Beyond Loss Values: Robust Dynamic Pruning via Loss Trajectory Alignment

Supplementary Material

Contents

A. Additional Implementation Details

We provide further implementation details to support reproducibility of our experimental setup below.

A.1. Details of Dataset

Specifically, **CIFAR-100N** and **CIFAR-10N** [54] consist of 50K human re-annotated training images of size 32×32, with 100 and 10 classes respectively, adapted from the original CIFAR-100 and CIFAR-10 datasets [24]. Both datasets include human-annotated noisy labels. We use the Real noisy-label set from CIFAR-100N, and both the Aggregate and Worst noisy-label sets from CIFAR-10N to represent *real* label noise. Following prior work [34], we inject *synthetic* symmetric and asymmetric label noise with rates {0.2, 0.5, 0.8} and {0.2, 0.4} respectively, to simulate controllable noise. **WebVision** [27] contains 2.4M images crawled from the Web, covering the same 1,000 categories as ImageNet-1K [8]. Following standard practice in [5], we use a subset of WebVision containing approximately 66K training images from the first 50 classes of the Google image. **Clothing-1M** [57] consists of 1M training images sourced from online shopping platforms, with naturally occurring noisy labels. We follow standard practice and perform fine-tuning on the entire Clothing-1M training set using a pre-trained model weights on ImageNet. **ImageNet-1K** [8] consists of 1.2M training images with 1000 classes. We inject *synthetic* asymmetric label noise with rate 0.2 following settings in [39].

A.2. Details of Experimental Setup

Table A summarizes the experimental setups and hyper-parameters used across four main datasets and methods. For CIFAR-100N and CIFAR-10N, we follow the settings from [42], using a ResNet-18 [15] backbone trained for 200 epochs optimized by LARS [59] with momentum of 0.9, weight decay of 5e-4, and batch size of 128. The initial learning rates are set to 5.2 (CIFAR-100N) and 5.62 (CIFAR-10N), both with a OneCycle learning rate scheduler (cosine annealing). For WebVision, we adopt the standard setup from [5, 39], using an InceptionResNetV2 [49] trained for 100 epochs optimized by SGD with batch size of 32. For Clothing-1M, we fine-tune a ResNet-50 [15] pre-trained on ImageNet-1K [8] for 10 epochs using a batch size of 32. The initial learning rates of WebVision and Clothing-1M are set to 0.02 and 0.002, decayed by a factor of 10 halfway through the total training epochs. For ImageNet-1K, we use the default settings for corresponding architectures with 100 epoch training. Unless otherwise stated, we apply standard data augmentations, including normalization, random cropping, and horizontal flipping for all images during model training.

To ensure fairness and reproducibility, we adopt the default optimal hyper-parameter settings from the original papers for both static and dynamic data pruning baselines. For InfoBatch [42], we use the pruning probability $r = 0.5$ for CIFAR-100N and CIFAR-10N, and a more aggressive $r = 0.75$ for the large-scale datasets WebVision and Clothing-1M. For SeTa [67], we use $r = 0.1$ with window scale $\alpha = 0.9$ and group number $k = 5$ on CIFAR-100N and CIFAR-10N. Due to training collapse of SeTa on WebVision and Clothing-1M, we only report results with InfoBatch on these two datasets. The annealing ratio δ is set to 0.875 for both InfoBatch and SeTa across all settings. For our ***AlignPrune***, we follow the same pruning hyper-parameters as the corresponding dynamic pruning method. The trajectory window size N is set to 25 by default, and reduced to 5 for Clothing-1M due to its short fine-tuning schedule. Pearson's correlation is used as the default correlation function ρ.

A.3. Details of Noise Injection

For *synthetic* label noise on CIFAR-100N and CIFAR-10N, we follow the protocol in [34]. Specifically, we inject: **Symmetric noise** with rates {0.2, 0.5, 0.8}, where $r\%$ of the labels from each class c are randomly flipped to the next consecutive class $c + 1$, and labels from the final class are wrapped around to class 0. **Asymmetric noise** with rates {0.2, 0.4}, where selected samples are flipped within their super-class as defined in [54]. For ImageNet-1K experiments, we follow [39] to inject asymmetric noise with rate of 0.2. This controlled noise injection allows us to benchmark the robustness of different baselines across varying noisy types and noisy rates.

A.4. Details of Baseline Method

We clarify the selection criteria for dynamic pruning baselines as follows: Several recent methods are excluded from our evaluation due to incompatibility with our experimental setting. Specifically, SCAN [12] and DISSect [64] are tailored for contrastive pre-training frameworks like CLIP [43], which significantly differ from our focus on noisy learning. Additionally, methods by [44], [58], [55] and [14] are excluded due to the lack of publicly available implementations, and their reported results are limited to clean-label scenarios with different hyper-parameters, making fair and reproducible comparison infeasible.

Remark. Although AlignPrune is designed as a plug-and-play replacement to enhance dynamic pruning methods, its integration assumes an epoch-level sample-wise scoring and ranking mechanism based on loss trajectories. We selected InfoBatch [42] and SeTa [67] as representative dynamic pruning baselines because both provide public implementations that align well with our framework. In contrast, DivBS [17] and IES [61] adopt fundamentally different designs: DivBS performs pruning at batch level, making per-sample loss trajectory inapplicable. Moreover, applying such batch-level pruning under label noise tends to remove meaningful samples while retaining noisy ones. IES, while structurally com-

Table A. **Summary of the experimental setups and hyper-parameters on CIFAR-100N, CIFAR-10N, WebVision and Clothing-1M datasets.**

Hyper-parameters		CIFAR-100N	CIFAR-10N	WebVision	Clothing-1M
Training Setup	architecture	ResNet-18	ResNet-18	InceptionResNetV2	ResNet-50 (Pre-trained)
	training epoch	200	200	100	10
	batch size	128	128	32	32
	optimizer	LARS	LARS	SGD	SGD
	learning rate	5.2	5.62	0.02	0.002
	lr scheduler	OneCycle	OneCycle	MultiStep-50th	MultiStep-5th
	weight decay	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
InfoBatch	r	0.5	0.5	0.75	0.75
	δ	0.875	0.875	0.875	0.875
SeTa	r	0.1	0.1	–	–
	α	0.9	0.9		
	k	5	5		
	δ	0.875	0.875		
Ours	N	25	25	25	5
	ρ	Pearson	Pearson	Pearson	Pearson

patible, becomes functionally equivalent to InfoBatch once AlignPrune is applied, with the only difference being the use of a fixed threshold versus a dynamic mean-based threshold. Therefore, we integrate AlignPrune only with InfoBatch and SeTa, while including DivBS and IES in the comparison for completeness.

B. Additional Experimental Results

B.1. Extended Baseline Comparisons

Main Results under Label Noise. Tables B and C present a comprehensive comparison of AlignPrune with both static and dynamic data pruning baselines on CIFAR-100N and CIFAR-10N under three pruning ratios. Overall, static pruning methods perform significantly worse than dynamic methods, particularly in the presence of label noise. This highlights the limitations of static coreset selection when noisy labels are involved. In contrast, the proposed AlignPrune consistently achieves the best performance across all pruning ratios and noise types, demonstrating strong robustness and generalizability under diverse noisy-label conditions.

Results with Re-labeling under Label Noise. We further explore the integration of AlignPrune with robust learning techniques designed to mitigate label noise. As highlighted in [39], re-labeling strategies can improve model performance by explicitly correcting noisy annotations. We adopt SOP+ [29], a recent method designed to model label noise by disentangling noisy and clean labels through a trainable over-parameterized consistency module and self-regularization, into our framework. Specifically, we use the original hyper-parameter settings from SOP+, and implement it on top of InfoBatch [42] and Prune4ReL [39] to represent state-of-the-art dynamic and static pruning approaches, respectively. As shown in Tabs. D and E, SOP+ improves all methods, and when combined with AlignPrune, it achieves the state-of-the-art performance across benchmarks. These results confirm that AlignPrune remains compatible with existing re-labeling techniques and can further benefit from them under noisy scenarios.

B.2. Additional Analysis and Discussions

B.2.1. Analysis on Efficiency Comparison

As reported in Tab. 4, AlignPrune with InfoBatch can achieve better classification accuracy while reducing the total training time compared to vanilla InfoBatch. Although AlignPrune introduces a minor computational overhead due to the computation of the Dynamic Alignment Score (DAS) at each epoch, the benefit arises from its improved sample selection efficiency. Unlike InfoBatch, which ranks samples solely based on per-epoch loss, AlignPrune leverages DAS for more robust ranking, enabling more effective identification and pruning of low-utility and noisy samples.

Specifically, as shown in Tab. F, replacing the original loss-based metric with DAS, and combining with the soft-limit nature of the pruning probability, leads AlignPrune to achieve a higher average pruning ratio per epoch. We argue this is not simply a side effect of more aggressive pruning, but rather a direct consequence of the superior sample selection quality of DAS. Because DAS provides a clearer and more confident signal for identifying noisy and uninformative samples compared to raw loss, it can decisively discard a larger fraction of the dataset at each epoch without harming performance. Thus, this improved selection efficiency is what drives both the accuracy gains and the reduction in total training time.

Table B. **Extended classification results on CIFAR-100N with ResNet-18.** Performance gaps relative to the full-training setting are indicated as superscripts. Static pruning baselines are highlighted with gray. The mean Δ is computed across all noise types. † indicates Prune4ReL with standard CE loss to ensure fair comparison on pure pruning methods.

Noisy Type →	Clean	Real	Symmetric			Asymmetric		Mean
Pruning Method ↓			0.2	0.5	0.8	0.2	0.4	Δ
Full-training	78.2	56.1	71.4	58.6	39.8	72.4	63.3	–
Prune Ratio ∼30%								
Static Random	73.8 −4.4	53.3 −2.8	67.9 −3.4	51.1 −7.5	30.3 −9.5	68.4 −4.0	57.8 −5.5	-5.3
SmallL [19]	71.3 −6.9	58.0 +1.9	69.2 −2.2	51.9 −6.7	13.7 −26.1	68.2 −4.2	57.0 −6.2	-7.2
Margin [7]	75.4 −2.8	48.0 −8.1	59.2 −12.2	15.9 −42.7	6.3 −33.5	57.3 −15.1	42.1 −21.2	-19.4
Forget [52]	74.8 −3.4	57.7 +1.6	70.2 −1.1	51.3 −7.3	12.5 −27.4	68.4 −4.0	59.2 −4.1	-6.5
GraNd [40]	73.0 −5.2	40.2 −15.9	49.4 −21.9	9.3 −49.3	6.2 −33.6	51.8 −20.5	30.3 −33.0	-25.6
Glister [21]	73.9 −4.3	51.5 −4.6	59.9 −11.5	40.4 −18.2	16.4 −23.4	62.9 −9.5	49.7 −13.6	-12.2
Moderate [56]	73.2 −5.0	52.1 −4.0	60.2 −11.2	30.2 −28.4	9.9 −29.9	60.0 −12.4	45.7 −17.6	-15.5
SSP [48]	74.9 −3.3	53.0 −3.1	60.3 −11.1	35.8 −22.9	7.9 −31.9	62.9 −9.5	50.4 −12.8	-13.5
Prune4ReL† [39]	73.2 −5.0	52.3 −3.8	61.2 −10.1	37.0 −21.6	9.0 −30.8	61.1 −11.3	48.9 −14.4	-13.8
Dyn-Unc [16]	77.2 −1.0	58.5 +2.3	70.9 −0.5	55.4 −3.2	24.3 −15.5	71.8 −0.6	62.6 −0.7	-2.7
DUAL [6]	78.1 −0.1	58.8 +2.7	71.3 −0.1	55.4 −3.2	22.7 −17.1	72.2 −0.2	60.8 −2.4	-2.9
Dynamic Random [36]	77.3 −0.9	54.7 −1.4	69.9 −1.4	58.8 +0.2	40.1 +0.3	71.5 −0.9	63.2 −0.1	-0.6
DivBS [17]	77.1 −1.1	54.6 −1.5	63.5 −7.9	51.0 −7.6	20.8 −19.0	65.9 −6.5	56.2 −7.1	-7.2
IES [61]	76.1 −2.1	53.9 −2.2	64.2 −7.2	51.0 −7.6	37.0 −2.8	65.3 −7.0	53.8 −9.5	-5.5
InfoBatch [42]	79.0 +0.8	56.1 +0.0	71.4 +0.0	59.7 +1.1	41.8 +2.0	71.9 −0.5	64.2 +0.9	+0.6
InfoBatch + Ours	79.3 +1.1	59.4 +3.3	71.8 +0.4	66.0 +7.4	41.8 +2.0	72.6 +0.2	68.0 +4.7	**+2.7**
SeTa [67]	79.0 +0.8	55.6 −0.5	70.2 −1.2	59.0 +0.4	41.6 +1.8	71.4 −0.9	63.2 −0.1	+0.0
SeTa + Ours	79.3 +1.1	56.3 +0.2	70.8 −0.5	60.5 +1.9	41.6 +1.8	71.9 −0.5	64.3 +1.0	**+0.7**
Prune Ratio ∼50%								
Static Random	72.1 −6.1	51.8 −4.3	64.0 −7.4	47.2 −11.5	22.9 −16.9	65.3 −7.0	54.6 −8.7	-8.8
SmallL [19]	65.8 −12.4	56.1 −0.1	64.9 −6.4	57.9 −0.8	18.0 −21.8	64.2 −8.1	58.8 −4.4	-7.7
Margin [7]	71.2 −7.0	36.3 −19.8	44.7 −26.7	11.0 −47.6	5.9 −33.9	43.7 −28.7	26.8 −36.5	-28.6
Forget [52]	69.8 −8.4	56.4 +0.3	68.4 −2.9	60.4 +1.8	16.5 −23.3	66.1 −6.3	60.7 −2.5	-5.9
GraNd [40]	64.0 −14.2	25.8 −30.3	28.6 −42.8	4.1 −54.5	4.3 −35.5	36.6 −35.8	18.2 −45.1	-36.9
Glister [21]	69.5 −8.8	49.2 −6.9	55.0 −16.3	34.3 −24.4	12.9 −26.9	58.2 −14.2	45.0 −18.3	-16.5
Moderate [56]	68.7 −9.5	49.0 −7.2	52.7 −18.6	22.3 −36.3	6.4 −33.4	55.1 −17.3	41.0 −22.3	-20.6
SSP [48]	70.6 −7.6	49.2 −6.9	52.9 −18.5	31.1 −27.5	7.5 −32.3	58.6 −13.8	47.5 −15.8	-17.5
Prune4ReL† [39]	69.1 −9.1	51.7 −4.4	58.2 −13.2	32.7 −25.9	7.4 −32.4	58.7 −13.7	45.3 −17.9	-16.7
Dyn-Unc [16]	74.1 −4.2	59.7 +3.5	71.4 +0.0	61.7 +3.1	22.8 −17.0	71.1 −1.2	66.5 +3.2	-1.8
DUAL [6]	74.6 −3.6	59.9 +3.8	71.3 −0.1	61.9 +3.3	20.6 −19.2	71.1 −1.3	62.9 −0.4	-2.5
Dynamic Random [36]	75.3 −2.9	54.1 −2.0	70.4 −1.0	59.5 +0.9	40.7 +0.9	70.1 −2.3	64.8 +1.6	-0.7
DivBS [17]	77.1 −1.1	53.8 −2.3	63.8 −7.5	46.6 −12.0	18.0 −21.9	64.5 −7.8	56.7 −6.6	-8.5
IES [61]	75.5 −2.7	53.2 −2.9	62.7 −8.7	50.2 −8.4	34.4 −5.4	63.3 −9.1	54.8 −8.5	-6.5
InfoBatch [42]	77.7 −0.5	56.0 −0.1	71.3 −0.1	60.5 +1.9	42.2 +2.4	71.8 −0.6	65.2 +2.0	+0.7
InfoBatch + Ours	78.5 +0.3	60.7 +4.6	71.6 +0.3	62.0 +3.4	42.6 +2.8	72.6 +0.3	68.6 +5.3	**+2.4**
SeTa [67]	77.5 −0.7	55.7 −0.4	70.7 −0.7	60.0 +1.4	40.5 +0.7	71.9 −0.5	64.5 +1.2	+0.2
SeTa + Ours	78.4 +0.2	56.3 +0.1	71.2 −0.2	61.0 +2.4	41.9 +2.1	72.2 −0.2	66.0 +2.7	**+1.0**
Prune Ratio ∼70%								
Static Random	69.7 −8.5	45.9 −10.2	56.7 −14.6	39.0 −19.6	15.8 −24.0	58.4 −13.9	49.5 −13.8	-14.9
SmallL [19]	55.7 −22.5	49.8 −6.3	57.0 −14.4	54.6 −4.1	22.0 −17.8	56.7 −15.7	53.8 −9.5	-12.9
Margin [7]	56.7 −21.5	10.6 −45.5	12.3 −59.1	6.5 −52.2	4.5 −35.3	17.1 −55.3	5.2 −58.1	-46.7
Forget [52]	58.6 −19.6	50.0 −6.1	61.8 −9.6	59.3 +0.7	22.8 −17.0	58.4 −14.0	56.1 −7.2	-10.4
GraNd [40]	46.2 −32.0	14.7 −41.4	10.1 −61.3	2.9 −55.8	3.4 −36.4	20.1 −52.3	9.1 −54.2	-47.6
Glister [21]	61.1 −17.1	43.6 −12.5	46.1 −25.3	22.4 −36.2	9.1 −30.7	49.9 −22.5	38.5 −24.8	-24.1
Moderate [56]	60.6 −17.6	43.3 −12.8	46.2 −25.2	17.7 −41.0	5.1 −34.7	46.6 −25.8	36.5 −26.7	-26.3
SSP [48]	60.3 −17.9	42.2 −14.0	41.3 −30.1	21.6 −37.0	6.1 −33.8	49.2 −23.2	37.7 −25.6	-25.9
Prune4ReL† [39]	61.4 −16.8	46.2 −10.0	51.1 −20.3	28.0 −30.7	6.1 −33.7	52.4 −20.0	42.8 −20.5	-21.7
Dyn-Unc [16]	63.9 −14.3	57.5 +1.4	67.1 −4.3	63.4 +4.7	24.6 −15.2	65.8 −6.6	63.0 −0.2	-4.9
DUAL [6]	68.9 −9.3	57.9 +1.7	66.4 −5.0	62.5 +3.9	19.9 −19.9	67.3 −5.1	61.5 −1.8	-5.1
Dynamic Random [36]	75.2 −3.0	53.3 −2.9	70.2 −1.2	62.7 +4.1	41.0 +1.2	71.9 −0.5	67.3 +4.0	+0.3
DivBS [17]	76.9 −1.3	53.6 −2.6	61.5 −9.8	45.8 −12.5	19.4 −20.4	63.5 −8.8	58.5 −4.8	-8.6
IES [61]	74.8 −3.4	52.6 −3.6	61.0 −10.4	50.3 −8.3	28.6 −11.2	62.6 −9.8	51.9 −11.4	-8.3
InfoBatch [42]	77.1 −1.1	55.0 −1.2	71.4 +0.1	63.6 +5.0	42.4 +2.6	72.2 −0.2	67.8 +4.5	+1.4
InfoBatch + Ours	77.5 −0.7	58.3 +2.2	72.2 +0.9	64.7 +6.1	42.8 +3.0	72.5 +0.1	69.1 +5.9	**+2.5**
SeTa [67]	77.2 −1.1	55.2 −1.0	71.6 +0.2	62.4 +3.8	42.4 +2.6	72.0 −0.4	67.4 +4.1	+1.2
SeTa + Ours	77.8 −0.4	55.5 −0.6	72.3 +0.9	63.3 +4.7	42.7 +2.9	72.7 +0.3	67.6 +4.3	**+1.7**

Table C. **Extended classification results on CIFAR-10N with ResNet-18.** Performance gaps relative to the full-training setting are indicated as superscripts. Static pruning methods are highlighted with gray. The mean Δ is computed across all noise types. † indicates Prune4ReL with standard CE loss to ensure fair comparison on pure pruning methods.

Noisy Type →	Clean	Real		Symmetric			Asymmetric		Mean
Pruning Method ↓		Real-A	Real-W	0.2	0.5	0.8	0.2	0.4	Δ
Full-training	95.6	90.7	78.3	91.3	85.4	65.6	90.6	85.8	–
Prune Ratio ∼30%									
Static Random	94.6 −1.0	89.9 −0.7	75.0 −3.3	89.0 −2.3	81.2 −4.2	57.6 −8.0	87.9 −2.8	82.5 −3.3	-3.2
SmallL [19]	88.9 −6.7	87.2 −3.5	78.2 −0.1	89.3 −2.0	72.1 −13.3	24.0 −41.6	87.4 −3.3	68.9 −16.9	-10.9
Margin [7]	94.5 −1.1	88.9 −1.8	64.6 −13.7	85.4 −5.9	15.0 −70.5	12.0 −53.6	79.8 −10.8	47.6 −38.2	-24.4
Forget [52]	93.6 −2.0	89.5 −1.1	69.6 −8.8	84.5 −6.8	59.0 −26.4	22.5 −43.1	80.5 −10.2	58.3 −27.5	-15.7
GraNd [40]	93.9 −1.7	87.8 −2.9	40.5 −37.8	65.9 −25.4	17.5 −68.0	11.0 −54.6	60.5 −30.1	38.7 −47.1	-33.4
Glister [21]	93.1 −2.5	88.7 −2.0	75.6 −2.8	83.8 −7.5	73.1 −12.4	37.1 −28.6	84.1 −6.5	74.3 −11.5	-9.2
Moderate [56]	92.1 −3.5	89.9 −0.7	66.8 −11.5	87.8 −3.5	52.8 −32.6	17.8 −47.9	80.8 −9.9	58.3 −27.5	-17.1
SSP [48]	93.4 −2.2	88.6 −2.1	66.0 −12.3	83.4 −7.9	53.1 −32.4	16.6 −49.0	80.3 −10.4	59.5 −26.3	-17.8
Prune4ReL† [39]	93.0 −2.6	88.3 −2.4	66.1 −12.3	81.8 −9.5	52.2 −33.2	15.8 −49.8	81.1 −9.6	59.0 −26.8	-18.3
Dyn-Unc [16]	95.3 −0.3	89.6 −1.0	78.7 +0.4	89.6 −1.7	78.6 −6.9	44.8 −20.8	82.6 −8.0	76.1 −9.7	-6.0
DUAL [6]	95.5 −0.1	90.1 −0.5	80.4 +2.0	91.7 +0.4	79.2 −6.2	40.2 −25.5	87.1 −3.6	67.8 −18.0	-6.4
Dynamic Random [36]	94.8 −0.8	90.3 −0.3	77.3 −1.0	90.8 −0.5	84.2 −1.2	64.8 −0.8	91.5 +0.9	84.5 −1.3	-0.6
DivBS [17]	95.3 −0.3	90.3 −0.3	78.3 +0.0	85.3 −6.0	79.2 −6.3	50.3 −15.3	88.0 −2.6	82.8 −3.0	-4.2
IES [61]	95.0 −0.6	89.8 −0.8	79.8 +1.5	87.5 −3.8	80.0 −5.4	46.0 −19.7	88.9 −1.8	82.3 −3.5	-4.3
InfoBatch [42]	95.3 −0.3	91.0 +0.3	78.1 −0.3	92.0 +0.7	85.6 +0.1	67.5 +1.9	91.3 +0.7	87.0 +1.3	+0.5
InfoBatch + Ours	95.3 −0.3	91.4 +0.7	82.0 +3.7	92.7 +1.4	87.9 +2.5	67.8 +2.2	92.9 +2.3	90.0 +4.2	**+2.1**
SeTa [67]	95.2 −0.4	90.6 −0.1	76.3 −2.0	91.8 +0.5	84.7 −0.7	64.2 −1.5	90.8 +0.1	88.4 +2.6	-0.2
SeTa + Ours	95.3 −0.3	90.7 +0.0	79.1 +0.8	92.5 +1.2	85.0 −0.5	64.6 −1.0	91.8 +1.2	89.3 +3.5	**+0.6**
Prune Ratio ∼50%									
Static Random	93.3 −2.3	88.6 −2.1	72.4 −5.9	87.4 −3.9	77.5 −8.0	51.4 −14.2	85.8 −4.9	77.9 −7.9	-6.1
SmallL [19]	84.1 −11.5	84.1 −6.6	78.0 −0.3	84.7 −6.6	80.8 −4.6	30.9 −34.7	84.7 −6.0	60.9 −24.9	-11.9
Margin [7]	93.3 −2.3	85.2 −5.5	44.6 −33.8	70.7 −20.7	13.5 −72.0	10.7 −54.9	57.5 −33.1	38.6 −47.2	-33.7
Forget [52]	92.4 −3.2	88.4 −2.3	72.2 −6.1	85.7 −5.6	68.3 −17.2	29.8 −35.8	79.9 −10.7	57.4 −28.4	-13.7
GraNd [40]	91.2 −4.4	78.6 −12.0	21.0 −57.4	39.2 −52.1	7.1 −78.4	6.9 −58.7	37.2 −53.5	32.3 −53.5	-46.2
Glister [21]	91.4 −4.2	86.8 −3.8	71.8 −6.6	80.2 −11.1	67.9 −17.5	32.8 −32.8	80.9 −9.7	71.5 −14.3	-12.5
Moderate [56]	90.8 −4.8	88.2 −2.4	66.9 −11.4	89.9 −1.4	48.8 −36.6	17.4 −48.2	80.1 −10.5	57.7 −28.1	-17.9
SSP [48]	91.5 −4.1	87.1 −3.6	65.8 −12.6	81.2 −10.1	50.7 −34.8	16.3 −49.4	79.4 −11.2	58.6 −27.2	-19.1
Prune4ReL† [39]	91.6 −4.1	85.9 −4.8	63.9 −14.5	78.4 −12.9	49.6 −35.9	16.3 −49.3	78.8 −11.8	56.7 −29.1	-20.3
Dyn-Unc [16]	95.4 −0.2	89.3 −1.4	78.8 +0.5	90.9 −0.4	84.5 −1.0	47.1 −18.5	80.7 −10.0	71.2 −14.6	-5.7
DUAL [6]	95.4 −0.2	90.1 −0.6	81.3 +3.4	92.0 +0.7	83.6 −1.8	41.8 −23.8	89.0 −1.7	75.3 −10.5	-4.3
Dynamic Random [36]	94.5 −1.1	89.2 −1.5	76.1 −2.3	88.9 −2.4	83.7 −1.8	63.3 −2.3	90.6 +0.0	83.7 −2.1	-1.7
DivBS [17]	94.9 −0.7	89.8 −0.8	78.0 −0.4	85.9 −5.4	79.6 −5.9	48.6 −17.0	88.8 −1.8	84.6 −1.2	-4.2
IES [61]	95.0 −0.6	89.8 −0.8	78.6 +0.2	87.0 −4.3	79.1 −6.4	35.7 −29.9	88.5 −2.2	78.3 −7.5	-6.4
InfoBatch [42]	95.0 −0.6	90.5 −0.2	77.7 −0.6	92.0 +0.7	87.5 +2.1	64.3 −1.3	92.0 +1.3	87.5 +1.7	+0.4
InfoBatch + Ours	95.0 −0.6	91.3 +0.6	82.4 +4.1	92.8 +1.5	87.6 +2.1	64.5 −1.1	92.1 +1.5	89.8 +4.0	**+1.5**
SeTa [67]	94.8 −0.8	89.4 −1.3	78.5 +0.2	91.3 +0.0	86.0 +0.5	53.0 −12.7	91.7 +1.1	87.9 +2.1	-1.4
SeTa + Ours	94.9 −0.7	89.7 −1.0	79.3 +1.0	91.9 +0.6	87.6 +2.2	60.2 −5.5	92.2 +1.6	88.5 +2.7	**+0.1**
Prune Ratio ∼70%									
Static Random	90.2 −5.4	86.6 −4.1	69.0 −9.3	85.3 −6.0	73.8 −11.6	41.4 −24.3	82.3 −8.3	75.3 −10.5	-9.9
SmallL [19]	75.1 −20.5	76.3 −14.3	74.4 −4.0	78.2 −13.1	76.1 −9.4	35.4 −30.2	73.1 −17.6	62.7 −23.1	-16.5
Margin [7]	91.2 −4.4	71.7 −19.0	15.1 −63.3	22.6 −68.7	12.3 −73.1	9.8 −55.8	34.0 −56.7	27.3 −58.5	-49.9
Forget [52]	89.8 −5.8	86.2 −4.5	70.9 −7.5	85.5 −5.8	71.5 −13.9	33.0 −32.7	78.4 −12.2	55.2 −30.6	-14.1
GraNd [40]	78.1 −17.5	52.1 −38.6	10.2 −68.1	14.3 −77.0	8.2 −77.2	7.8 −57.8	18.9 −71.8	28.5 −57.3	-58.1
Glister [21]	88.5 −7.1	83.7 −7.0	65.1 −13.3	75.0 −16.4	58.3 −27.1	37.1 −28.6	75.3 −15.4	64.7 −21.1	-17.0
Moderate [56]	87.8 −7.8	84.8 −5.8	66.3 −12.0	87.3 −4.0	47.8 −37.7	16.7 −48.9	77.7 −13.0	55.4 −30.4	-19.9
SSP [48]	87.8 −7.8	82.0 −8.7	60.5 −17.8	73.9 −17.4	41.2 −44.2	15.8 −49.9	74.7 −15.9	55.1 −30.7	-24.0
Prune4ReL† [39]	88.4 −7.2	82.2 −8.5	59.7 −18.6	73.3 −18.0	43.5 −42.0	16.2 −49.4	74.2 −16.5	56.1 −29.7	-23.7
Dyn-Unc [16]	92.2 −3.4	86.3 −4.3	79.6 +1.2	90.1 −1.2	86.8 +1.3	47.2 −18.5	81.9 −8.8	71.4 −14.4	-6.0
DUAL [6]	93.5 −2.1	90.3 −0.3	80.4 +2.1	91.7 +0.4	86.6 +1.1	45.2 −20.5	90.8 +0.1	75.2 −10.6	-3.7
Dynamic Random [36]	93.0 −2.6	87.4 −3.2	75.9 −2.5	87.4 −3.9	84.6 −0.8	60.2 −5.5	91.4 +0.8	86.8 +1.0	-2.1
DivBS [17]	94.8 −0.8	89.8 −0.9	79.8 +1.5	86.2 −5.1	80.3 −5.1	48.5 −17.1	90.0 −0.6	83.8 −2.0	-3.8
IES [61]	94.5 −1.1	89.0 −1.6	76.7 −1.6	86.1 −5.3	74.6 −10.9	35.4 −30.3	87.6 −3.0	73.3 −12.5	-8.3
InfoBatch [42]	94.3 −1.3	89.9 −0.8	79.4 +1.0	92.4 +1.1	87.0 +1.6	61.0 −4.7	92.5 +1.8	88.3 +2.5	+0.2
InfoBatch + Ours	94.3 −1.3	90.8 +0.1	81.0 +2.7	92.8 +1.5	87.9 +2.5	61.1 −4.5	92.9 +2.3	89.2 +3.4	**+0.8**
SeTa [67]	94.3 −1.3	89.5 −1.2	77.7 −0.6	92.2 +0.9	87.1 +1.7	30.4 −35.2	92.2 +1.5	88.1 +2.3	-4.0
SeTa + Ours	94.5 −1.1	90.0 −0.7	79.4 +1.1	92.6 +1.3	87.7 +2.3	58.0 −7.6	92.5 +1.9	88.5 +2.7	**+0.0**

Table D. **Classification results of re-labeling integration on CIFAR-100N with ResNet-18.** We combine AlignPrune with SOP+ [29] and compare it against re-labeling-augmented pruning baselines. Performance gaps relative to the full-training setting are indicated as superscripts. The mean Δ is computed across all noise types.

Noisy Type →	Real	Symmetric			Asymmetric		Mean
Pruning Method ↓		0.2	0.5	0.8	0.2	0.4	Δ
Full-training	56.1	71.4	58.6	39.8	72.4	63.3	–
Prune Ratio ∼30%							
Prune4ReL [39]	64.4 $^{+8.3}$	73.0 $^{+1.7}$	68.2 $^{+9.6}$	22.3 $^{-17.5}$	72.5 $^{+0.1}$	65.9 $^{+2.6}$	+0.8
InfoBatch [42]	66.5 $^{+10.4}$	74.0 $^{+2.6}$	68.9 $^{+10.3}$	21.5 $^{-18.3}$	74.6 $^{+2.2}$	71.7 $^{+8.4}$	+2.6
InfoBatch + Ours	67.5 $^{+11.4}$	76.3 $^{+5.0}$	70.4 $^{+11.8}$	32.5 $^{-7.3}$	75.8 $^{+3.5}$	72.9 $^{+9.6}$	**+5.7**
Prune Ratio ∼50%							
Prune4ReL [39]	62.2 $^{+6.1}$	69.9 $^{-1.5}$	63.5 $^{+4.9}$	23.8 $^{-16.0}$	68.0 $^{-4.4}$	61.1 $^{-2.1}$	-2.2
InfoBatch [42]	66.8 $^{+10.7}$	75.2 $^{+3.8}$	69.0 $^{+10.4}$	24.0 $^{-15.8}$	74.3 $^{+1.9}$	71.6 $^{+8.4}$	+3.2
InfoBatch + Ours	67.6 $^{+11.5}$	76.2 $^{+4.9}$	71.6 $^{+12.9}$	32.9 $^{-7.0}$	75.8 $^{+3.4}$	73.1 $^{+9.8}$	**+5.9**
Prune Ratio ∼70%							
Prune4ReL [39]	56.9 $^{+0.7}$	62.9 $^{-8.5}$	57.1 $^{-1.5}$	15.1 $^{-24.7}$	62.5 $^{-9.8}$	54.6 $^{-8.7}$	-8.7
InfoBatch [42]	65.8 $^{+9.7}$	74.5 $^{+3.2}$	69.4 $^{+10.8}$	20.5 $^{-19.3}$	74.2 $^{+1.8}$	70.7 $^{+7.4}$	+2.3
InfoBatch + Ours	66.0 $^{+9.9}$	76.4 $^{+5.0}$	70.5 $^{+11.9}$	30.7 $^{-9.1}$	75.2 $^{+2.8}$	72.3 $^{+9.0}$	**+4.9**

Table E. **Classification results of re-labeling integration on CIFAR-10N with ResNet-18.** We combine AlignPrune with SOP+ [29] and compare it against re-labeling-augmented pruning baselines. Performance gaps relative to the full-training setting are indicated as superscripts. The mean Δ is computed across all noise types.

Noisy Type →	Real		Symmetric			Asymmetric		Mean
Pruning Method ↓	Real-A	Real-W	0.2	0.5	0.8	0.2	0.4	Δ
Full-training	90.7	78.3	91.3	85.4	65.6	90.6	85.8	–
Prune Ratio ∼30%								
Prune4ReL [39]	94.0 $^{+3.4}$	89.7 $^{+11.4}$	94.2 $^{+2.9}$	91.1 $^{+5.7}$	56.5 $^{-9.2}$	94.0 $^{+3.4}$	92.0 $^{+6.2}$	+3.4
InfoBatch [42]	94.8 $^{+4.1}$	91.1 $^{+12.8}$	95.2 $^{+3.9}$	94.0 $^{+8.6}$	66.8 $^{+1.2}$	94.7 $^{+4.1}$	93.2 $^{+7.4}$	+6.0
InfoBatch + Ours	95.1 $^{+4.4}$	91.7 $^{+13.4}$	95.8 $^{+4.5}$	94.6 $^{+9.1}$	70.7 $^{+5.1}$	95.5 $^{+4.9}$	93.5 $^{+7.8}$	**+7.0**
Prune Ratio ∼50%								
Prune4ReL [39]	92.6 $^{+1.9}$	88.4 $^{+10.1}$	92.8 $^{+1.5}$	88.7 $^{+3.3}$	49.5 $^{-16.2}$	92.9 $^{+2.2}$	89.9 $^{+4.1}$	+1.0
InfoBatch [42]	94.9 $^{+4.2}$	90.0 $^{+11.7}$	95.0 $^{+3.7}$	92.8 $^{+7.4}$	66.8 $^{+1.1}$	94.7 $^{+4.1}$	92.0 $^{+6.2}$	+5.5
InfoBatch + Ours	95.0 $^{+4.4}$	90.0 $^{+11.7}$	96.0 $^{+4.7}$	94.7 $^{+9.2}$	71.1 $^{+5.4}$	95.3 $^{+4.6}$	93.2 $^{+7.4}$	**+6.8**
Prune Ratio ∼70%								
Prune4ReL [39]	90.2 $^{-0.5}$	84.0 $^{+5.7}$	90.2 $^{-1.1}$	83.3 $^{-2.2}$	42.8 $^{-22.8}$	89.8 $^{-0.8}$	84.0 $^{-1.8}$	-3.3
InfoBatch [42]	94.0 $^{+3.3}$	87.7 $^{+9.4}$	94.5 $^{+3.2}$	91.3 $^{+5.9}$	51.4 $^{-14.3}$	92.4 $^{+1.7}$	90.4 $^{+4.6}$	+2.0
InfoBatch + Ours	94.0 $^{+3.3}$	87.2 $^{+8.8}$	95.1 $^{+3.8}$	92.2 $^{+6.8}$	58.5 $^{-7.2}$	94.8 $^{+4.1}$	91.7 $^{+5.9}$	**+3.7**

Table F. **Average ratio of pruned samples per epoch.** Results are reported with ResNet-18 under same prune probability.

	InfoBatch	**Ours**
Avg. pruning ratio per epoch	28.77%	**32.94%**

B.2.2. Analysis on Trajectory Window Size

We provide more results on the effect of trajectory window size N below. We expand the ablation study in Fig. 4b to include smaller (2, 3, 4) and larger window sizes (30, 40, 50) as shown in Table G. Results show that the proposed AlignPrune performs consistently with window size ranging from 4 to 50 across all noisy-label types, with minimal variance presented. However, when using extremely small window size 2 or 3, the performance show significant drop, where the loss trajectory under this condition cannot fully capture the learning dynamics as expected. This confirms that the necessity of such trajectory window design, but also shows that AlignPrune remains robust across a practical and appropriate range of values.

B.2.3. Analysis on Correlation Function

We provide more results on the choice of correlation function ρ below. We expand the ablation study in Fig. 4c by additionally evaluating Dynamic Time Warping (DTW) as

Table G. **Ablation on AlignPrune with varying window sizes.** We evaluate AlignPrune using window sizes ranging from 2 to 50, aiming to validate the necessity of such trajectory window design.

Noisy Type →	Clean	Real	Symmetric			Asymmetric	
Window Size N ↓			0.2	0.5	0.8	0.2	0.4
2	77.3	55.2	70.3	58.8	40.3	71.4	63.2
3	78.0	55.9	70.7	59.2	40.8	71.2	63.1
4 - 50 (avg ± std)	78.9 ± 0.2	56.5 ± 0.3	71.3 ± 0.3	60.4 ± 0.3	41.4 ± 0.3	72.2 ± 0.2	64.3 ± 0.3

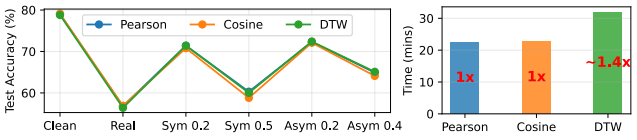

Figure A. **Ablation on ρ: Accuracy (Left); Time Cost (Right).**

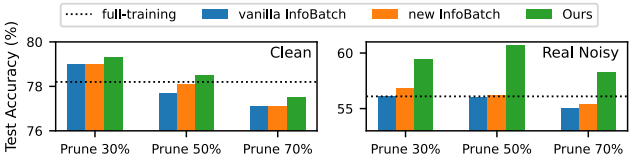

Figure B. **Comparison between calibrated InfoBatch and Ours.**

the correlation function. As illustrated in Fig. A, while DTW achieves comparable accuracy to both Pearson and Cosine similarity across all noise types, a substantially higher computational time cost can be observed. We therefore keep to use Pearson by default to maintain optimal training efficiency without sacrificing robustness.

B.2.4. Analysis on Fairness under Extra Reference Supervision

To ensure the performance gains of AlignPrune do not stem solely from access to the extra reference data, we calibrate the loss threshold of InfoBatch using the same reference supervision. Specifically, we define a sample remaining region via epoch-wise reference loss statistics (using the mean and standard deviations). As shown in Fig. B, this calibrated strategy yields negligible improvement over vanilla InfoBatch, indicating the bottleneck lies in the loss-based ranking metric itself, not the access to the clean reference set.

B.2.5. Analysis on Dependence of Clean Data

We provide further discussions on the intuitions behind the effectiveness of AlignPrune with extremely limited or even without clean references.

Discussion on Reference Set Scale. From an intuitive perspective, DAS measures the degree of alignment between a sample's loss trajectory and the principal learning dynamics of the target task captured by the clean reference set. Even when the clean reference set is small, its averaged trajectory remains a stable indicator of the model's intended learning trend, because correlation used during DAS computation focuses on *relative trends* rather than absolute loss magnitudes, which is robust to scale differences. To provide evidence from results,

- Results in Fig. 5 show remarkable stability: with only 0.1% clean fraction, DAS provides enough signal to prune noisy samples while preserving informative ones, except under extreme noise (symmetric noise with 0.8 rate), where the clean signal becomes too weak.
- Table 5 provides a comparison with *Clean Validation-set* versus *Noisy Train-set* as the reference dataset. Performance remains consistent for clean-label setting but degrades significantly for noisy-label case when using noisy reference data. This confirms that clean reference data is crucial for providing the accurate principal learning dynamics.

In summary, these results indicate that DAS does not depend heavily on the quantity of clean data, but rather on its ability to provide a reliable signal of the principal learning dynamics.

Discussion on Reference Set Quality. Using a pseudo-clean set as reference remains effective because a coreset selection method can *filter* the noisy dataset to produce a subset whose average learning dynamic is statistically closer to the true clean dynamic, which can be reflected from the downstream performance of the selected coreset. Even if this reference set is not perfectly purely clean, it has a lower noise level compared to the original noisy set, making its average trajectory a more reliable signal. To provide evidence from results,

- Table 5 shows that using high-quality pseudo-clean subsets (estimated by SmallL or Moderate) yields performance nearly identical to using a real clean reference set, demonstrating the practical viability of this approach in scenarios when clean data is unavailable.
- Table 5 also provides a comparison with *Noisy Train-set* versus *Pseudo Clean-set*. Performance degrades significantly under noisy scenarios, which further confirms that the quality and correctness of the reference set, not just its availability, are critical.

In summary, these results demonstrate that AlignPrune's effectiveness is not contingent on a purely clean reference set. Instead, it highlights the method's practical flexibility: it can remain effective provided with a coreset method capable of extracting a pseudo-clean subset that is reasonably pure

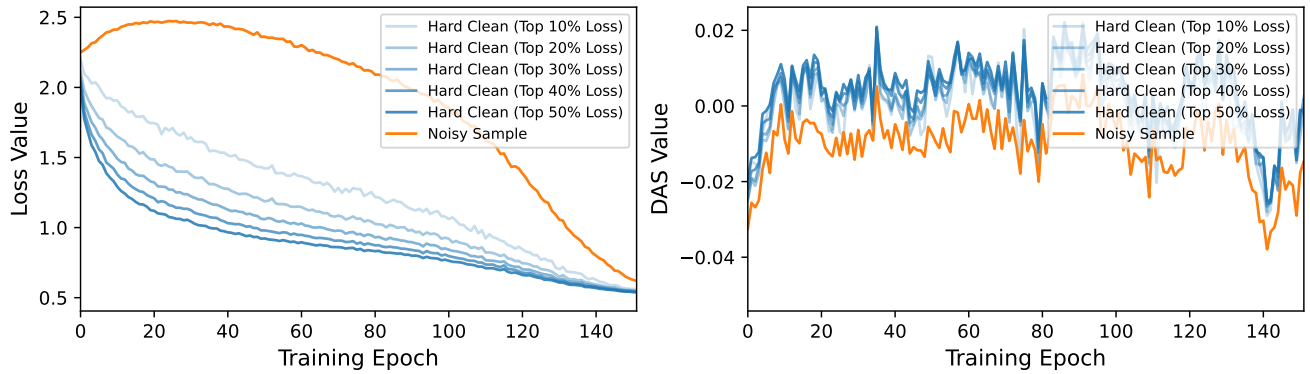

Figure C. **Loss and DAS curves of hard *vs*. noisy samples.** Hard clean samples are identified as correctly labeled instances with the highest per-sample average loss across training, where Top x% Loss denotes the top x% of clean samples by average loss.

Table H. **More results with estimated pseudo clean set.**

Reference-set Fraction →	100%		10%		1%	
Reference-set Type ↓	Clean	Real	Clean	Real	Clean	Real
Clean	79.0	56.8	78.9	56.3	79.1	56.0
Pseudo by ELFS [66]	78.8	56.6	79.1	56.3	78.9	56.1

Table I. **Ablation on AlignPrune with varying noise rates in reference.** We inject controlled noise into the reference to validate its noise sensitivity.

Ref. Noise (%) →	0%	10%	20%	30%	40%
Acc. ± std	56.8 ± 0.1	56.5 ± 0.2	56.4 ± 0.1	56.2 ± 0.2	53.7 ± 0.1

to offer a reliable reference trajectory.

More Results with Pseudo Clean Data. To further demonstrate that AlignPrune does not strictly require any clean reference labels, we evaluate its performance using a pseudo-clean reference set estimated by one more recent label-free coreset method ELFS [66]. As shown in Table H, similarly consistent performance can be observed compared to using a clean validation reference set, further confirming AlignPrune's robustness in fully unsupervised noisy regimes.

Discussion on Sensitivity to Reference Noise. To address concerns regarding the purity of the reference set, we conduct a sensitivity analysis by injecting controlled synthetic noise directly into the reference. As shown in Table I, AlignPrune's performance on CIFAR-100N remains remarkably stable with reference noise ratios **up to 30%**, dropping marginally for rates beyond 40%, which further confirms the robustness of our trajectory alignment metric even under highly imperfect reference supervision.

Remark. A broader study of distribution shift between the reference set and training distribution is orthogonal to our focus and is left to future work.

B.2.6. Analysis on Effectiveness under Clean Labels

By definition, DAS of one sample measures the degree of alignment between its loss trajectory and the average trajectory of the reference set, reflecting whether the sample has the synchronized behavior as the clean reference samples. Additionally, reference samples typically reflect the principal learning dynamics of the target task. Thus, a sample with high DAS exhibits behavior that is more synchronized with principal learning dynamics, suggesting that it is more *informative* and *contributes meaningfully* to generalization.

Under clean-label scenario, this behavior implies that AlignPrune naturally degrades into a surrogate of conventional dynamic pruning method. In other words, when noise is absent, DAS serves as a proxy for identifying informative samples in a way that is consistent with established pruning methods such as InfoBatch or SeTa. This dual behavior, robust under noisy-label and consistent under clean-label, supports the versatility of AlignPrune across both noisy and noise-free settings.

B.2.7. Analysis on Hard *vs.* Noisy Samples

A critical challenge in robust learning is distinguishing *hard-but-clean* samples from *truly-noisy* ones, as both typically exhibit high loss values compared to *easy-clean* samples. However, their learning dynamics differ fundamentally. We plot the per-sample loss-value curves and DAS-value curves for these *hard-clean* and *noisy* samples in Fig. C. Hard clean samples are identified as correctly labeled instances with the highest per-sample average loss across training.

On the loss plots, loss values of hard clean samples are initially high, overlapping in magnitude with the noisy ones, which are nearly indistinguishable. This confirms why loss-based methods fail in the noisy setting. In contrast, the DAS curves clearly separate the two groups. Hard clean samples, despite their high loss, demonstrate a consistent and monotonic decrease that correlates positively with the reference trend, yielding persistently higher DAS. Noisy samples exhibit a more erratic and inconsistent pattern, resulting in

Table J. **Statistical significance analysis on CIFAR-100N with ResNet-18.**

Noisy Type →	Clean	Real	Symmetric			Asymmetric	
Pruning Method ↓			0.2	0.5	0.8	0.2	0.4
InfoBatch	79.0 ± 0.1	56.1 ± 0.2	71.4 ± 0.2	59.7 ± 0.2	**41.8 ± 0.1**	71.9 ± 0.2	64.2 ± 0.2
InfoBatch + Ours	**79.3 ± 0.1**	**59.4 ± 0.2**	**71.8 ± 0.1**	**66.0 ± 0.2**	**41.8 ± 0.1**	**72.6 ± 0.1**	**68.0 ± 0.2**
SeTa	79.0 ± 0.1	55.6 ± 0.1	70.2 ± 0.1	59.0 ± 0.2	**41.6 ± 0.1**	71.4 ± 0.2	63.2 ± 0.2
SeTa + Ours	**79.3 ± 0.1**	**56.3 ± 0.1**	**70.8 ± 0.2**	**60.5 ± 0.2**	**41.6 ± 0.1**	**71.9 ± 0.1**	**64.3 ± 0.2**

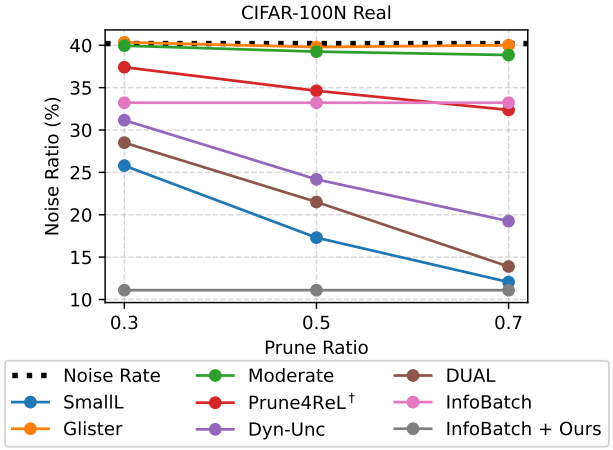

Figure D. **Noise ratio in the retained subset on CIFAR-100N with Real noise-label.** Best viewed in color.

consistently low or negative correlations. These observations confirm that DAS is orthogonal to loss magnitude, enabling AlignPrune to robustly retain *hard-but-clean* samples while filtering out *truly-noisy* ones.

B.2.8. Analysis on Noise Ratio in Retained Subset

To quantitatively verify that AlignPrune effectively filters out noisy samples, we measure the fraction of noisy samples in the retained training subset across various pruning ratios. Fig. D reports this fraction as noise ratio by each method on CIFAR-100N with 40.2% Real noisy-label.

As shown, the subset obtained by InfoBatch + Ours (AlignPrune) yields a significantly lower noise ratio compared to both static baselines (SmallL, Prune4ReL) and dynamic baselines (InfoBatch). This improved subset purity confirms that the performance gains observed in our main results are attributable to the more precise identification and removal of noisy samples.

B.2.9. Analysis on Preservation of Unbiasedness

Existing dynamic pruning methods guarantee unbiased gradient estimation by applying expectation rescaling to the gradients of the retained samples, which is mathematically independent of how the subset is chosen. Our proposed AlignPrune operates solely as a plug-and-play replacement for the ranking metric, thereby strictly preserving this unbiased gradient estimation guarantee.

Table K. **Classification results on NEWS with 3-layer MLP.**

Noisy Type →	Clean	Symmetric		Pairflip
Pruning Method ↓		0.2	0.5	0.45
Full-training	42.5	37.1	26.7	26.5
InfoBatch	42.8 +0.3	36.7 −0.4	26.2 −0.6	27.4 +0.9
InfoBatch + Ours	**42.9** +0.4	**37.9** +0.8	**28.4** +1.7	**27.7** +1.2

B.3. Additional Validation Beyond Image

To further demonstrate the broad applicability of AlignPrune, we evaluate it on the NEWS dataset from [22] for text classification. We follow the experimental settings in [60] to inject the symmetric and pairflip label noise with rates 0.2, 0.5 and 0.45, respectively. As shown in Tab. K, AlignPrune consistently outperforms baseline methods across various types of label noise, which confirms its effectiveness in different modalities beyond the image domain.

B.4. Additional Statistical Significance Analysis

As mentioned in the experimental setup, each experiment is repeated three times, and we report the average accuracy across the runs. Here, we provide a statistical significance analysis corresponding to the results in Tab. 1 under prune ratio of 30%. Results in Tab. J present the mean accuracy along with standard deviations, demonstrating the statistical reliability and stability of AlignPrune. These results confirm the consistency of performance gains observed with our approach.