruning damage}} - \underbrace{R_g}_{\text{distillation recovery}}. \tag{7}$$

An ideal distillation-aware pruning rule therefore minimizes I_g, rather than D_g alone. Conventional pruning favors removals that cause little immediate damage, whereas DAP favors damage that can be effectively recovered by the intended distillation procedure.

Exactly evaluating I_g is impractical, since it would require separately distilling and evaluating every candidate removal. We therefore derive a local approximation that decomposes the remaining damage into immediate damage and distillation recoverability in the following sections.

4.1 Local Post-Distillation Pruning Criterion

We approximate I_g around the dense parent. Let $H_{\mathrm{ref}} := \nabla^2_\theta L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T)$. Because $L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T) = 0$ and $\nabla_\theta L_{\mathrm{ref},\theta_{\mathrm{T}}}(\theta_T) = 0$, a second-order expansion gives

$$D_g \approx \frac{1}{2}\delta_g^\top H_{\mathrm{ref}} \delta_g =: \widehat{D}_g. \tag{8}$$

For the matched-teacher setting, let L_{KD} denote the intended distillation objective and $H_{\mathrm{KD}} := \nabla^2_\theta L_{\mathrm{KD}}(\theta_T)$. Assuming $\nabla_\theta L_{\mathrm{KD}}(\theta_T) \approx 0$, linearization around θ_T gives the mask-constrained local update

$$m_g \approx -\eta\, z_g \odot (H_{\mathrm{KD}} \delta_g).$$

Evaluating the residual perturbation $\delta_g + m_g$ under the quadratic reference model and retaining terms through first order in η yields

$$\widehat{I}_g = \underbrace{\frac{1}{2}\delta_g^\top H_{\mathrm{ref}} \delta_g}_{\widehat{D}_g} \underbrace{- \eta\, \delta_g^\top H_{\mathrm{ref}} \big[z_g \odot (H_{\mathrm{KD}} \delta_g)\big]}_{\widehat{R}_g}. \tag{9}$$

Thus, DAP ranks pruning decisions by estimated remaining damage $\widehat{I}_g = \widehat{D}_g - \widehat{R}_g$. The coefficient η controls the scale of the local recovery term; we parameterize it relative to the downstream learning rate and study its sensitivity in Appendix A.3.

This stationary-point derivation applies to the matched-teacher setting. For the cross-teacher experiments in Sec. 5.4, we use the same practical score with teacher-conditioned gradient second moments rather than interpreting Eq. 9 as an exact cross-teacher Taylor expansion. Full derivations are provided in Appendix A.5.

4.2 Scalable Curvature Approximation

Computing the curvature terms in Eq. 9 exactly is infeasible at LLM scale. We therefore discard cross-layer interactions and approximate each prunable linear layer $W \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$ using factorized activation and output-gradient second moments.

For $J \in \{\mathrm{ref}, \mathrm{KD}\}$, we use the sampled-target auxiliary loss $\ell_J^{\mathrm{curv}} = -\log \pi_\theta(y \mid \tau)$. For reference statistics, $\tau \sim \mathcal{D}_{\mathrm{ref}}$ and $y \sim \pi_{\theta_T}(\cdot \mid \tau)$; for recovery statistics, $\tau \sim \mathcal{D}_{\mathrm{KD}}$ and y is sampled from the intended downstream teacher. All gradients are evaluated through the dense parent. Let A_J and G_J denote the resulting activation and output-gradient second-moment factors. Their shared-weight aggregation and normalization are given in Appendix A.5.3.

The resulting layer block is approximated as

$$F_J^{(W)} \approx G_J \otimes A_J.$$

For the reference objective, $F_{\mathrm{ref}} = H_{\mathrm{ref}}$ in population; in the matched-teacher setting, F_{KD} likewise has the local Fisher/Hessian interpretation on the corresponding context distribution. For a different downstream teacher, F_{KD} is instead used as a teacher-conditioned gradient-second-moment surrogate.

To make the score scalable, we further diagonalize the output-gradient factor:

$$F_J^{(W)} \approx \mathrm{Diag}(\gamma_J) \otimes A_J, \qquad \gamma_J = \mathrm{diag}(G_J). \tag{10}$$

This removes interactions between output rows while retaining the full input factor A_J, restricting modeled recovery to surviving weights in the same output row. Storage per objective and layer is reduced to $O(d_{\mathrm{in}}^2 + d_{\mathrm{out}})$.

Importantly, procedure specificity enters through the procedure-conditioned contexts and sample/-token weighting used to construct $\mathcal{D}_{\mathrm{KD}}$, rather than merely through changing the KL direction on identical fixed contexts.

For a single removed weight W_{ab}, Eq. 9 then gives the practical DAP score

$$S_{ab} = \frac{1}{2} W_{ab}^2 \gamma_{\mathrm{ref},a} [A_{\mathrm{ref}}]_{bb} - \eta W_{ab}^2 \gamma_{\mathrm{ref},a} \gamma_{\mathrm{KD},a} \sum_{c \neq b} [A_{\mathrm{ref}}]_{bc} [A_{\mathrm{KD}}]_{cb}. \tag{11}$$

The first term estimates immediate damage and the second the signed recovery from surviving weights in row a. We prune the lowest-scoring weights under the compression constraint, treating individual scores as an additive approximation to joint mask selection. Appendix A.5 provides the complete single-weight and factorized derivations.

Algorithm 1 Approximation path of DAP.

1: **Recovery across the full model**
$\widehat{R}_i = \eta w_i^2 \sum_{j \neq i} [H_{\mathrm{ref}}]_{ij} [H_{\mathrm{KD}}]_{ji}$
$\Downarrow$ *discard cross-layer interactions*
2: **Recovery within each layer**
$H_J \approx \mathrm{blkdiag}_\ell\big(H_J^{(\ell)}\big)$
$\Downarrow$ *gradient second moment; factorize*
3: **Kronecker-factored layer curvature**
$H_J^{(\ell)} \approx G_J^{(\ell)} \otimes A_J^{(\ell)}$
$\Downarrow$ *diagonalize G; keep A full*
4: **Recovery within each output row**
$H_J^{(\ell)} \approx \mathrm{Diag}(\gamma_J^{(\ell)}) \otimes A_J^{(\ell)}$
$\Downarrow$ *subtract recovery from damage*
5: **Prune by remaining damage**
$S_{ab} = \widehat{D}_{ab} - \widehat{R}_{ab}$; prune the lowest scores.

5 Experiments

Our experiments are designed to test the central prediction of DAP: the best student after pruning need not be the best student after distillation, because pruning perturbations differ in how recoverable they are under the intended learning dynamics. We therefore organize the evaluation around four questions. **(Q1) Effectiveness:** does optimizing for predicted post-distillation damage produce stronger students than conventional pruning under matched budgets? **(Q2) Mechanism:** does recoverability explain the discrepancy between immediate and post-distillation quality, and can DAP predict it before training? **(Q3) Specificity:** does the preferred pruning decision change when the intended teacher or distillation procedure changes, as a dynamics-dependent notion of recoverability predicts? **(Q4) Scalability:** do the approximations required for LLM-scale pruning preserve enough of this signal to retain DAP's post-distillation advantage? Together, these experiments test the consequence, mechanism, specificity, and practical realizability of the proposed formulation.

5.1 Experimental Setup

Models and pruning. We evaluate DAP by constructing descendant students from dense LLMs under both unstructured and semi-structured weight pruning. We primarily study **Qwen3-8B** (Yang et al., 2025a) and additionally evaluate DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025) (hereafter, **DeepSeek-8B**) in Appendix A.2. Our primary setting prunes an 8B parent to 50% sparsity, yielding a student with an approximately 4B parameter budget. Unless otherwise specified, we prune the linear projections in the attention and MLP blocks while leaving the embeddings, normalization layers, and language-model head unchanged. For DAP, we use the scalable diagonal-G approximation in our main experiments and evaluate the full-G variant separately in Table 4. We set the local recovery coefficient to $\eta = 1.5\eta_0$, where η_0 is the learning rate of the corresponding downstream distillation procedure. η is selected on held-out validation data and its sensitivity is reported in Appendix A.3. We compare against standard pruning methods, including magnitude pruning (Han et al., 2015), SparseGPT (Frantar & Alistarh, 2023), WANDA (Sun et al., 2024), and WANDA++ (Yang et al., 2025c), as well as independently trained dense students of comparable size, which is a standalone 4B base model instruction-tuned on general-purpose instruction data that excludes the target evaluation tasks. All pruned students are constructed under the same sparsity budget and subsequently receive the same downstream distillation procedure and training budget. We defer full training details (optimizer, learning-rate schedule, batch size, and compute budget) to Appendix A.1.

Distillation and evaluation. We evaluate students after distillation using both SFT and OPD, allowing us to test whether DAP remains effective across different knowledge-transfer procedures. Our primary metric is downstream task performance after distillation, and we report mean ± std over three independently trained models.

Table 1: **Main post-distillation performance (%).** Avg. denotes the arithmetic mean across the three benchmarks; standard deviations are reported for individual benchmarks only.

Student Construction	Sparsity	SFT				OPD			
		MATH-500	StrategyQA	APPS	Avg.	MATH-500	StrategyQA	APPS	Avg.
Qwen3-8B (teacher)	0%	73.6 ± 1.1	74.5 ± 1.2	51.5 ± 1.8	66.5	73.6 ± 1.1	74.5 ± 1.2	51.5 ± 1.8	66.5
Qwen3-4B	0%	48.9 ± 1.9	51.8 ± 1.1	32.7 ± 2.1	44.5	53 ± 1.1	57.1 ± 1.4	36.8 ± 0.9	49.0
Magnitude	50%	50.8 ± 1.7	52.9 ± 0.8	34.0 ± 1.9	45.9	55 ± 1.2	58.3 ± 0.9	38.1 ± 1.5	50.5
SparseGPT		55.7 ± 2	57.8 ± 0.6	37.8 ± 1.7	50.4	60.1 ± 0.9	63.4 ± 0.9	42.2 ± 1.4	55.2
WANDA		58 ± 1.2	59.2 ± 0.8	39.4 ± 1.2	52.2	62.6 ± 0.8	65.1 ± 0.7	43.9 ± 1.2	57.2
WANDA++		57 ± 2	58.6 ± 0.9	38.7 ± 1.4	51.4	61.5 ± 1.1	64.4 ± 0.8	43.1 ± 1.4	56.3
DAP (Ours)		62.5 ± 1.8	61.9 ± 1.1	42.1 ± 1.2	**55.5**	67.4 ± 0.9	67.8 ± 0.9	46.9 ± 1.1	**60.7**
Magnitude	2 : 4	45.0 ± 2.0	48.0 ± 1.0	30.2 ± 2.2	41.1	49.5 ± 1.4	53.2 ± 1.1	34.0 ± 1.7	45.6
SparseGPT		51.6 ± 2.1	54.1 ± 0.8	35.2 ± 1.8	47.0	56.4 ± 1.1	59.5 ± 1.0	39.0 ± 1.5	51.6
WANDA		52.0 ± 1.4	54.4 ± 0.9	35.6 ± 1.4	47.3	56.8 ± 1.0	59.9 ± 0.8	39.3 ± 1.3	52.0
WANDA++		54.8 ± 1.9	56.2 ± 1.0	37.3 ± 1.5	49.4	59.6 ± 1.2	62.0 ± 0.9	41.2 ± 1.4	54.3
DAP (Ours)		59.8 ± 2.0	59.4 ± 1.2	40.1 ± 1.3	**53.1**	64.8 ± 1.0	65.4 ± 1.0	44.5 ± 1.2	**58.2**

Calibration data. All pruning methods use the same total calibration budget. We construct a shared calibration pool $\mathcal{D}_{\text{cal}} = \mathcal{D}_{\text{ref}} \cup \mathcal{D}_{\text{KD}}$ of 256 sequences, each with a sequence length of 4,096 tokens. For DAP, we split this pool evenly into two disjoint sets of 128 sequences each. The reference set $\mathcal{D}_{\text{ref}}$ is used to estimate the behavior that pruning should preserve: its prompts are drawn from the target-task distribution and its responses are generated by the dense parent. The distillation-statistics set $\mathcal{D}_{\text{KD}}$ is used to estimate how the intended distillation procedure will repair pruning damage; it follows the same data distribution and objective as the downstream distillation run, but is used only to compute curvature statistics and never for parameter updates. Conventional pruning baselines receive the full shared calibration pool $\mathcal{D}_{\text{cal}}$, so DAP does not use additional calibration data despite estimating two sets of statistics. All calibration data are disjoint from the downstream distillation training data and evaluation sets. Additional details are provided in Appendix A.4.

5.2 DAP Produces Better Post-Distillation Students

We first evaluate the primary question of our study: **Effectiveness**. We use the final post-distillation performance as the metric and compare it across each student produced under the same compression and training budgets. Table 1 reports results for both SFT and OPD, using Qwen3-8B as the parent teacher and evaluating on descendant students generated by pruning with 50% unstructured sparsity and 2:4 semi-structured sparsity (Zhou et al., 2021). Within each sparsity setting and distillation procedure, all pruning methods use the same parent, pruning budget, and downstream training budget (0.1B tokens). We additionally include a Generic Instruct 4B student to compare teacher-derived descendants against an independently constructed student of comparable capacity. Results for DeepSeek-R1-Distill-Llama-8B are provided in Appendix A.2.

Across all tasks, distillation procedures, and sparsity structures, DAP achieves the strongest post-distillation performance among pruning methods. At 50% unstructured sparsity, DAP improves average performance over the strongest baseline by $+3.3$ points with SFT and $+3.5$ with OPD; under the more restrictive 2:4 sparsity, the gains increase to $+3.7$ and $+3.9$ points, respectively. These consistent gains show that optimizing for recoverability remains beneficial even under structured pruning constraints.

5.3 Can DAP Predict Post-Distillation Damage?

We next address **Q2 (Mechanism)**. Fig. 1 established the first part of this question: pruning methods differ in recoverability, so minimizing immediate pruning damage D need not select the student with the smallest remaining damage I after distillation. We now test the second part: whether DAP can predict this post-distillation outcome before training. Fig. 3 compares measured remaining damage (solid lines) I with DAP's prediction (dashed lines) $\widehat{I} = \widehat{D} - \widehat{R}$ as sparsity increases. Up to 60% sparsity, the surrogate captures the method ordering well enough to consistently identify the student

with the lowest measured remaining damage, despite occasional inversions between closely matched baselines such as WANDA and WANDA++.

At higher sparsity, the approximation begins to break down. At 65% sparsity, DAP is still predicted to have the lowest remaining damage, whereas the measured ordering favors WANDA. This discrepancy persists at higher sparsities. This degradation is consistent with the local nature of our surrogate: larger pruning perturbations move farther from the dense-model expansion point, making neglected higher-order terms increasingly important. Together with Fig. 1, these results answer Q2: recoverability helps explain why immediate pruning quality can mispredict post-distillation quality, while DAP's local surrogate provides a useful signal for selecting the post-distillation student in the moderate-sparsity regime, with a clear breakdown under sufficiently severe pruning.

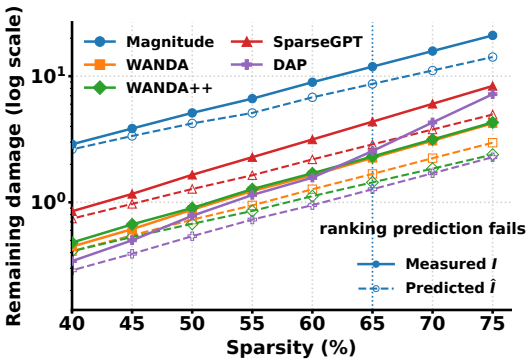


Figure 3: **DAP's post-distillation damage prediction across sparsity.**

5.4 Is Recoverability Specific to the Intended Distillation Dynamics?

Our formulation predicts that recoverability is not an intrinsic property of a DAP-pruned student, but depends on the downstream learning dynamics induced jointly by the intended teacher and the distillation procedure. Consequently, a pruning mask constructed to be recoverable under one downstream setting need not remain preferable when either the teacher or the distillation algorithm changes, which refers to the question **Specificity**.

Separating pruning quality from procedure-specific recovery compatibility. We first ask whether DAP identifies a generally stronger sparse student, or whether its advantage depends specifically on the distillation procedure that will subsequently repair the pruning perturbation. Holding the dense parent and teacher fixed, we construct two students at the same sparsity, $z_{\mathrm{DAP}}^{\mathrm{SFT}}$ and $z_{\mathrm{DAP}}^{\mathrm{OPD}}$, which differ only in the downstream procedure used to estimate the recovery term. We then distill each student with both SFT and OPD under identical teachers, prompts, and training budgets. As a procedure-independent control, we additionally include DAP with $\eta = 0$, which uses the same reference-curvature estimator but removes the recovery term entirely.

Table 2: **Matching pruning to downstream distillation.** Blue cells denote matched construction–distillation pairs; the bottom row reports the average matched-minus-mismatched improvement.

Pruning Criterion	**Distilled with SFT**				**Distilled with OPD**			
(statistics used for $\mathcal{D}_{\mathrm{KD}}$)	MATH-500	StrategyQA	APPS	Avg.	MATH-500	StrategyQA	APPS	Avg.
DAP, $\eta = 0$ (no recovery term)	52.0 ± 1.5	53.7 ± 0.9	34.8 ± 1.2	46.8	56.3 ± 1.3	59.1 ± 0.9	39.0 ± 1	51.5
DAP-SFT ($z_{\mathrm{DAP}}^{\mathrm{SFT}}$)	62.5 ± 1.8	61.9 ± 1.1	42.1 ± 1.2	**55.5**	64.6 ± 1.1	64.7 ± 1.0	43.9 ± 1.0	57.7
DAP-OPD ($z_{\mathrm{DAP}}^{\mathrm{OPD}}$)	59.8 ± 1.6	58.9 ± 1.0	39.2 ± 1.2	52.6	67.4 ± 0.9	67.8 ± 0.9	46.9 ± 1.1	**60.7**
Matched − mismatched (Avg.): SFT column $= +2.9$ OPD column $= +3.0$								

Table 2 shows a clear matching effect: SFT favors the SFT-conditioned mask, while OPD reverses the ordering and favors the OPD-conditioned mask, yielding average gains of 2.9 and 3.0 points over the mismatched mask respectively. Thus, DAP does not identify a universally stronger sparse model. It selects pruning perturbations that are better matched to the intended recovery dynamics.

Separating teacher lineage from recovery compatibility. We next examine what makes a descendant student compatible with its intended distillation teacher. A student pruned directly from its teacher enjoys a natural form of lineage compatibility: its surviving parameters are inherited from the same model that will subsequently distill it. DAP introduces a distinct source of compatibility by conditioning the pruning mask on the recovery dynamics of the intended teacher. We ask whether

this recovery-matching benefit remains useful even when the student is not a descendant of that teacher, and whether it can fully substitute for direct teacher–student lineage.

To disentangle these effects, we fix the distillation algorithm to SFT and consider both Qwen3-8B and DeepSeek-8B as dense parents. For each parent, we construct two DAP students using identical parent-specific reference statistics but recovery statistics induced by either Qwen or DeepSeek, yielding $z_{\mathrm{DAP}}^{\mathrm{Qwen}}$ and $z_{\mathrm{DAP}}^{\mathrm{DeepSeek}}$. We then distill both students with each teacher under identical training budgets. This design allows us to vary two forms of compatibility independently: whether the pruning mask is matched to the downstream teacher, and whether the student itself is descended from that teacher.

Table 3: **Teacher-conditioned recoverability across dense parents.** For each dense parent, we construct two DAP students using the same parent-specific reference statistics but recovery statistics induced by either Qwen3-8B or DeepSeek-8B. Each student is then distilled with both teachers using SFT. Blue cells denote matched recovery–distillation teacher pairs and red cells denote mismatched.

Dense Parent	DAP Recovery Teacher	Distilled with Qwen3-8B				Distilled with DeepSeek-8B			
	(used for $\mathcal{D}_{\mathrm{KD}}$)	MATH-500	StrategyQA	APPS	Avg.	MATH-500	StrategyQA	APPS	Avg.
Qwen3-8B	Qwen3-8B	62.5 ± 1.8	61.9 ± 1.1	42.1 ± 1.2	**55.5**	55.8 ± 1.8	55.9 ± 1.1	37.6 ± 1.3	49.8
	DeepSeek-8B	60.2 ± 1.7	60.0 ± 1.0	40.1 ± 1.2	53.4	58.4 ± 1.7	59.0 ± 1.1	40.4 ± 1.2	**52.6**
DeepSeek-8B	Qwen3-8B	60.2 ± 1.7	60.7 ± 1.0	40.8 ± 1.2	**53.9**	58.1 ± 1.7	57.9 ± 1.1	39.3 ± 1.2	51.8
	DeepSeek-8B	57.7 ± 1.7	58.4 ± 1.0	38.6 ± 1.2	51.6	60.9 ± 2.0	60.1 ± 0.9	41.9 ± 1.4	**54.3**
Matched − mismatched, Qwen parent				$+2.1$				$+2.9$	
Matched − mismatched, DeepSeek parent				$+2.3$				$+2.5$	

Table 3 reveals two complementary effects. First, for a fixed parent, matching the DAP recovery statistics to the downstream teacher consistently improves post-distillation performance, including when the student is descended from the other model. Thus, DAP provides a teacher-specific compatibility benefit that does not require direct parameter inheritance from the distillation teacher. Second, when the recovery statistics are matched, students descended from the distillation teacher still achieve the strongest performance. **Teacher-conditioned pruning therefore improves the compatibility of even a non-descendant student**, but does not fully replace the advantage of direct teacher–student lineage. Together, these results suggest that post-distillation performance reflects two distinct sources of compatibility: structural continuity inherited from the parent teacher and recoverability induced by matching the pruning decision to the intended downstream teacher.

5.5 Effect of Output-Gradient Diagonalization

We isolate the effect of output-gradient diagonalization within the Kronecker-factored curvature surrogate. The full-G variant uses $G_J \otimes A_J$, while diagonal-G removes modeled cross-output-row curvature interactions and retains the full activation factor A_J. Both variants use the same pruning rule, calibration data, sparsity, and downstream training configuration.

Criterion	Post-SFT Avg.
Kronecker full-G	58.9
Diagonal-G (DAP)	55.5

Table 4: Qwen3-8B, 50% unstructured sparsity.

Diagonalizing G reduces post-SFT average performance from 58.9 to 55.5, a 3.4-point reduction, while reducing storage for each output-gradient factor from $O(d_{\mathrm{out}}^2)$ to $O(d_{\mathrm{out}})$. The diagonal-G variant remains 3.3 points above the strongest conventional pruning baseline (55.5 versus 52.2) in the same unstructured SFT setting (Table 1).

6 Conclusion and Limitations

We introduced DAP, which selects pruning decisions based on their predicted post-distillation damage rather than immediate pruning quality alone. Our results show that recoverability depends on the intended teacher and learning dynamics, and that accounting for it yields stronger distilled students. DAP relies on local curvature approximations and is evaluated within prune-then-distill LLM settings; extending the principle to broader architectures, compression methods, and longer-horizon training dynamics remains future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for assisting with translation, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, interpreting results, generating synthetic data sets, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability. We have reviewed all AI-assisted work. For example, we verified and tested LLM-generated code for correctness with 2 authors, and all the authors have proof read the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies pruning and knowledge distillation to enable substantially smaller language models to preserve performance close to that of their dense counterparts. We believe improving this efficiency–performance tradeoff is important for the sustainable and broad development of modern artificial intelligence.

As increasingly capable models demand greater computational, memory, energy, and financial resources, efficiency has become a central challenge for both research and deployment. Effective model compression can reduce inference and deployment costs, lower energy consumption, and decrease dependence on large-scale computing infrastructure while preserving useful model capabilities. These improvements can make advanced AI systems more accessible to researchers, companies, and institutions with limited computational resources.

Beyond reducing resource consumption, efficient models can contribute to wider economic and technological development by lowering barriers to AI adoption, supporting deployment on resource-constrained hardware, and enabling advanced models to be used across a broader range of industries and applications. At a larger scale, reducing the computational requirements of AI can help improve the sustainability of continued AI development and broaden participation in technological innovation across organizations, economies, and nations.

Our work does not introduce new model capabilities or target sensitive applications; rather, it improves the efficiency with which existing capabilities can be retained and deployed. As more efficient deployment can also lower the cost of potentially harmful applications, compressed models should continue to be subject to appropriate safety evaluation and deployment safeguards.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details needed to reproduce our results in the main paper and appendix. We specify the pruning objective and its approximations, including the derivation of the local recovery criterion and the diagonal-G and full-G factorized scores. For all experiments, we report the parent models, pruning targets, sparsity constraints, calibration procedure, distillation settings, and evaluation benchmarks. DAP uses a fixed calibration budget of 256 sequences of 4,096 tokens, split evenly between reference and distillation-statistics estimation, while calibration-dependent baselines receive the same total budget. All calibration data are disjoint from downstream training and evaluation data. We provide optimizer and training hyperparameters, including learning rates, batch configuration, sequence lengths, distillation budgets, and OPD generation parameters. Experiments are run in BF16 on $4\times$H100 80GB GPUs using FSDP2, and reported results use three independent training seeds. All methods within each comparison use identical downstream data, optimization settings, sparsity budgets, and training compute, with pruning masks fixed throughout distillation.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammed Dabbah, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Netanel Haber, Ehud Dov Karpas, Roi Koren, Itay Levy, Pavlo Molchanov, Shahar Mor, Zach Moshe, Najeeb Nabwani, Omri Puny, Ran Rubin, Itamar Schen, Ido Shahaf, Oren Tropp, Omer Ullman Argov, Ran Zilberstein, and Ran El-Yaniv. Puzzle: Distillation-based NAS for inference-optimized LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 3806–3830, 2025.

Jincheng Cao, Fanzhi Zeng, Leqi Liu, and Aryan Mokhtari. Curriculum learning-guided progressive distillation in large language models. *arXiv preprint arXiv:2605.11260*, 2026.

Dong Chen, Ning Liu, Yichen Zhu, Zhengping Che, Rui Ma, Fachao Zhang, Xiaofeng Mou, Yi Chang, and Jian Tang. Epsd: Early pruning with self-distillation for efficient model compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11258–11266, 2024.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

Rocktim Jyoti Das, Mingjie Sun, Liqun Ma, and Zhiqiang Shen. Beyond size: How gradients shape pruning decisions in large language models. *arXiv preprint arXiv:2311.04902*, 2023.

Peijie Dong, Lujun Li, and Zimian Wei. Diswot: Student architecture search for distillation without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11898–11908. IEEE, 2023.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Junhyuck Kim, Jihun Yun, Haechan Kim, Gyeongman Kim, Joonghyun Bae, and Jaewoong Cho. Pruning and distilling mixture-of-experts into dense language models. *arXiv preprint arXiv:2605.28207*, 2026.

Steven Kolawole, Lucio Dery, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. Everybody prune now: Structured pruning of llms with only forward passes. *arXiv preprint arXiv:2402.05406*, 2024.

Shengrui Li, Junzhe Chen, Xueting Han, and Jing Bai. Nuteprune: Efficient progressive pruning with numerous teachers for large language models. *arXiv preprint arXiv:2402.09773*, 2024.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026a.

Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18041–18059, 2026b.

Liming Lu, Zhenghan Chen, Xiaoyu Lu, Yihang Rao, Lujun Li, and Shuchao Pang. Uniads: Universal architecture-distiller search for distillation gap. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14167–14174, 2024.

Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. *Advances in Neural Information Processing Systems*, 38:13907–13938, 2026.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems*, 37:41076–41102, 2024.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th international conference on software engineering*, pp. 1–13, 2024.

Jinhyuk Park and Albert No. Prune your model before distill it. In *European Conference on Computer Vision*, pp. 120–136. Springer, 2022.

Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, et al. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*, 2024.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024.

Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Analyzing neural network robustness using graph curvature. In *2024 International Conference on Assured Autonomy (ICAA)*, pp. 110–113. IEEE, 2024.

Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Post-training neural network pruning using graph curvature. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. Plangenllms: A modern survey of llm planning capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19497–19521, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 16(6):1–27, 2025b.

Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, et al. Wanda++: Pruning large language models via regional gradients. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4321–4333, 2025c.

Chen Zhang, Qiuchi Li, Dawei Song, Zheyu Ye, Yan Gao, and Yao Hu. Towards the law of capacity gap in distilling language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22504–22528, 2025.

Hengyuan Zhang, Shiping Yang, Xiao Liang, Chenming Shang, Yuxuan Jiang, Chaofan Tao, Jing Xiong, Hayden Kwok-Hay So, Ruobing Xie, Angel X Chang, et al. Find your optimal teacher: Personalized data synthesis via router-guided multi-teacher distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14619–14637, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n:m fine-grained structured sparse neural networks from scratch. In *International Conference on Learning Representations*, 2021.

A Appendix

A.1 Training Details

All descendant students within a comparison use the same downstream distillation data, training budget, and optimization configuration; only the pruning mask differs. Pruning masks are fixed throughout distillation, and pruned weights remain zero. For SFT, we train for **0.1B tokens** using **AdamW** with learning rate **1e-4**, weight decay **0.01**, global batch size **8 per device** $\times$ **2 gradient-accumulation steps**, and maximum sequence length **4,096** tokens. For OPD, we use the same optimization budget with objective in (Agarwal et al., 2024), and generate on-policy responses using **temperature 0.7, top-p 0.95, maximum generation length 4,096 tokens, and 1 sample per prompt**. Training is performed in **BF16** on **a single node of 4×H100 80GB GPUs (FSDP2 sharding via verl)**. We report mean and standard deviation over three independently trained runs using different training seeds. **The final checkpoint at the end of the training is used for evaluation**. All student-construction methods within the same experimental block use identical training hyperparameters and compute budgets.

A.2 Additional Results on DeepSeek-8B

Table 5: **DeepSeek-R1-Distill-Llama-8B post-distillation performance (%).**

Student Construction	Sparsity	SFT				OPD			
		MATH-500	StrategyQA	APPS	Avg.	MATH-500	StrategyQA	APPS	Avg.
DeepSeek-8B (teacher)	0%	71.8 ± 1.0	72.8 ± 1.4	50.2 ± 1.7	64.9	71.8 ± 1.0	72.8 ± 1.4	50.2 ± 1.7	64.9
Llama-3.1-Minitron-4B	0%	48.5 ± 2.1	50.8 ± 0.9	32.6 ± 1.8	44.0	52.5 ± 1.0	55.9 ± 1.2	37.1 ± 1.1	48.5
Magnitude	50%	49.3 ± 1.5	51.7 ± 1.1	33.5 ± 1.8	44.8	53.7 ± 1.4	56.7 ± 0.7	37.2 ± 1.4	49.2
SparseGPT		54.0 ± 2.2	56.5 ± 0.8	37.5 ± 1.5	49.3	57.9 ± 0.8	62.4 ± 1.2	42.3 ± 1.3	54.2
WANDA		56.0 ± 1.0	58.1 ± 0.9	39.1 ± 1.5	51.1	61.0 ± 0.7	63.6 ± 0.9	43.1 ± 1.1	55.9
WANDA++		55.3 ± 1.7	57.3 ± 1.1	38.3 ± 1.2	50.3	59.5 ± 1.0	63.4 ± 1.0	43.2 ± 1.6	55.4
DAP (Ours)		60.9 ± 2.0	60.1 ± 0.9	41.9 ± 1.4	**54.3**	66.0 ± 1.1	65.8 ± 0.7	46.8 ± 1.0	**59.5**
Magnitude	2 : 4	43.6 ± 1.8	46.8 ± 1.2	29.8 ± 2.5	40.1	47.6 ± 1.6	52.3 ± 0.9	34.0 ± 1.6	44.6
SparseGPT		50.0 ± 2.0	52.9 ± 1.1	34.8 ± 1.6	45.9	55.2 ± 1.3	58.0 ± 0.8	38.3 ± 1.4	50.5
WANDA		50.0 ± 1.2	53.2 ± 1.1	35.1 ± 1.7	46.1	54.3 ± 0.9	59.0 ± 0.7	39.2 ± 1.1	50.8
WANDA++		53.1 ± 1.6	54.8 ± 0.8	36.9 ± 1.8	48.3	58.2 ± 1.0	60.3 ± 1.1	40.5 ± 1.2	53.0
DAP (Ours)		58.0 ± 1.7	57.6 ± 1.4	39.7 ± 1.1	**51.8**	63.5 ± 1.2	63.3 ± 0.8	44.4 ± 1.3	**57.1**

A.3 Sensitivity to the Local Recovery Step Size

We study the sensitivity of DAP to the local recovery coefficient η in $S_i = \widehat{D}_i - \eta C_i$, where C_i denotes the unscaled recovery interaction. Because η arises from the local one-step surrogate rather than the downstream optimizer itself, we use the downstream learning rate only as a reference scale, defining $\eta_0 = \alpha_{\mathrm{LR}}$, and vary η relative to η_0. The main experiments use $\eta = 1.5\eta_0$. Results use Qwen3-8B at 50% unstructured sparsity with SFT, with calibration data and downstream training held fixed. The case $\eta = 0$ removes the recovery term and provides a preservation-only control.

Introducing the recovery term substantially improves post-distillation performance over the preservation-only criterion, with performance peaking near $\eta = 1.5\eta_0$. Smaller values underweight recoverability and remain closer to immediate-damage pruning, whereas excessively large values increasingly prioritize predicted recovery at the expense of immediate preservation. The resulting peak at an intermediate recovery step size is consistent with DAP's intended tradeoff between the two terms.

η/η_0	Mask overlap (%)	Post-SFT Avg.
0	65.8	46.80
0.50	78.4	51.25
1.00	91.3	53.92
1.25	96.5	55.04
1.50	100.0	55.50
1.75	95.7	55.12
2.00	88.9	54.18

Table 6: **Sensitivity to the local recovery step size.** Mask overlap is measured relative to the mask used in the main experiments ($\eta = 1.5\eta_0$).

A.4 Calibration Data and Curvature Estimation

All pruning methods are given the same total calibration budget. We construct a shared held-out calibration pool $\mathcal{D}_{\text{cal}} = \mathcal{D}_{\text{ref}} \cup \mathcal{D}_{\text{KD}}$ containing 256 sequences, each containing 4,096 tokens. DAP partitions this pool evenly into two disjoint subsets of 128 sequences each with different roles: a reference set $\mathcal{D}_{\text{ref}}$ for estimating the behavior that pruning should preserve, and a distillation-statistics set $\mathcal{D}_{\text{KD}}$ for estimating recovery statistics conditioned on the intended downstream distillation setting. These calibration samples are used only for student construction and never for parameter updates, and are disjoint from the downstream distillation training data and all evaluation sets.

Reference calibration data. The reference set $\mathcal{D}_{\text{ref}}$ is drawn from the target-task distribution. For each prompt, responses are generated by the dense parent teacher π_{θ_T}. These samples define the reference objective $L_{\text{ref},\theta_{\text{T}}}$ and are used to estimate the curvature associated with behavior preservation under pruning.

For curvature estimation, we use the sampled-target auxiliary loss

$$\ell_{\text{ref}}^{\text{curv}} = -\log \pi_\theta(y \mid \tau), \qquad \tau \sim \mathcal{D}_{\text{ref}}, \quad y \sim \pi_{\theta_T}(\cdot \mid \tau),$$

with gradients evaluated through the dense parent at $\theta = \theta_T$. For each prunable linear layer, we record the layer input activation a_{ref} and output gradient $g_{\text{ref}} = \nabla_s \ell_{\text{ref}}^{\text{curv}}$, from which we estimate

$$A_{\text{ref}} = \mathbb{E}[a_{\text{ref}} a_{\text{ref}}^\top], \qquad G_{\text{ref}} = \mathbb{E}[g_{\text{ref}} g_{\text{ref}}^\top]. \tag{12}$$

For the scalable diagonal-G variant used in the main experiments, we retain only

$$\gamma_{\text{ref}} = \text{diag}(G_{\text{ref}}) = \mathbb{E}[g_{\text{ref}} \odot g_{\text{ref}}]. \tag{13}$$

Because the sampled targets are drawn from the dense parent itself, the corresponding gradient second moment has the usual Fisher interpretation and equals the local reference curvature under the population construction described in Appendix A.5.3.

Distillation-statistics calibration data. The set $\mathcal{D}_{\text{KD}}$ is used only to estimate how pruning-induced perturbations are expected to be repaired under the intended downstream setting. Its construction follows the data-generation, autoregressive-context, filtering, and sample/token-weighting pipeline of the corresponding distillation procedure. Thus, an SFT-conditioned and an OPD-conditioned mask may use the same prompt distribution while inducing different context distributions or different weighting of those contexts.

Importantly, procedure dependence does not arise merely from replacing forward KL by reverse KL on the same fixed contexts: at distribution matching, the two have the same local Fisher under the same context distribution and normalization. Instead, DAP conditions its recovery statistics on the contexts and weighting induced by the intended downstream procedure.

For the recovery statistics, we use the same sampled-target auxiliary loss

$$\ell_{\text{KD}}^{\text{curv}} = -\log \pi_\theta(y \mid \tau), \qquad \tau \sim \mathcal{D}_{\text{KD}},$$

where the target y is sampled from the intended downstream teacher. In the default descendant setting, this teacher is the dense parent π_{θ_T}. In the cross-teacher experiment of Sec. 5.4, the targets are instead sampled from the intended downstream teacher π_{θ_Q}, while gradients are still evaluated through the fixed dense parent from which the student is pruned.

No optimization step is taken on $\mathcal{D}_{\text{KD}}$. For each prunable layer, we collect the input activation a_{KD} and output gradient $g_{\text{KD}} = \nabla_s \ell_{\text{KD}}^{\text{curv}}$, and estimate

$$A_{\text{KD}} = \mathbb{E}[a_{\text{KD}} a_{\text{KD}}^\top], \qquad G_{\text{KD}} = \mathbb{E}[g_{\text{KD}} g_{\text{KD}}^\top], \tag{14}$$

or, for the diagonal-G approximation,

$$\gamma_{\text{KD}} = \text{diag}(G_{\text{KD}}) = \mathbb{E}[g_{\text{KD}} \odot g_{\text{KD}}]. \tag{15}$$

When the intended downstream teacher equals the dense parent, G_{KD} has the usual local Fisher/Hessian interpretation on the frozen $\mathcal{D}_{\text{KD}}$ context distribution. When the intended downstream teacher differs from the parent, G_{KD} is instead the teacher-conditioned gradient second moment used by the practical DAP score; it is not interpreted as the exact Hessian of a stationary-point KD objective at θ_T. The full construction is derived in Appendix A.5.3.

Calibration budget and comparison with pruning baselines. The total calibration budget is fixed to 256 sequences of 4,096 tokens each, corresponding to 1,048,576 calibration tokens per parent model. DAP uses 128 sequences for $\mathcal{D}_{\mathrm{ref}}$ and the remaining 128 sequences for $\mathcal{D}_{\mathrm{KD}}$, corresponding to 524,288 tokens for estimating each set of statistics. Calibration-dependent pruning baselines are instead given the full shared pool $\mathcal{D}_{\mathrm{cal}}$ of 256 sequences as their calibration data. Methods that do not require calibration data, such as magnitude pruning, are applied in their standard form. Thus, DAP does not receive additional calibration samples or tokens relative to the competing pruning methods; it only partitions the same fixed calibration budget to estimate preservation and recovery statistics separately.

Practical second-moment estimation. All statistics are estimated from ordinary forward and backward passes through the dense parent model. For each prunable linear layer, we accumulate sample averages of $a_J a_J^\top$ and either $g_J g_J^\top$ or $g_J \odot g_J$, for $J \in \{\mathrm{ref}, \mathrm{KD}\}$. No explicit Hessian, Hessian-vector product, or second-order autograd graph is constructed.

Under the layerwise factorization described in Appendix A.5.3, the corresponding gradient-second-moment block is approximated by

$$F_J^{(W)} \approx G_J \otimes A_J.$$

For the reference statistics, $F_{\mathrm{ref}}^{(W)}$ is the factorized approximation to the reference Hessian block. For KD statistics in the matched-teacher setting, $F_{\mathrm{KD}}^{(W)}$ similarly approximates the local KD curvature entering the stationary-point derivation. In the cross-teacher setting, the same factorized quantity is used as a teacher-conditioned gradient-second-moment surrogate in the practical DAP score.

Unless otherwise specified, the main experiments further diagonalize the output-gradient factor,

$$F_J^{(W)} \approx \mathrm{Diag}(\gamma_J) \otimes A_J,$$

while the full-G variant

$$F_J^{(W)} \approx G_J \otimes A_J$$

is used only in the approximation ablation.

A.5 Additional Derivations

This section provides additional derivations for the local post-distillation criterion and the scalable pruning scores used by DAP. We first derive the first-order recovery approximation in Eq. 9, then specialize it to a single-weight removal, and finally derive the diagonal-G and full-G factorized criteria.

A.5.1 Derivation of the Local Post-Distillation Criterion

Throughout this local derivation, we consider the matched-teacher setting in which the dense parent is also the intended distillation teacher. Accordingly, H_{ref} and H_{KD} are evaluated at θ_T, and we assume $\nabla_\theta L_{\mathrm{KD}}(\theta_T) \approx 0$, as in the main text. If the downstream teacher differs from the parent, this gradient need not vanish; the exact Taylor expansion would contain an additional first-order drift term. Our cross-teacher experiments therefore use the teacher-conditioned second-moment extension described in Appendix A.5.3, rather than interpreting Eq. 9 as an exact cross-teacher expansion.

Recall that a pruning decision g induces the perturbation

$$\delta_g = (z_g - \mathbf{1}) \odot \theta_T,$$

and that the local distillation update is approximated by

$$m_g \approx -\eta\, z_g \odot (H_{\mathrm{KD}} \delta_g).$$

For convenience, define

$$q_g := z_g \odot (H_{\mathrm{KD}} \delta_g),$$

so that $m_g \approx -\eta q_g$.

Under the quadratic approximation of the reference objective, the remaining damage after this update is

$$\begin{aligned} I_g &\approx \frac{1}{2}(\delta_g + m_g)^\top H_{\mathrm{ref}}(\delta_g + m_g) \\ &= \frac{1}{2}\delta_g^\top H_{\mathrm{ref}}\delta_g + \delta_g^\top H_{\mathrm{ref}} m_g + \frac{1}{2} m_g^\top H_{\mathrm{ref}} m_g, \end{aligned} \tag{16}$$

where we use the symmetry of H_{ref} to combine the two cross terms. Substituting $m_g \approx -\eta q_g$ gives

$$\begin{aligned} I_g &\approx \frac{1}{2}\delta_g^\top H_{\mathrm{ref}}\delta_g - \eta\, \delta_g^\top H_{\mathrm{ref}} q_g + \frac{\eta^2}{2} q_g^\top H_{\mathrm{ref}} q_g \\ &= \frac{1}{2}\delta_g^\top H_{\mathrm{ref}}\delta_g - \eta\, \delta_g^\top H_{\mathrm{ref}}\big[z_g \odot (H_{\mathrm{KD}}\delta_g)\big] \\ &\quad + \frac{\eta^2}{2} q_g^\top H_{\mathrm{ref}} q_g. \end{aligned} \tag{17}$$

The final term is second order in the local recovery step size η. Retaining terms only through first order in η therefore yields

$$\widehat{I}_g = \underbrace{\frac{1}{2}\delta_g^\top H_{\mathrm{ref}}\delta_g}_{\widehat{D}_g} \underbrace{- \eta\, \delta_g^\top H_{\mathrm{ref}}\big[z_g \odot (H_{\mathrm{KD}}\delta_g)\big]}_{\widehat{R}_g}, \tag{18}$$

which is Eq. 9. Here, $\widehat{D}_g$ estimates immediate damage and $\widehat{R}_g$ estimates the signed first-order reduction in reference damage from updates to surviving parameters. Negative $\widehat{R}_g$ predicts increased damage; $\widehat{I}_g$ need not be nonnegative because the quadratic term in η is omitted.

A.5.2 Derivation of the Single-Weight Criterion

The specialization below inherits the matched-teacher assumptions of Appendix A.5.1.

Consider removing a single weight w_i. Let e_i denote the i-th standard basis vector. The corresponding mask satisfies $(z_i)_i = 0$ and $(z_i)_j = 1$ for $j \neq i$, so the pruning perturbation is

$$\delta_i = -w_i e_i. \tag{19}$$

The immediate-damage term in Eq. 9 becomes

$$\begin{aligned} \widehat{D}_i &= \frac{1}{2}\delta_i^\top H_{\mathrm{ref}}\delta_i \\ &= \frac{1}{2} w_i^2 e_i^\top H_{\mathrm{ref}} e_i \\ &= \frac{1}{2} w_i^2 [H_{\mathrm{ref}}]_{ii}. \end{aligned} \tag{20}$$

For the recovery term,

$$\begin{aligned} \widehat{R}_i &= \eta\, \delta_i^\top H_{\mathrm{ref}}\big[z_i \odot (H_{\mathrm{KD}}\delta_i)\big] \\ &= \eta w_i^2 e_i^\top H_{\mathrm{ref}}\big[z_i \odot (H_{\mathrm{KD}} e_i)\big]. \end{aligned} \tag{21}$$

Expanding over parameter coordinates gives

$$\begin{aligned} \widehat{R}_i &= \eta w_i^2 \sum_j [H_{\mathrm{ref}}]_{ij}(z_i)_j [H_{\mathrm{KD}}]_{ji} \\ &= \eta w_i^2 \sum_{j \neq i} [H_{\mathrm{ref}}]_{ij} [H_{\mathrm{KD}}]_{ji}, \end{aligned} \tag{22}$$

where the $j = i$ term vanishes because $(z_i)_i = 0$. Thus, recovery is modeled only through parameters that remain after pruning.

Combining Eqs. 20 and 22 yields the single-weight specialization of Eq. 9:

$$\widehat{I}_i = \frac{1}{2} w_i^2 [H_{\mathrm{ref}}]_{ii} - \eta w_i^2 \sum_{j \neq i} [H_{\mathrm{ref}}]_{ij} [H_{\mathrm{KD}}]_{ji}. \tag{23}$$

The first term estimates the immediate damage caused by removing w_i, while the second estimates the signed first-order recovery contribution from updates to the surviving parameters.

A.5.3 Derivation of the Factorized DAP Score

We discard cross-layer interactions and consider a linear layer with $W \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$. Let

$$q_{\mathrm{ref}}(\cdot \mid \tau) = \pi_{\theta_T}(\cdot \mid \tau), \qquad q_{\mathrm{KD}}(\cdot \mid \tau) = \pi_{\theta_Q}(\cdot \mid \tau),$$

where $\theta_Q = \theta_T$ in the default matched-teacher setting and θ_Q denotes the intended downstream teacher in the cross-teacher setting. For $J \in \{\mathrm{ref}, \mathrm{KD}\}$, define

$$\ell_J^{\mathrm{curv}}(\theta; \tau, y) = -\log \pi_\theta(y \mid \tau), \qquad \tau \sim \mathcal{D}_J, \quad y \sim q_J(\cdot \mid \tau),$$

and

$$u_J = \nabla_\theta \ell_J^{\mathrm{curv}}|_{\theta=\theta_T}, \qquad F_J = \mathbb{E}[u_J u_J^\top].$$

The calibration distributions, target distributions, and sampled targets are held fixed during differentiation.

For the reference objective, $q_{\mathrm{ref}} = \pi_{\theta_T}$, so the score identity gives

$$\mathbb{E}[u_{\mathrm{ref}}] = 0, \qquad F_{\mathrm{ref}} = \nabla_\theta^2 L_{\mathrm{ref},\theta_T}(\theta_T) = H_{\mathrm{ref}}.$$

When the intended teacher also equals the parent, the same identity applies to F_{KD} on the frozen $\mathcal{D}_{\mathrm{KD}}$ context distribution. In particular, forward and reverse KL have the same local Fisher at distribution matching when evaluated on the same fixed contexts with the same normalization. Therefore, procedure conditioning in our estimator comes from the procedure-dependent distribution $\mathcal{D}_{\mathrm{KD}}$ and its sample/token weighting, rather than from KL direction alone.

When $\theta_Q \neq \theta_T$, $\mathbb{E}[u_{\mathrm{KD}}]$ generally need not vanish and F_{KD} is not identified with $\nabla_\theta^2 L_{\mathrm{KD}}(\theta_T)$. In this case, F_{KD} is the teacher-conditioned gradient-second-moment surrogate used by the practical DAP score.

At output logits s, let $p_T = \pi_{\theta_T}(\cdot \mid \tau)$ and let q_J denote the corresponding next-token target distribution. Then

$$\nabla_s \ell_J^{\mathrm{curv}} = p_T - e_y,$$

and

$$\mathbb{E}_{y \sim q_J}\left[(p_T - e_y)(p_T - e_y)^\top\right] = \mathrm{Diag}(q_J) - p_T q_J^\top - q_J p_T^\top + p_T p_T^\top.$$

When $q_J = p_T$, this reduces to

$$\mathrm{Diag}(p_T) - p_T p_T^\top,$$

the usual Fisher at distribution matching. Thus, changing the intended teacher can change the KD gradient second moment even when the dense parent through which gradients are evaluated is fixed.

Shared weights and normalization. For each $J \in \{\mathrm{ref}, \mathrm{KD}\}$, consider a calibration sample (τ, y) with auxiliary loss $\ell_J^{\mathrm{curv}} = -\log \pi_\theta(y \mid \tau)$. This is the unreduced sample negative log-likelihood: when y is a sequence, its log-likelihood sums over predicted tokens, while the objective averages over samples.

Let $t = 1, \ldots, n_J$ index the valid invocations of a shared linear layer $s_{J,t} = W a_{J,t}$ within this sample, and define

$$g_{J,t} = \nabla_{s_{J,t}} \ell_J^{\mathrm{curv}}\big|_{\theta=\theta_T}.$$

The index t enumerates layer invocations, not individual loss terms. All quantities are evaluated at the dense parent, with contexts and sampled targets held fixed. The complete weight gradient is

$$U_J := \nabla_W \ell_J^{\mathrm{curv}}|_{\theta_T} = \sum_{t=1}^{n_J} g_{J,t} a_{J,t}^\top.$$

Consequently, under row-wise vectorization, the exact layer gradient-second-moment block associated with F_J is

$$\begin{aligned} F_J^{(W)} &= \mathbb{E}[\mathrm{vec}_r(U_J)\,\mathrm{vec}_r(U_J)^\top] \\ &= \mathbb{E}\left[\sum_{t,t'=1}^{n_J} (g_{J,t}g_{J,t'}^\top) \otimes (a_{J,t}a_{J,t'}^\top)\right]. \end{aligned} \tag{24}$$

In the matched-teacher setting this block has the usual Fisher interpretation; in the cross-teacher setting it is a raw teacher-conditioned gradient second moment.

We first discard cross-invocation terms $t \neq t'$. We then separate activation and gradient second moments under the distribution obtained by pooling valid invocations across calibration samples. Writing $\bar{n}_J = \mathbb{E}[n_J]$, define

$$\begin{aligned} A_J &= \frac{1}{\bar{n}_J}\mathbb{E}\left[\sum_{t=1}^{n_J} a_{J,t}a_{J,t}^\top\right], \\ G_J &= \mathbb{E}\left[\sum_{t=1}^{n_J} g_{J,t}g_{J,t}^\top\right]. \end{aligned} \tag{25}$$

These two approximations give

$$F_J^{(W)} \approx \mathbb{E}\left[\sum_t (g_{J,t}g_{J,t}^\top) \otimes (a_{J,t}a_{J,t}^\top)\right] \approx G_J \otimes A_J. \tag{26}$$

Discarding cross-invocation terms is an additional approximation; these terms need not vanish in a transformer.

For N_J calibration samples containing $M_J = \sum_{k=1}^{N_J} n_{J,k}$ valid invocations, the corresponding estimators are

$$\widehat{A}_J = \frac{1}{M_J}\sum_{k,t} a_{J,k,t}a_{J,k,t}^\top, \qquad \widehat{G}_J = \frac{1}{N_J}\sum_{k,t} g_{J,k,t}g_{J,k,t}^\top.$$

Thus, activation moments are averaged over invocations, whereas gradient moments are summed within each sample and then averaged over samples. Any batch or token reduction applied during backpropagation is undone before forming these per-sample gradient moments. The invocation-count factor is therefore included in G_J, rather than omitted from the second moment.

For the diagonal-G variant used by DAP, we further set

$$\gamma_J = \mathrm{diag}(G_J)$$

and approximate

$$[F_J^{(W)}]_{(a,b),(a',c)} \approx 1[a = a']\,\gamma_{J,a}[A_J]_{bc}. \tag{27}$$

Thus, after the diagonal-G approximation, a removed weight W_{ab} interacts only with other weights W_{ac} in the same output row.

For the reference term, $F_{\mathrm{ref}} = H_{\mathrm{ref}}$, so

$$[H_{\mathrm{ref}}]_{(a,b),(a,b)} = [F_{\mathrm{ref}}^{(W)}]_{(a,b),(a,b)} \approx \gamma_{\mathrm{ref},a}[A_{\mathrm{ref}}]_{bb}. \tag{28}$$

The estimated immediate damage is therefore

$$\widehat{D}_{ab} = \frac{1}{2}W_{ab}^2\gamma_{\mathrm{ref},a}[A_{\mathrm{ref}}]_{bb}. \tag{29}$$

In the matched-teacher setting, F_{KD} supplies the factorized approximation to H_{KD} in Eq. 22. In the cross-teacher setting, we use the same algebraic score with the teacher-conditioned F_{KD} in this recovery slot; this defines the practical second-moment extension rather than an exact cross-teacher Taylor expansion. Hence

$$\begin{aligned} \widehat{R}_{ab} &\approx \eta W_{ab}^2 \sum_{c\neq b} [F_{\mathrm{ref}}^{(W)}]_{(a,b),(a,c)}[F_{\mathrm{KD}}^{(W)}]_{(a,c),(a,b)} \\ &\approx \eta W_{ab}^2 \gamma_{\mathrm{ref},a}\gamma_{\mathrm{KD},a} \sum_{c\neq b} [A_{\mathrm{ref}}]_{bc}[A_{\mathrm{KD}}]_{cb}. \end{aligned} \tag{30}$$

Combining Eqs. 29 and 30 gives the practical DAP score

$$S_{ab} = \frac{1}{2} W_{ab}^2 \gamma_{\mathrm{ref},a} [A_{\mathrm{ref}}]_{bb} - \eta W_{ab}^2 \gamma_{\mathrm{ref},a} \gamma_{\mathrm{KD},a} \sum_{c \neq b} [A_{\mathrm{ref}}]_{bc} [A_{\mathrm{KD}}]_{cb}, \tag{31}$$

which is Eq. 11. The first term estimates the immediate damage from removing W_{ab}, while the second captures the signed recovery contribution assigned by the practical teacher- and procedure-conditioned second-moment score.

A.5.4 Full-G Factorized Criterion

For the higher-fidelity factorized variant, we retain the full output-gradient second moment G_J, so that

$$F_J^{(W)} \approx G_J \otimes A_J. \tag{32}$$

Under row-wise vectorization, the corresponding interaction between weights W_{ab} and W_{cd} is

$$[F_J^{(W)}]_{(a,b),(c,d)} \approx [G_J]_{ac} [A_J]_{bd}. \tag{33}$$

Since $F_{\mathrm{ref}} = H_{\mathrm{ref}}$, the immediate-damage term for a removed weight W_{ab} is

$$\widehat{D}_{ab}^{\mathrm{fullG}} = \frac{1}{2} W_{ab}^2 [G_{\mathrm{ref}}]_{aa} [A_{\mathrm{ref}}]_{bb}. \tag{34}$$

As above, in the matched-teacher setting the KD second moment approximates the KD Hessian entering the local criterion, whereas in the cross-teacher setting the same expression defines the practical teacher-conditioned extension. The recovery interaction is therefore

$$\begin{aligned} \widehat{R}_{ab}^{\mathrm{fullG}} &\approx \eta W_{ab}^2 \sum_{(c,d) \neq (a,b)} [F_{\mathrm{ref}}^{(W)}]_{(a,b),(c,d)} [F_{\mathrm{KD}}^{(W)}]_{(c,d),(a,b)} \\ &\approx \eta W_{ab}^2 \sum_{(c,d) \neq (a,b)} [G_{\mathrm{ref}}]_{ac} [G_{\mathrm{KD}}]_{ca} [A_{\mathrm{ref}}]_{bd} [A_{\mathrm{KD}}]_{db}. \end{aligned} \tag{35}$$

The unrestricted double sum factorizes as

$$\begin{aligned} &\sum_{c,d} [G_{\mathrm{ref}}]_{ac} [G_{\mathrm{KD}}]_{ca} [A_{\mathrm{ref}}]_{bd} [A_{\mathrm{KD}}]_{db} \\ &\qquad = [G_{\mathrm{ref}} G_{\mathrm{KD}}]_{aa} [A_{\mathrm{ref}} A_{\mathrm{KD}}]_{bb}. \end{aligned} \tag{36}$$

Removing the $(c,d) = (a,b)$ contribution therefore gives

$$\begin{aligned} \widehat{R}_{ab}^{\mathrm{fullG}} = \eta W_{ab}^2 \Big(&[G_{\mathrm{ref}} G_{\mathrm{KD}}]_{aa} [A_{\mathrm{ref}} A_{\mathrm{KD}}]_{bb} \\ &- [G_{\mathrm{ref}}]_{aa} [G_{\mathrm{KD}}]_{aa} [A_{\mathrm{ref}}]_{bb} [A_{\mathrm{KD}}]_{bb} \Big). \end{aligned} \tag{37}$$

The resulting full-G factorized criterion is

$$\begin{aligned} S_{ab}^{\mathrm{fullG}} = &\frac{1}{2} W_{ab}^2 [G_{\mathrm{ref}}]_{aa} [A_{\mathrm{ref}}]_{bb} \\ &- \eta W_{ab}^2 \Big([G_{\mathrm{ref}} G_{\mathrm{KD}}]_{aa} [A_{\mathrm{ref}} A_{\mathrm{KD}}]_{bb} \\ &\qquad - [G_{\mathrm{ref}}]_{aa} [G_{\mathrm{KD}}]_{aa} [A_{\mathrm{ref}}]_{bb} [A_{\mathrm{KD}}]_{bb} \Big). \end{aligned} \tag{38}$$

Unlike the diagonal-G criterion, which restricts modeled recovery to weights in the same output row, the full-G variant retains cross-output-row interactions within each layer. The diagonal-G criterion used by DAP trades these additional interactions for a substantially more scalable second-moment representation.