THEORY-GUIDED INSTANCE-DEPENDENT EARLY STOPPING VIA SECOND-ORDER LOSS DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Instance-dependent early stopping (IES) reduces training computation by skipping backpropagation for examples whose losses have stabilized, but the second-order loss-difference criterion used by IES lacks a theoretical justification. We develop Theory-Guided Instance-Dependent Early Stopping (TG-IES), which analyzes this criterion under explicit assumptions and derives a noise-calibrated mastery threshold and a conditional stability bound for instance exclusion. We show that second-order differencing has a favorable bias–variance trade-off over first- and third-order differences in a specific signal-to-noise regime, and that its signal contains both gradient-drift and curvature terms. TG-IES derives its stopping threshold and patience from training-run statistics rather than using the manually selected threshold of IES. Across CIFAR-10 and CIFAR-100, TG-IES saves 53–58% of backpropagation instances in its main convolutional settings while keeping accuracy within 0.1–0.9 percentage points of full training; in the primary ResNet-18/CIFAR-10 setting, this corresponds to 35.7% FLOP-normalized savings. The method also transfers to ViT-Tiny, where it achieves smaller savings and incurs a modest accuracy cost. These results provide a theoretical basis for second-order instance-dependent early stopping and show that theory-guided calibration can substantially reduce training computation under the assumptions and experimental settings studied. The code is available at https://anonymous.4open.science/r/TG_IES_Results-8EA0.

1 INTRODUCTION

The computational resources required to train deep learning models have increased substantially (Sevilla et al., 2022). As a result, the energy use and carbon footprint of model training are also closely tied to the amount of computation required (Patterson et al., 2022). Algorithmic methods that reduce unnecessary forward and backward computation can lower this cost without changing the model architecture or hardware (Bartoldson et al., 2023).

Training modern neural networks repeatedly processes the same training instances, even though different instances are learned at different rates. Here, a *training instance* means one individual training sample, such as an input–label pair in a classification dataset. Studies of training dynamics show that some instances become easy to learn and remain stable, while others continue to be difficult or important for learning (Toneva et al., 2019; Swayamdipta et al., 2020; Paul et al., 2021). This difference in learning behavior suggests that not every instance needs the same amount of computation throughout training.

Several methods use this observation to reduce training cost. One group selects or prunes instances using static, one-shot, or early-estimated measures of importance, difficulty, or diversity (Paul et al., 2021; Sorscher et al., 2022; Zheng et al., 2023; Maharana et al., 2024). Another group makes the decision during training using signals such as loss, gradient information, or other measures of instance importance (Loshchilov & Hutter, 2015; Katharopoulos & Fleuret, 2018; Jiang et al., 2019; Mindermann et al., 2022; Qin et al., 2024; Bankes et al., 2024; Nagaraj et al., 2025; Jin et al., 2026). These methods share the goal of spending more computation on instances that are still useful for learning instead of treating all instances equally at every step. Recent work also shows that an instance's *loss trajectory*, meaning its sequence of loss values over training, can provide information about its learning behavior beyond a single loss value (Nagaraj et al., 2025).

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.

Our work focuses on a more specific question: *when does an individual instance no longer need a gradient update?* Yuan et al. (2025) introduced *Instance-dependent Early Stopping* (IES), which stops backpropagation for an instance when its loss becomes sufficiently stable. The IES criterion uses the second-order finite difference of the instance's loss:

$$\left|\Delta^2 L_i(\theta^{(t)})\right| = \left|L_i^{(t)} - 2L_i^{(t-1)} + L_i^{(t-2)}\right| < \delta, \tag{1}$$

Here, $L_i^{(t)}$ is the loss of instance i at epoch t. The first difference $L_i^{(t)} - L_i^{(t-1)}$ measures how much the loss moved and $\Delta^2 L_i$ measures how much that move itself changed. So a small $|\Delta^2 L_i|$ means the loss trajectory has become locally stable. In IES, an instance is *mastered* when this value stays below δ for a fixed number of epochs, and its gradient computation is then skipped. "Mastered" is an operational name for a stable loss trajectory. It does not imply zero loss, a zero gradient, or perfect prediction.

IES shows that this simple loss-based signal can reduce training computation, but several questions remain open. The original work does not explain why the second-order difference is a useful mastery signal, how its noise compares with first- and third-order differences, or how it relates to the optimization dynamics. It also does not analyze how excluding instances changes the training trajectory. Finally, the mastery threshold δ and the patience value are manually chosen rather than derived from the training run.

Research question. *Can we explain why the second-order loss difference works as a mastery signal, derive its threshold from observable training dynamics, and characterize how excluding instances affects the training trajectory?*

Our answer has three parts. First, under a local near-convergence model (Section 4), we analyze the second-order loss difference and explain when it is a useful signal. We show that relative loss changes can remove the effect of an instance's loss scale, that $N = 2$ has a favorable bias–variance trade-off in a specific signal-to-noise regime and that the second-order difference contains both gradient-drift and curvature terms.

Second, we use these results to develop a practical training procedure called **Theory-Guided Instance-Dependent Early Stopping (TG-IES)** (Section 5). TG-IES estimates the noise level during training and uses it to set the mastery threshold δ. It also determines the patience value K^*, which specifies how many consecutive low-signal epochs an instance must have before it is excluded from training. We initialize K^* using a conditional reference horizon K_{safe}. Finally, we show that excluding these instances is equivalent to using an approximate gradient update. The difference between this update and the full-training update is exactly the combined gradient of the excluded instances. We then bound how much this difference can change the training trajectory.

Third, we evaluate the theory and the method across datasets, architectures, training schedules, and competing methods, measuring both computational savings and accuracy and directly testing the assumptions behind the stability analysis (Sections 7 and 8). Proofs and further results are in the Appendix.

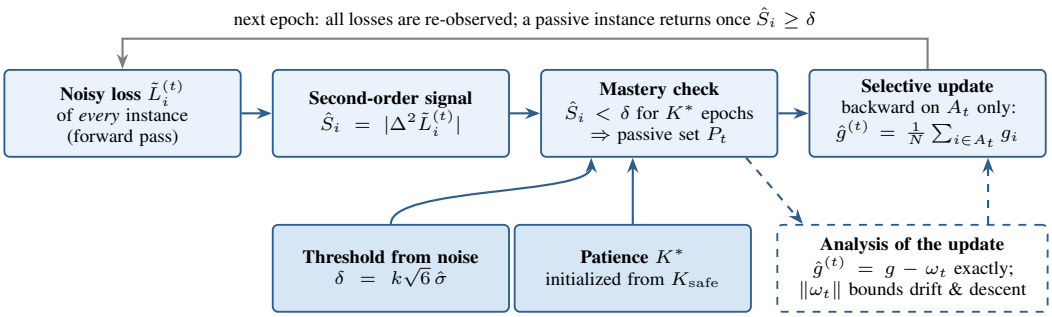

Figure 1: TG-IES uses each instance's loss trajectory to derive a second-order loss signal. Observed noise sets the mastery threshold δ, and a conditional reference horizon K_{safe} initializes patience K^*. The mastery rule decides when an instance becomes passive and when it re-enters training. Skipping passive instances alters updates by the passive-set gradient ω_t, governing drift and descent (dashed box). Experiments assess compute savings, accuracy, and the method's assumptions.

Table 1: Notation used in the main text.

Symbol	Meaning	Symbol	Meaning
$L_i^{(t)}$	loss of instance i after epoch t	$\theta^{(t)}$, η_t	parameters after epoch t; step size
$\tilde{L}_i^{(t)}$	recorded loss $L_i^{(t)} + \xi_{i,t}$ (noise scale σ)	L	training loss $\frac{1}{N}\sum_i L_i$
$\Delta^1 L_i^{(t)}$	$L_i^{(t)} - L_i^{(t-1)}$: how much the loss moved	g_i	$\nabla_\theta L_i$: update direction from instance i
$\Delta^2 L_i^{(t)}$	$\Delta^1 L_i^{(t)} - \Delta^1 L_i^{(t-1)}$: change of that move	g	$\frac{1}{N}\sum_i g_i$: full-data gradient
$\Delta^N L_i^{(t)}$	the difference applied N times	H_i	$\nabla_\theta^2 L_i$: local curvature of L_i
δ, K^*	mastery threshold on $\lvert\Delta^2 \tilde{L}_i\rvert$; patience	A_t, P_t	active / passive instances

Contributions. We provide a theoretical and empirical analysis of IES and develop Theory-Guided Instance-Dependent Early Stopping (TG-IES). Our contributions are:

- **Theory for the second-order loss signal.** We study when the second-order loss difference is useful. We analyze its ratio-level scale cancellation, the bias–variance trade-off, and the effects of gradient drift and curvature.
- **Theory-guided calibration.** We use the noise observed during training to set the mastery threshold δ. We also use a conditional reference horizon to initialize the patience value K^*.
- **Conditional stability of the selective update.** We analyze how excluding instances affects training. We bound the resulting trajectory drift and loss descent using the combined gradient of the excluded instances.
- **Empirical evaluation.** We evaluate TG-IES across different datasets, model architectures, optimizers, and training settings. We measure computation savings and accuracy, and evaluate the assumptions used in the stability analysis.

2 Related Work

Static data pruning. Static data pruning selects a fixed subset of training examples before or near the beginning of training. Existing methods use forgetting events (Toneva et al., 2019), early gradient-norm scores (Paul et al., 2021), training-dynamics statistics (Swayamdipta et al., 2020), or scores obtained from a smaller proxy model (Coleman et al., 2019). Other approaches consider dataset coverage, diversity, or scaling behavior when selecting the subset (Sorscher et al., 2022; Zheng et al., 2023; Maharana et al., 2024). In contrast, IES and TG-IES make an instance-level decision at each epoch based on the current training trajectory.

Dynamic and online data selection. Dynamic methods update the selection of training examples during training. Importance-sampling approaches focus on examples that contribute more to gradient estimation (Loshchilov & Hutter, 2015; Katharopoulos & Fleuret, 2018), while Selective-Backprop (Jiang et al., 2019) uses a probabilistic rule to skip backpropagation for low-loss examples. Other methods use holdout models (Mindermann et al., 2022; Bankes et al., 2024), epoch-level loss statistics (Qin et al., 2024), trajectory-alignment scores (Qin et al., 2026; Nagaraj et al., 2025), or moving averages of loss (Zhou et al., 2026). Jin et al. (2026) provide theoretical guarantees for a different dynamic-pruning mechanism. These methods mainly focus on deciding which examples to process or how examples are selected or ranked, whereas TG-IES calibrates its exclusion rule from the observed loss noise and analyzes the gradient error that exclusion introduces.

Instance-Level Early Stopping. Curriculum learning controls when examples *enter* training, typically progressing from easier to harder examples (Bengio et al., 2009; Soviany et al., 2022). Whereas IES and TG-IES determine when an example can *leave*. Our direct predecessor is IES (Yuan et al., 2025), which introduced instance-level early stopping using a second-order loss-difference signal. TG-IES extends IES with an analysis of this signal, noise-based threshold calibration, and a conditional stability analysis of the resulting selective update.

3 Setup and Assumptions

We analyze training one instance at a time by studying the per-instance loss trajectory and then connecting it to the full training update. For each instance, we record how its loss changes from epoch to epoch. For analysis, training is idealized as gradient descent, $\theta^{(t+1)} = \theta^{(t)} - \eta_t\, g^{(t)}$ (constant $\eta_t \equiv \eta$ unless stated), where a superscript (t) marks a value at $\theta^{(t)}$. Table 1 lists the symbols used throughout.

Thus, $\Delta^2 L_i$ measures whether the per-epoch change in an instance's loss is still changing. **What we assume, and why.** Each result names the idealizations it uses; Section B adds two technical ones (Assumptions B.1 and B.2) and discusses failures, and proofs are in Section C.

Assumption 3.1 (Smoothness)**.** L is β-smooth: $\|g(\theta) - g(\theta')\| \le \beta\|\theta - \theta'\|$. *Purpose:* gradients cannot change arbitrarily fast, which turns per-step errors into multi-step bounds.

Assumption 3.2 (Local geometric decay)**.** Near a solution θ^*, each excess loss shrinks by a constant fraction per epoch: $L_i^{(t)} - L^* = c_i r_i^t$, with scale $c_i > 0$ and rate $r_i \in (0, 1)$. *Purpose:* a simple late-training model in which finite differences are exact.

Assumption 3.3 (Independent noise)**.** $\xi_{i,t}$ is conditionally mean-zero with variance σ^2, independent over epochs for fixed i. *Purpose:* noise does not persist, so its effect on Δ^N is exact (Section M).

Assumption 3.4 (Sub-Gaussian noise)**.** $\xi_{i,t}$ is conditionally σ^2-sub-Gaussian (needed only for tail bounds).

Assumption 3.5 (Small rate dispersion)**.** $r_i \in [\bar{r} - \delta_r, \bar{r} + \delta_r]$ for all i, with δ_r small. *Purpose:* broadly similar decay rates, needed for one cross-instance comparison (Proposition C.2 and Section G).

4 Why the Second-Order Difference Works

We answer three questions about the IES signal $\Delta^2 L_i$, each as: question, result, meaning, limitation.

4.1 Does the signal depend on the size of the loss?

Question. Instances differ widely in loss size, but a useful signal should reflect *how fast* an instance converges: two instances with different loss sizes and the same decay rate should look alike. Under Assumption 3.2, $\Delta^N(L_i^{(t)} - L^*) = c_i\, r_i^{t-N}(r_i - 1)^N$ (Lemma C.1), which gives the following.

Theorem 4.1 (The loss scale cancels in the ratio)**.** *Under Assumption 3.2, for every $N \ge 1$ and every instance i,*

$$\frac{\Delta^N(L_i^{(t)} - L^*)}{\Delta^{N-1}(L_i^{(t)} - L^*)} = \frac{r_i - 1}{r_i}, \tag{2}$$

where the denominator is nonzero for every finite t.

Meaning. The scale c_i cancels, so instances with different loss sizes but equal rates give the same ratio: near convergence, differences reflect *how* a loss changes, not only its size. With small rate dispersion, the spread of $|\Delta^2 L_i|$ across instances also stays within $O(\delta_r)$ of that of $|\Delta^1 L_i|$ (Proposition C.2). **Limitation.** Only the *ratio* is scale-free; $\Delta^2 L_i$ itself still grows with c_i.

4.2 Why the second order, rather than the first or third?

Question. Each extra order of differencing removes more of the remaining trend in a loss trajectory, but it also amplifies measurement noise. Which order balances the two?

Theorem 4.2 (Exact bias–variance decomposition)**.** *Under Assumptions 3.2 and 3.3,*

$$\mathbb{E}\big[(\Delta^N \tilde{L}_i^{(t)})^2\big] = \underbrace{c_i^2 r_i^{2(t-N)}(r_i - 1)^{2N}}_{\textit{remaining trend}} + \underbrace{\tbinom{2N}{N}\sigma^2}_{\textit{noise}}. \tag{3}$$

Meaning. The right side, $\mathrm{MSE}(N, t)$, is the mean squared size of the observed N-th difference. Raising N shrinks its trend term (when $r_i > \frac{1}{2}$) but grows its noise term combinatorially (2, 6, 20 for $N = 1, 2, 3$), so the balance is explicit:

Proposition 4.3 (When $N = 2$ minimizes the error)**.** *Under Assumptions 3.2 and 3.3, let $\alpha_i := (1 - r_i)^2/r_i^2$ and $r_i > \frac{1}{2}$ (so $\alpha_i < 1$). Then $N = 2$ minimizes $\mathrm{MSE}(N, t)$ over $\{1, 2, 3\}$ if and only if $c_i^2 r_i^{2t}$ lies in the non-empty signal-to-noise (SNR) band*

$$\frac{4\sigma^2}{\alpha_i(1 - \alpha_i)} \;\le\; c_i^2 r_i^{2t} \;\le\; \frac{14\sigma^2}{\alpha_i^2(1 - \alpha_i)}. \tag{4}$$

Meaning. The quantity $c_i^2 r_i^{2t}$ represents the squared remaining signal, while σ^2 represents the observation noise. The result identifies an intermediate regime during training. Early on, the learning

signal is strong enough to make higher-order differences useful. Very late in training, the signal becomes so weak that the extra noise of higher orders takes over, making the first-order difference ($N = 1$) the better choice. The second-order difference strikes the ideal balance between these two extremes, which is exactly the phase targeted by the noise-calibrated TG-IES threshold. **Limitation.** This does not show that $N = 2$ is always optimal. It is optimal only when the signal lies in the band in Equation (4).

Pure noise gives $\Delta^2\xi_{i,t} = \xi_{i,t} - 2\xi_{i,t-1} + \xi_{i,t-2}$; with independent terms, the coefficients $1, -2, 1$ give variance $(1^2 + 2^2 + 1^2)\sigma^2 = 6\sigma^2$. The threshold can therefore be tied to the measured noise scale $\hat{\sigma}$ rather than fixed in loss units.

Corollary 4.4 (Noise-calibrated threshold)**.** *Under Assumptions 3.3 and 3.4, $\Delta^2\xi_{i,t}$ has mean zero and variance $6\sigma^2$. With $\delta = k\sqrt{6}\,\sigma$, $\Pr(|\Delta^2\xi_{i,t}| > \delta) \approx 2(1 - \Phi(k))$ for approximately Gaussian noise (Φ: standard normal CDF), and $\Pr(|\Delta^2\xi_{i,t}| > \delta) \le 2\exp(-k^2/2)$ in general.*

Empirical check. At the epoch when each instance's signal first reaches the calibrated threshold (46,779 instances), $N = 2$ has the lowest empirical error, two and three orders of magnitude below $N = 1$ and $N = 3$, as predicted. A naive late-epoch average instead favors $N = 1$, because already noise-dominated instances dominate it (Section P).

4.3 What does the second difference actually measure?

Question. A second difference is often read as curvature, but the parameters also move between measurements. A step changes L_i by about $-\eta\langle g_i, g\rangle$, so $D_i^{(t)} := \langle g_i^{(t-2)}, g^{(t-2)}\rangle - \langle g_i^{(t-1)}, g^{(t-1)}\rangle$ measures the change in per-step progress on i. $O(\eta^k)$ marks a term scaling like η^k for small η.

Theorem 4.5 (What the second difference measures)**.** *Under Assumption B.2, along gradient descent with step size η,*

$$\Delta^2 L_i^{(t)} = \underbrace{\eta\, D_i^{(t)}}_{\substack{\text{change in} \\ \text{progress, } O(\eta)}} + \underbrace{\tfrac{\eta^2}{2}\big(g^{(t-1)\top} H_i(\theta^{(t-1)}) g^{(t-1)} - g^{(t-2)\top} H_i(\theta^{(t-2)}) g^{(t-2)}\big)}_{\text{change in curvature, } O(\eta^2)} + \underbrace{R_{i,t}}_{\substack{\text{remainder,} \\ O(\eta^3)}}, \quad (5)$$

with $|R_{i,t}| \le \frac{M_3}{6}\eta^3\big(\|g^{(t-1)}\|^3 + \|g^{(t-2)}\|^3\big)$.

Meaning. This equation shows that $\Delta^2 L_i$ should not be treated as a direct measure of curvature. Instead, it captures a mix of training dynamics: its main component is the change in the instance's per-step progress, while curvature only plays a secondary role at order $O(\eta^2)$. Because of this, it is better to interpret $\Delta^2 L_i$ as a general stability or convergence signal. **Limitation.** This mathematical ordering assumes very small step sizes. At the actual step sizes used in our experiments, the progress (drift) term does not completely dominate and only correlates weakly with the observed signal (Pearson $r = 0.23$–0.37, Section K). Also, a small $\Delta^2 L_i$ means the loss trajectory has flattened out and does not guarantee that the gradient itself is small (Section 5.3).

5 From the Signal to TG-IES

5.1 The algorithm

TG-IES retains the IES mastery check but calibrates its threshold and persistence from training statistics. At each epoch (Algorithm 1), it (1) forwards all instances and records $\tilde{L}_i^{(t)}$, (2) computes $\hat{S}_i = |\Delta^2 \tilde{L}_i^{(t)}|$ once three observations are available, and (3) backpropagates only through the active set

$$\hat{g}^{(t)} = \frac{1}{N}\sum_{i \in A_t} g_i(\hat{\theta}^{(t)}), \qquad \hat{\theta}^{(t+1)} = \hat{\theta}^{(t)} - \eta_t \hat{g}^{(t)}. \quad (6)$$

An instance becomes passive after K^* consecutive epochs with $\hat{S}_i < \delta$ and reactivates when $\hat{S}_i \ge \delta$. At epoch 30, $\delta = k\sqrt{6}\,\hat{\sigma}$ and K^* are calibrated as described in Section H. Here, $k = \Phi^{-1}(1 - p/2)$. The $1/N$ normalization preserves the full-training update scale.

Patience. K^* is based on the conditional reference horizon $K_{\text{safe}} = \lfloor N\tau/(\beta_{\text{test}}\eta\varepsilon_0)\rfloor$ (Corollary E.2), clipped to $[1, K_{\max}]$. In all reported runs, $K_{\text{safe}} > K_{\max} = 15$, hence $K^* = 15$.

Algorithm 1 Theory-Guided Instance-Dependent Early Stopping (TG-IES).

Require: data D, step sizes η_t, calibration epoch $t_c = 30$, cap $K_{\max}$, epochs T
Init $\hat{\theta}^{(0)}$; $\mathrm{cnt}[i] = 0$; all i active
for $t = 1$ **to** T **do**
  Forward all $i \in D$; record $\tilde{L}_i^{(t)}$
  if $t < 3$ **then**
    $\hat{S}_i \leftarrow \infty$
  else
    $\hat{S}_i \leftarrow |\tilde{L}_i^{(t)} - 2\tilde{L}_i^{(t-1)} + \tilde{L}_i^{(t-2)}|$
  end if
  if $t = t_c$ **then**
    Calibrate $\hat{\sigma}, \varepsilon_0, \delta$ and K^*
  end if
  if $t \geq t_c$ **then**
    for each $i \in D$ **do**
      if $\hat{S}_i < \delta$ **then**
        $\mathrm{cnt}[i]$ += 1; mark passive if $\mathrm{cnt}[i] \geq K^*$
      else
        $\mathrm{cnt}[i] \leftarrow 0$; mark active
      end if
    end for
  end if
  $A_t \leftarrow \{i : \text{active}\}$; **break** if $A_t = \emptyset$
  $\hat{g}^{(t)} \leftarrow \frac{1}{N} \sum_{i \in A_t} \nabla_\theta L_i(\hat{\theta}^{(t)})$; $\hat{\theta}^{(t+1)} \leftarrow \hat{\theta}^{(t)} - \eta_t \hat{g}^{(t)}$
end for
return $\hat{\theta}^{(T)}$

5.2 What happens to optimization when instances are excluded?

Question. Once instances stop receiving updates, how far can the model move from full training, and does it still descend? We follow the chain *excluded instances → omitted gradient → deviation*.

Step 1: the omitted gradient. Since A_t and $P_t = D \setminus A_t$ partition the data, the TG-IES gradient is

$$\underbrace{\hat{g}^{(t)}}_{\text{TG-IES update}} = \underbrace{g(\hat{\theta}^{(t)})}_{\text{full-data gradient}} - \underbrace{\omega_t(\hat{\theta}^{(t)})}_{\text{omitted gradient}}, \qquad \text{where} \quad \omega_t(\theta) := \frac{1}{N} \sum_{i \in P_t} g_i(\theta). \tag{7}$$

Meaning. At the TG-IES parameters, the selective update equals the full-data gradient minus the aggregate gradient of the excluded instances. We call this term the *omitted-gradient residual* ω_t. This identity is exact and does not require the excluded instances to have individually small gradients. Their gradients may be large but cancel, or individually small but add together. The stability question is therefore about their *aggregate* gradient. Thus, TG-IES can be analyzed using the established framework of inexact gradient methods (So & Zhou, 2017; Sun et al., 2017; Hallak & Levy, 2024). Note that Equation (7) simply describes the mathematical effect of excluding instances; the algorithm itself never actually computes ω_t.

Step 2: the deviation. Let $\theta^{(t)}$ (full data) and $\hat{\theta}^{(t)}$ (TG-IES) start from the same parameters at epoch t_0; these are the trajectories that the two procedures actually produce.

Theorem 5.1 (Actual-trajectory divergence)**.** *Under Assumption 3.1, suppose* $\|\omega_t(\hat{\theta}^{(t)})\| \leq \rho_t$ *for* $t = t_0, \ldots, t_0 + H - 1$. *Then*

$$\left\|\hat{\theta}^{(t_0+H)} - \theta^{(t_0+H)}\right\| \leq \sum_{s=0}^{H-1} \eta_{t_0+s}\, \rho_{t_0+s} \prod_{j=s+1}^{H-1} \left(1 + \beta \eta_{t_0+j}\right). \tag{8}$$

For a constant step size η *and* $\rho_t \leq \rho$, *the right side is at most* $\frac{\rho}{\beta}\left[(1+\beta\eta)^H - 1\right]$.

Meaning. The sum accumulates the omitted-gradient errors over the H selective steps, and the product accounts for how an earlier error is amplified as the two trajectories evolve (proof in Section D). The key stability quantity is thus the *aggregate* omitted gradient: large individual passive gradients can partly cancel, and many small ones can add up. With one passive instance, the bound reduces as $\beta \to 0$ to our earlier single-removal certificate $H\eta\varepsilon/N$ (Theorem E.1), which ignores how a gap feeds back through the gradient. **Limitation.** The factor $(1+\beta\eta)^H$ grows with H, so the bound is informative over short horizons (Remark D.2), and it assumes a bound ρ_t that we do not measure.

Step 3: descent. Staying close to full training is more than optimization needs; it suffices that each step still decreases the full loss.

Theorem 5.2 (Descent under relative omission error)**.** *Under Assumption 3.1, if* $\|\omega_t(\hat{\theta}^{(t)})\| \le \nu \|\hat{g}^{(t)}\|$ *for some* $\nu \in [0,1)$ *and* $\eta_t < 2(1-\nu)/\beta$*, then*

$$L(\hat{\theta}^{(t+1)}) \le L(\hat{\theta}^{(t)}) - \eta_t\Big(1 - \nu - \frac{\beta\eta_t}{2}\Big)\|\hat{g}^{(t)}\|^2. \tag{9}$$

Meaning. If the gradient that TG-IES leaves out is at most a fraction $\nu < 1$ of the gradient it uses, every step decreases the full training loss, under a step-size limit that tightens as ν grows; $\nu = 0$ gives the classical $\eta_t < 2/\beta$ (Section D). Whether $\nu < 1$ holds in our runs is not measured.

5.3 Connecting mastery to the residual

Question. Both theorems need a small omitted gradient, but the algorithm observes only $\hat{S}_i$. Does a small signal imply a small instance gradient? **Important distinction.** The next result is a *sufficient theoretical route* under two further idealizations. It does not show that passive instances in deep networks satisfy them; Section 8 tests this empirically.

Assumption 5.3 (Per-instance smoothness and a shared solution)**.** Each L_i is Λ_i-smooth on the neighborhood U of Assumption 3.2, and $L_i(\theta^*) = L^*$ at the *same* θ^* for every i.

Theorem 5.4 (Mastery bounds the instance gradient)**.** *Under Assumptions 3.2 and 5.3 and the regularity condition of Lemma C.3 for* $f = L_i$*,*

$$\|g_i^{(t)}\| \le \frac{r_i}{1-r_i}\sqrt{2\Lambda_i\,\Delta^2 L_i^{(t)}}, \quad \textit{so} \quad \Delta^2 L_i^{(t)} < \delta \;\Rightarrow\; \|g_i^{(t)}\| < \varepsilon_i(\delta) := \frac{r_i}{1-r_i}\sqrt{2\Lambda_i\delta}. \tag{10}$$

Meaning. If every passive instance satisfies these assumptions along the TG-IES trajectory and $\varepsilon_i(\delta) \le \bar{\varepsilon}(\delta)$, then, by the triangle inequality,

$$\|\omega_t\| \;\le\; \frac{1}{N}\sum_{i\in P_t}\|g_i\| \;<\; \frac{1}{N}\sum_{i\in P_t}\varepsilon_i(\delta) \;\le\; \frac{|P_t|}{N}\,\bar{\varepsilon}(\delta), \tag{11}$$

so $\rho_t = (|P_t|/N)\,\bar{\varepsilon}(\delta)$ links the threshold to Theorem 5.1 (under noise, use $\delta + k\sqrt{6}\,\sigma$, except with probability $2e^{-k^2/2}$; Corollary C.4). **Limitation.** r_i and Λ_i are not estimated, a shared solution is not established for deep networks, and $\varepsilon_i(\delta)$ is uninformative as $r_i \to 1$. Passing instances do have 30–33$\times$ smaller mean gradient norms (Section P), but the per-instance bound often fails (Section 8), so we state stability through the aggregate residual.

6 Experimental Setup

Methods. Four methods share identical code and optimizer settings: **Baseline** (no removal); **IES (shipped)**, the released code (smoothed signal, train/eval mismatch, $1/|A_t|$ scaling); **IES (literal)**, our faithful implementation of the stated IES algorithm (Yuan et al., 2025) (unsmoothed signal, consistent passes, $1/|A_t|$ scaling); and **TG-IES**. We also run InfoBatch (Qin et al., 2024), BLS-InfoBatch (Zhou et al., 2026), AlignPrune (Qin et al., 2026), and Selective Backprop (Jiang et al., 2019) (Section L).

Data, models, and training. We use CIFAR-10/100 (Krizhevsky et al., 2009) with ResNet-18 (He et al., 2016) (primary), VGG-16 (Simonyan & Zisserman, 2014), DenseNet-121 (Huang et al., 2017), and a CIFAR-native ViT-Tiny (Dosovitskiy et al., 2021) trained from scratch (Section Q). CNNs use SGD (momentum 0.9, weight decay 5×10^{-4}, batch size 64, 200 epochs) with exponential decay ($\eta_0 = 0.1$, $\gamma = 0.96$ per epoch) or a fixed rate ($\eta = 10^{-3}$), plus Adam (Kingma & Ba, 2014) and single-seed linear, step, and cosine schedules. The ViT uses AdamW with warmup and cosine decay.

One calibration policy. TG-IES derives δ and K^* from each run's own $\hat{\sigma}$ and ε_0, with constants fixed for all eleven settings ($p = 0.3$, $\tau = 0.02$, $\beta_{\text{test}} = 1$, the 20th percentile for ε_0, $K_{\max} = 15$, one recalibration at epoch 30). Unlike the dataset-specific multiplier in the released IES code, none is retuned, so we call this *noise-adaptive threshold calibration*, not a parameter-free method (Section N).

Metrics. Accuracy is the best test accuracy over 200 epochs (Yuan et al., 2025). Backprop savings are $100 \times (1 - \text{backpropagated instances}/(N \times \text{epochs}))$. Main settings use seeds $\{0, 1, 2\}$ and report mean $\pm$ std, with paired statistics in Section V. All runs use one RTX 5090 GPU.

7 RESULTS

Table 2: Primary setting (ResNet-18, CIFAR-10, SGD, exponential decay), 3 seeds.

Method	Acc. (%)	Saved (%)
Baseline	95.03 ± 0.13	0.03 ± 0.00
IES (shipped)	94.71 ± 0.09	45.66 ± 0.38
IES (literal)	94.85 ± 0.17	49.02 ± 0.44
InfoBatch	94.32 ± 0.12	38.91 ± 0.01
BLS-InfoBatch	94.68 ± 0.18	35.84 ± 0.09
AlignPrune	94.48 ± 0.05	20.83 ± 0.07
TG-IES (ours)	94.85 ± 0.28	$\mathbf{53.63 \pm 0.30}$

Table 3: Component ablation (primary setting, seed 0). Starred: calibrated; others keep IES (literal)'s values. Restricted TG-IES variants, not competitors.

Calibrated	Acc. (%)	Saved (%)	δ	K^*
Neither	94.92	48.59	0.001	1
δ only	94.68	**66.62**	0.0077*	1
K^* only	95.04	3.41	0.001	15*
Both (TG-IES)	94.61	53.48	0.0160*	15*

Primary setting. TG-IES achieves the highest instance-level backward savings among the compared methods while matching IES (literal)'s mean accuracy. The paired seed results have the same savings advantage in all three seeds (Section V). The cumulative trajectories in Figure 2 show that the advantage emerges mainly in the latter half of training.

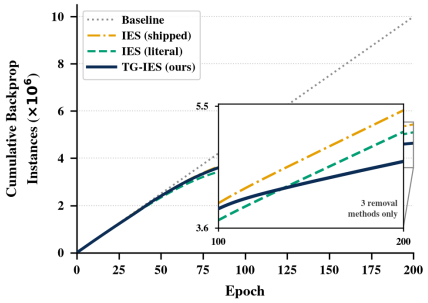

Figure 2: Cumulative backpropagation instances for ResNet-18/CIFAR-10. TG-IES uses fewer backward updates than both IES variants late in training while maintaining similar test accuracy. Inset: instance-dependent methods over epochs 100–200.

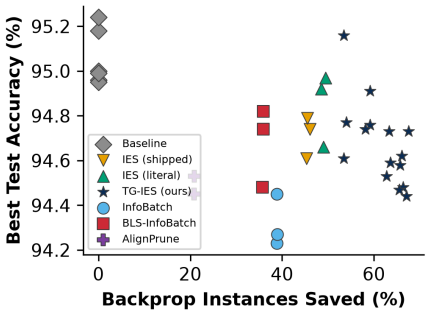

Figure 3: Best test accuracy vs. backpropagation instances saved across the primary and additional baseline comparisons. TG-IES achieves substantial savings while remaining closer to full-training accuracy than more aggressive pruning methods.

What the savings mean in compute. Every IES-family method still forwards every instance, and a backward pass costs about twice a forward pass (measured: 1.995–1.998×), so FLOP savings are $\frac{2}{3}$ of backprop savings: $\mathbf{35.7}\%$ for TG-IES, versus 30.4% and 32.7% for IES (shipped) and IES (literal). End-to-end (Section O), TG-IES is 25.5% faster than full training and uses an estimated 39% less energy but has the highest peak GPU memory (1454 vs. 1162 MiB), from its one-time recalibration.

Table 4: All three-seed settings, mean ± std ([†]primary). IES (literal), the higher-saving IES variant in every setting, is shown; all runs are in Table 11.

	Baseline	IES (literal)		TG-IES (ours)	
Setting	Acc. (%)	Acc. (%)	Saved (%)	Acc. (%)	Saved (%)
SGD with exponential learning-rate decay					
ResNet-18 / CIFAR-10[†]	95.03 ± 0.13	94.85 ± 0.17	49.02 ± 0.44	94.85 ± 0.28	53.63 ± 0.30
VGG-16 / CIFAR-10	93.30 ± 0.12	93.15 ± 0.17	41.02 ± 1.12	93.11 ± 0.14	54.43 ± 0.12
DenseNet-121 / CIFAR-10	95.36 ± 0.14	95.24 ± 0.02	51.60 ± 0.89	95.28 ± 0.09	54.53 ± 0.66
ResNet-18 / CIFAR-100	76.88 ± 0.41	76.78 ± 0.40	18.49 ± 0.43	76.01 ± 0.08	57.76 ± 0.04
No learning-rate decay (ResNet-18 / CIFAR-10)					
SGD, fixed rate	92.21 ± 0.08	91.77 ± 0.11	49.42 ± 0.17	91.91 ± 0.18	29.27 ± 0.30
Adam	92.91 ± 0.06	92.42 ± 0.10	60.09 ± 0.29	92.44 ± 0.10	30.18 ± 0.62
Vision Transformer (AdamW, warmup and cosine decay)					
ViT-Tiny / CIFAR-10	84.07 ± 0.35	83.86 ± 0.48	16.68 ± 0.32	83.71 ± 0.47	25.52 ± 0.26
ViT-Tiny / CIFAR-100	59.25 ± 0.24	58.77 ± 0.24	12.51 ± 0.21	58.56 ± 0.47	26.28 ± 0.14

Figure 3 summarizes the accuracy–compute trade-off across the primary and additional baseline comparisons. It provides a visual summary of how TG-IES compares with the IES variants and other dynamic pruning methods.

Where the gains hold, and where they do not. Table 4 shows two regimes. *With decaying learning rates, TG-IES saves more than both IES variants:* 53.6–57.8% in all four exponential-decay CNN settings, within 0.2 accuracy points of full training on CIFAR-10. The largest gain is on CIFAR-100 (57.76% vs. 18.49% for IES (literal)), which also has the largest CNN accuracy cost (0.87 points).

Without decay, IES saves more: TG-IES saves 29.27% vs. 49.42% with a fixed rate and 30.18% vs. 60.09% with Adam, at accuracy within 0.15 points of IES (literal). There the calibrated threshold is much smaller ($\delta = 0.00247$ and 0.000257 at epoch 31, vs. 0.01597 with decay), so fewer instances complete K^* passing checks and the final excluded fraction falls from 87.4% to 42.0% and 45.6% (Section K examines why; Section I adds three schedules). The advantage is thus schedule-specific.

Beyond CNNs. On ViT-Tiny, the same constants save 25.5–26.3%, 1.5–$2.1\times$ IES (literal), with a larger accuracy cost (up to 0.69 points) and the widest seed spread of any method (Section Q): the mechanism transfers, the magnitude does not.

Matched compute. At a matched budget (Section L), InfoBatch and BLS-InfoBatch, which skip both passes, lead TG-IES by 2.5–4.3 accuracy points at 20–30% of full compute; the gap narrows to 0.1–0.8 points by 40% and reverses on CIFAR-100. Selective Backprop skips 66.72% of backpropagation but loses 1.72 accuracy points. TG-IES is one point on a compute–accuracy trade-off, and below about 30% of full compute, InfoBatch-family methods reach higher accuracy in these runs.

8 What Drives the Savings, and How Stable Is Exclusion?

Component ablation. Table 3 calibrates δ and K^* separately, keeping the other at its IES (literal) value. Calibrating δ alone raises savings from 48.6% to 66.6%, exceeding full TG-IES (53.5%) at 0.24 points lower accuracy, whereas $K^* = 15$ with IES's strict $\delta = 0.001$ saves only 3.4% because noise resets the counter before 15 consecutive passes. Thus, δ drives the compute benefit, while K^* controls exclusion conservativeness; both must be calibrated to the same noise.

Patience sweep and trajectory divergence. Across $K^* \in 1, 2, 4, 8, 15, 30$ (primary setting, 3 seeds), savings fall from 66.98% to 44.13% as accuracy rises from 94.55% to 94.87% (baseline 95.08%; Section R). The parameter distance $|\hat{\theta}^{(t)} - \theta^{(t)}_{\text{baseline}}|$ to a shared-seed full-training run decreases for every K^*, from about 54 at epoch 20 to 29–37 at epoch 200, then plateaus. Larger K^* consistently gives smaller distance, as Theorem 5.1 suggests, because longer patience keeps the passive set smaller. This is consistent with a bounded aggregate residual but does not establish it; the norm-normalized distance still rises before plateauing.

Does the per-instance gradient condition hold? Every 20 epochs, a non-intrusive probe computes the true gradient of 300 random passive instances (primary setting, 3 seeds). Across 27 probes, 47.4% exceed ε_0 on average, with no downward trend; the median remains near ε_0, while a heavy tail makes the mean 2–$10\times$ higher (Section S). Thus, the per-instance floor underlying K_{safe} is too strong empirically, motivating the aggregate residual ω_t for stability. Since the probe does not measure ω_t, the condition of Theorem 5.1 remains an assumption.

9 Limitations

Conditional stability. Theorems 5.1 and 5.2 assume a bound on ω_t along the actual trajectory that the implementation neither estimates nor controls, so they identify what governs stability without certifying TG-IES globally. With about half of the sampled passive instances above the per-instance floor, K_{safe} only initializes K^*; monitoring ω_t online is future work. Theorem 5.4 also needs per-instance smoothness and a shared interpolating solution, not established for deep networks. **Scope.** The theory assumes full-batch gradient descent, unlike our SGD, Adam, and AdamW runs; without learning-rate decay, IES saves more; the evidence covers CIFAR-scale data and one small ViT; and most of the savings advantage comes from δ, not K^*.

10 Conclusion

We study instance-dependent early stopping using a second-order per-instance loss signal. Under explicit assumptions, its normalized differences remove loss-scale effects; $N = 2$ minimizes the MSE in a defined SNR regime; and the signal captures per-instance progress and curvature. We calibrate the mastery threshold from the observed noise scale and model selective updates as full-data gradient descent plus an omitted-gradient residual, yielding conditional trajectory and descent bounds. Empirically, TG-IES cuts instance-level backward passes in the main setting, with gains over IES depending on the learning-rate schedule. TG-IES is thus a noise-adaptive, theory-grounded IES method with stability guarantees conditional on the omitted-gradient residual.

IMPACT STATEMENT

This paper advances compute-efficient training; we identify no societal consequences requiring specific highlighting.

AI USE STATEMENT

In this paper we used generative AI tools (ChatGPT, Quillbot, and claude) to assist with our research and writing. The key theoretical frameworks, mathematical claims, algorithmic designs, and codebases were developed solely by the authors. We used Claude exclusively for code generation help. Additionally, we sought assistance from ChatGPT and Quillbot to enhance language, rearrange for readability, revise author-drafted parts, and format related literature. We have rigorously reviewed all AI-assisted work. All mathematical formulations and proofs were independently derived and verified step-by-step by the authors to ensure correctness and theoretical soundness. Likewise, any AI-assisted code was manually reviewed, integrated, and empirically tested. Generative AI was not used to run experiments or conduct data analysis, and all reported numbers come directly from our own experiments. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

William Bankes, George Hughes, Ilija Bogunovic, and Zi Wang. Reducr: Robust data downsampling using class priority reweighting. *Advances in Neural Information Processing Systems*, 37:82781–82810, 2024.

Brian R Bartoldson, Bhavya Kailkhura, and Davis Blalock. Compute-efficient deep learning: Algorithmic trends and opportunities. *Journal of Machine Learning Research*, 24(122):1–77, 2023.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Nadav Hallak and Kfir Yehuda Levy. A study of first-order methods with a deterministic relative-error gradient oracle. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 17313–17332, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2261–2269. IEEE, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Angela H Jiang, Daniel L-K Wong, Giulio Zhou, David G Andersen, Jeffrey Dean, Gregory R Ganger, Gauri Joshi, Michael Kaminksy, Michael Kozuch, Zachary C Lipton, et al. Accelerating deep learning by focusing on the biggest losers. *arXiv preprint arXiv:1910.00762*, 2019.

Chenhan Jin, Shengze Xu, Qingsong Wang, Fan Jia, Dingshuo Chen, and Tieyong Zeng. Orderdp: A theoretically guaranteed lossless dynamic data pruning framework. *arXiv preprint arXiv:2606.08574*, 2026.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2525–2534, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. In *International Conference on Learning Representations (ICLR)*, 2024.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 17–23 Jul 2022.

Manish Nagaraj, Deepak Ravikumar, and Kaushik Roy. Coresets from trajectories: Selecting data via correlation of loss differences. *arXiv preprint arXiv:2508.20230*, 2025.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34: 20596–20607, 2021.

Huaiyuan Qin, Muli Yang, Gabriel James Goenawan, Kai Wang, Zheng Wang, Peng Hu, Xi Peng, and Hongyuan Zhu. Beyond loss values: Robust dynamic pruning via loss trajectory alignment. *arXiv preprint arXiv:2604.07306*, 2026.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Zhou Daquan, Lei Shang, Baigui Sun, Xuansong Xie, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*, volume 2024, pp. 39723–39744, 2024.

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. In *2022 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Anthony Man-Cho So and Zirui Zhou. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *Optimization Methods and Software*, 32 (4):963–992, 2017.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:19523–19536, 2022.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

Tao Sun, Hao Jiang, Lizhi Cheng, and Wei Zhu. A convergence framework for inexact nonconvex and nonsmooth algorithms and its applications to several iterations. *arXiv preprint arXiv:1709.04072*, 2017.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Qifu Wen, Xi Zeng, Zihan Zhou, Shuaijun Liu, Mehdi Hosseinzadeh, Ningxin Su, and Reza Rawassizadeh. Grades: Significantly faster training in transformers with gradient-based early stopping. *arXiv preprint arXiv:2509.01842*, 2025.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

Suqin Yuan, Runqi Lin, Lei Feng, Bo Han, and Tongliang Liu. Instance-dependent early stopping. In *International conference on learning representations*, volume 2025, pp. 83343–83367, 2025.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *International Conference on Learning Representations (ICLR)*, 2023.

Qing Zhou, Bingxuan Zhao, Tao Yang, Hongyuan Zhang, Junyu Gao, and Qi Wang. Batch loss score for dynamic data pruning. *arXiv preprint arXiv:2604.04681*, 2026.

Haodong Zhu, Yangyang Ren, Yanjing Li, Mingbao Lin, Linlin Yang, Xuhui Liu, Xiantong Zhen, Haiguang Liu, and Baochang Zhang. Unbiased dynamic pruning for efficient group-based policy optimization. *arXiv preprint arXiv:2603.04135*, 2026.

A Extended Related Work

Why training-time compute matters. Training compute for state-of-the-art models has grown roughly $10\times$ every year since 2012 and continues to grow (Sevilla et al., 2022), with a directly proportional energy and carbon cost (Patterson et al., 2022). Bartoldson et al. (2023)'s survey of algorithmic (rather than hardware) efficiency gains frames exactly the category of method this paper contributes to: a technique reducing the number of forward/backward passes a fixed training recipe performs, orthogonal to and stackable with hardware- or precision-level efficiency gains. Section 7's FLOP-normalized accounting of TG-IES's savings is this paper's own direct, measured contribution to that question.

Static data pruning and difficulty scoring. Toneva et al. (2019) track how often an example is learned then forgotten across epochs and show unforgettable examples can be removed with little accuracy loss. Paul et al. (2021) score examples early in training with the gradient-norm-based EL2N score and prune a static subset before the rest of training. Swayamdipta et al. (2020) map examples along confidence and variability axes computed from training dynamics ("dataset cartography"). Coleman et al. (2019) select a subset using scores from a small, cheap proxy model rather than the target model itself. Sorscher et al. (2022) show, empirically and via a statistical-mechanics analysis of a toy model, that a good pruning metric can convert the usual power-law scaling of test error with dataset size into exponential scaling; Zheng et al. (2023) (CCS) show that pruning to the highest-difficulty examples alone hurts distributional coverage at high pruning rates and instead balance importance against coverage explicitly; Maharana et al. (2024) (D2 Pruning) use a message-passing formulation over a similarity graph to balance the same trade-off. All of these commit to a fixed subset before or shortly after training starts; IES and TG-IES instead make a per-instance, per-epoch decision using the current trajectory.

Dynamic and online example selection. Loshchilov & Hutter (2015) rank minibatches online by loss to bias sampling toward harder examples. Katharopoulos & Fleuret (2018) use importance sampling to reduce gradient variance, building on Needell et al. (2014). Jiang et al. (2019) (Selective-Backprop) skip backpropagation for examples with low relative loss via a probabilistic accept/reject rule, close in spirit to IES but formalized on a different argument (a probabilistic threshold on the loss distribution, versus a finite-difference criterion on the loss trajectory), with no theoretical treatment on either side. Mindermann et al. (2022) (RHO-LOSS) select points simultaneously learnable, worth learning, and not yet learnt via a holdout-model criterion; Bankes et al. (2024) (REDUCR) extend this with class-priority reweighting for robustness, orthogonal to the calibration question here. Qin et al. (2024) (InfoBatch) prune examples below the epoch-mean loss with probability r, rescaling kept low-score instances' gradients by $1/(1-r)$, annealing back to the full dataset for the final epochs; unlike IES/TG-IES it skips both forward and backward on a pruned example. Qin et al. (2026) (AlignPrune) replace loss-magnitude ranking with a Dynamic Alignment Score (correlation between an example's loss trajectory and a reference trajectory) for label-noise robustness, a different objective than mastery-based removal; Nagaraj et al. (2025) use a related trajectory-correlation signal for one-shot coreset selection against a held-out reference, with convergence guarantees for the resulting subset – the closest published criterion to IES/TG-IES's own signal, differing in using a held-out reference rather than a self-referential finite difference, and selecting once rather than every epoch. Zhou et al. (2026) (BLS) use an EMA of loss as a cheap, signal-processing alternative to TG-IES's noise-calibrated δ for the same denoising problem. Concurrently, Jin et al. (2026) (OrderDP) give a formal guarantee for dynamic pruning via an unbiased ranking surrogate with a lossless-convergence proof, a different mechanism than the conditional, residual-based stability analysis here – two different answers to the same open question rather than competitors on the same construction. Zhu et al. (2026) address an analogous unbiased-pruning problem for group-based RL policy optimization (GRPO rollout pruning for LLM reasoning), noted for completeness; its setting (sequence-level RL) is different enough that we do not attempt a direct comparison. The same question – what to spend a gradient update on – is asked one level up, over pretraining-data domains, in LLM pretraining (Xie et al., 2023), a parallel literature at a coarser granularity. Wen et al. (2025) (GradES) stops updating individual transformer projection matrices once their gradient norm falls below a threshold, a per-parameter-group decision complementary to, rather than competing with, TG-IES's per-instance decision (Section Q); we are not aware of a direct empirical comparison in either direction.

Curriculum learning. Bengio et al. (2009) order examples easy-to-hard over training; Soviany et al. (2022) survey the resulting literature. This is complementary to IES/TG-IES: curriculum learning changes when an example enters training, while IES/TG-IES changes when it leaves.

Direct predecessor. Yuan et al. (2025) is the paper this work theoretically grounds; Section 1 details the five gaps closed.

B Extended Discussion of Assumptions

Assumption B.1 (Uniform bounds on c_i)**.** There exist $0 < c_{\min} \le c_{\max} < \infty$ with $c_i \in [c_{\min}, c_{\max}]$ for all i (Assumption 3.2's amplitude).

Assumption B.2 (Bounded third derivatives)**.** There exists $M_3 < \infty$ and a neighborhood U of the GD trajectory such that for all i, all $\theta \in U$, and unit vectors u, v, w: $|\nabla^3 L_i(\theta)[u, v, w]| \le M_3$.

Remark B.3 (Why β-smoothness, and where it is used). Assumption 3.1 caps how fast the full gradient can change as parameters move. It is the only regularity assumption used by the actual-trajectory results (Theorems 5.1 and 5.2), where it turns a per-step gradient error into a multi-step parameter gap. Because smoothness alone also allows a gap to be amplified from step to step, the resulting bound carries the factor $(1 + \beta\eta)^H$; Remark D.2 discusses the additional contraction condition that would remove it.

Remark B.4 (Scope of the geometric-trajectory assumption). Assumption 3.2 is a local, near-convergence approximation, not a global description of training: loss trajectories in practice are non-monotonic and noisy, and r_i varies over time. Section 4.2's quartile analysis finds it a better approximation in the second half of training than the first.

Remark B.5 (Noise-model realism). Assumption 3.3 is an idealization: SGD revisits overlapping minibatches, correlating noise across nearby t; Assumption 3.4 may also be violated, since deep-learning gradient noise can be heavy-tailed. Under correlated or heavy-tailed noise, Theorem 4.2's variance formula becomes a lower bound and Corollary 4.4's tail bound may be optimistic. Section M tests time-independence directly and finds the measured autocorrelation small and consistent with a detrending-method artifact rather than real correlation, correcting Theorem 4.2 at $N{=}2$ by under 1.3% even under the conservative reading.

Remark B.6 (Why small rate dispersion). r_i is instance i's per-epoch decay rate; Assumption 3.5 says different instances' rates do not spread too far from a common $\bar{r}$, letting Proposition C.2 treat $\mathrm{CV}(X(2))$ as a small perturbation of $\mathrm{CV}(X(1))$ via a first-order Taylor argument valid only when the spread is genuinely small. Section G shows explicitly what breaks when it is not.

Remark B.7 (Why bounded c_i). c_i is instance i's excess-loss amplitude; bounding it away from 0 and ∞ keeps $\mathrm{CV}(\cdot)$ well-defined and the perturbation constant in Proposition C.2 finite, rather than dominated by one pathological instance.

Remark B.8 (Why bounded third derivatives). This bounds how quickly curvature itself can change, keeping Theorem 4.5's Taylor remainder genuinely smaller than the $O(\eta^2)$ curvature term it isolates; without it, "curvature is $O(\eta^2)$, drift is $O(\eta)$" would not be a controlled statement.

Remark B.9 (An unused assumption, kept for future work). A bounded gradient envelope on a window ($G := \sup_t \|g^{(t)}\| < \infty$) is a natural companion to β-smoothness for a future contraction argument bounding trajectory divergence over many steps (Remark D.2), but no theorem, corollary, or proof in this paper's current form invokes it. We name it here rather than silently omit it.

C Full Proofs

This appendix proves the results of Sections 4 and 5.3. Four statements moved out of the main text for space (Lemmas C.1 and C.3, Proposition C.2, and Corollary C.4) are given here next to their proofs. The actual-trajectory results are proved in Section D, and the single-removal certificate in Section E.

Lemma C.1 (Finite differences of a geometric trajectory)**.** *Under Assumption 3.2,* $\Delta^N(L_i^{(t)} - L^*) = c_i\, r_i^{t-N}(r_i - 1)^N$.

Proof of Lemma C.1. Induction on N. *Base case* ($N = 0$): $\Delta^0(L_i^{(t)} - L^*) = c_i r_i^t$. *Inductive step.* Assume $\Delta^{N-1}(L_i^{(t)} - L^*) = c_i r_i^{t-(N-1)}(r_i - 1)^{N-1}$. Then

$$\begin{aligned}\Delta^N(L_i^{(t)} - L^*) &= \Delta^{N-1}(L_i^{(t)} - L^*) - \Delta^{N-1}(L_i^{(t-1)} - L^*) \\ &= c_i\, r_i^{t-N+1}(r_i - 1)^{N-1} - c_i\, r_i^{t-N}(r_i - 1)^{N-1} \\ &= c_i(r_i - 1)^{N-1}\big(r_i^{t-N+1} - r_i^{t-N}\big) = c_i\, r_i^{t-N}(r_i - 1)^N,\end{aligned}$$

which is the claim for N. □

Proof of Theorem 4.1. By Lemma C.1, $\Delta^N(L_i^{(t)} - L^*) = c_i\, r_i^{t-N}(r_i - 1)^N$ and $\Delta^{N-1}(L_i^{(t)} - L^*) = c_i\, r_i^{t-N+1}(r_i - 1)^{N-1}$. Dividing,

$$\frac{\Delta^N(L_i^{(t)} - L^*)}{\Delta^{N-1}(L_i^{(t)} - L^*)} = \frac{r_i^{t-N}}{r_i^{t-N+1}} \cdot (r_i - 1) = r_i^{-1}(r_i - 1) = \frac{r_i - 1}{r_i},$$

where c_i cancels; the denominator is nonzero whenever $c_i > 0$, $r_i \in (0, 1)$, $N \geq 1$. □

Proposition C.2 (CV closeness under small rate dispersion)**.** *Under Assumptions B.1, 3.2 and 3.5, let* $X_i(N) := |c_i r_i^{t-N}(1 - r_i)^N|$ *for* $N \in \{1, 2\}$ *at fixed t, and let* $\mathrm{CV}(X) := \sqrt{\mathrm{Var}(X)}/\mathbb{E}[X]$ *across instances* $I \sim \mathrm{Unif}\{1, \ldots, N\}$*. There is a finite constant* $C(t, \bar{r}, c_{\min}, c_{\max})$ *(explicit form in Section F) such that, for all sufficiently small* δ_r, $\big|\mathrm{CV}(X(2)) - \mathrm{CV}(X(1))\big| \leq C(t, \bar{r}, c_{\min}, c_{\max}) \cdot \delta_r$.

Proof of Proposition C.2. Step 1 (pointwise perturbation). Let $f_1(r) = r^{t-1}(1 - r)$, $f_2(r) = r^{t-2}(1 - r)^2$ on $[\bar{r} - \delta_r, \bar{r} + \delta_r]$; both are smooth there, so $L_N := \sup_r |f_N'(r)| < \infty$. Writing $r_i = \bar{r} + \epsilon_i$, $|\epsilon_i| \leq \delta_r$, and $\Delta_i^{(N)} := f_N(r_i) - f_N(\bar{r})$, the mean value theorem gives $|\Delta_i^{(N)}| \leq L_N \delta_r$. With $Y_i^{(N)} := c_i f_N(\bar{r})$, $E_i^{(N)} := c_i \Delta_i^{(N)}$: $X_i(N) = Y_i^{(N)} + E_i^{(N)}$, $|E_i^{(N)}| \leq c_{\max} L_N \delta_r$.

Step 2 (mean perturbation). $|\mathbb{E}[X(N)] - \mathbb{E}[Y(N)]| \leq c_{\max} L_N \delta_r$.

Step 3 (variance perturbation). $X_i(N)^2 - Y_i(N)^2 = E_i^{(N)}(2Y_i^{(N)} + E_i^{(N)})$ is $O(\delta_r)$ uniformly, so $|\,\mathrm{Var}(X(N)) - \mathrm{Var}(Y(N))| \leq (A_N + B_N)\delta_r$ for finite A_N, B_N, hence $|\sqrt{\mathrm{Var}(X(N))} - \sqrt{\mathrm{Var}(Y(N))}| \leq D_N \delta_r$.

Step 4 (Lipschitz bound on CV). $\phi(\mu, s) = s/\mu$ is C^1 with bounded gradient Λ_N on the relevant compact region, so $|\mathrm{CV}(X(N)) - \mathrm{CV}(Y(N))| \leq \Lambda_N(c_{\max} L_N + D_N)\, \delta_r$.

Step 5. $\mathrm{CV}(Y(N)) = \sqrt{\mathrm{Var}(c)}/\mathbb{E}[c]$ is identical for $N = 1, 2$ ($f_N(\bar{r})$ cancels in the ratio); applying Step 4 to both and the triangle inequality gives the claim with $C := \max_N \Lambda_N(c_{\max} L_N + D_N)$, made explicit in Section F. □

Proof of Theorem 4.2. By linearity, $\Delta^N \tilde{L}_i^{(t)} = a + b$ with $a := \Delta^N L_i^{(t)} = c_i r_i^{t-N}(r_i - 1)^N$ (deterministic, Lemma C.1) and $b := \Delta^N \xi_{i,t} = \sum_{k=0}^N (-1)^k \binom{N}{k} \xi_{i,t-k}$. Then $\mathbb{E}[(\Delta^N \tilde{L}_i^{(t)})^2] = a^2 + 2a\mathbb{E}[b] + \mathbb{E}[b^2]$. By Assumption 3.3, $\mathbb{E}[b] = 0$; since the $\xi_{i,t-k}$ are independent with variance σ^2, $\mathbb{E}[b^2] = \sigma^2 \sum_k \binom{N}{k}^2 = \sigma^2 \binom{2N}{N}$ (Vandermonde). Combining gives Equation (3). □

Proof of Proposition 4.3. Write $\alpha_i := (1 - r_i)^2/r_i^2$, so $r_i^{2(t-N)}(r_i - 1)^{2N} = r_i^{2t}\alpha_i^N$ and $\mathrm{MSE}(N, t) = c_i^2 r_i^{2t}\alpha_i^N + \binom{2N}{N}\sigma^2$. Using $\binom{2}{1} = 2$, $\binom{4}{2} = 6$, $\binom{6}{3} = 20$, $\mathrm{MSE}(1, t) - \mathrm{MSE}(2, t) = c_i^2 r_i^{2t}\alpha_i(1 - \alpha_i) - 4\sigma^2$ and $\mathrm{MSE}(3, t) - \mathrm{MSE}(2, t) = 14\sigma^2 - c_i^2 r_i^{2t}\alpha_i^2(1 - \alpha_i)$. Hence

$$\mathrm{MSE}(2, t) \leq \mathrm{MSE}(1, t) \iff c_i^2 r_i^{2t}\alpha_i(1 - \alpha_i) \geq 4\sigma^2, \tag{12}$$

$$\mathrm{MSE}(2, t) \leq \mathrm{MSE}(3, t) \iff c_i^2 r_i^{2t}\alpha_i^2(1 - \alpha_i) \leq 14\sigma^2. \tag{13}$$

For $\alpha_i < 1$ ($r_i > \frac{1}{2}$), $\alpha_i(1 - \alpha_i) > 0$, so dividing Equations (12) and (13) by it gives the two sides of Equation (4); $N = 2$ minimizes over $\{1, 2, 3\}$ exactly when both hold. The band is non-empty since $4\alpha_i \leq 14$ for every $\alpha_i < 1$. □

Proof of Corollary 4.4. By Assumption 3.3, $\mathrm{Var}(\Delta^2\xi_{i,t}) = \mathrm{Var}(\xi_{i,t} - 2\xi_{i,t-1} + \xi_{i,t-2}) = \sigma^2(1 + 4 + 1) = 6\sigma^2$, matching Theorem 4.2 at $N = 2$. The Gaussian approximation is standard. Under Assumption 3.4, $\Delta^2\xi_{i,t}$ is a linear combination of independent σ^2-sub-Gaussian variables with squared coefficients summing to 6, hence $6\sigma^2$-sub-Gaussian; the standard tail inequality gives $2\exp(-k^2/2)$. □

Proof of Theorem 4.5. Let $\Delta\theta^{(s)} = -\eta g^{(s)}$ (full-batch). Third-order Taylor expansion with remainder (Assumption B.2):

$$L_i(\theta^{(s)} + \Delta\theta^{(s)}) = L_i^{(s)} + \langle g_i^{(s)}, \Delta\theta^{(s)}\rangle + \tfrac{1}{2}\Delta\theta^{(s)\top} H_i(\theta^{(s)})\Delta\theta^{(s)} + r_{i,s}, \quad |r_{i,s}| \le \tfrac{M_3}{6}\eta^3\|g^{(s)}\|^3.$$

Applying at $s = t-1, t-2$ and forming $\Delta^2 L_i^{(t)} = [L_i^{(t)} - L_i^{(t-1)}] - [L_i^{(t-1)} - L_i^{(t-2)}]$, the $O(\eta)$ terms combine into the drift term and the $O(\eta^2)$ terms into the curvature difference; $R_{i,t} := r_{i,t-1} - r_{i,t-2}$ and the triangle inequality give the stated remainder bound. □

Lemma C.3 (Smoothness bounds the gradient by suboptimality)**.** *If f is Λ-smooth on U and $\theta^* \in U$ satisfies $f(\theta^*) \le f(\theta - \frac{1}{\Lambda}\nabla f(\theta))$ for $\theta \in U$, then $\|\nabla f(\theta)\|^2 \le 2\Lambda(f(\theta) - f(\theta^*))$.*

Proof of Lemma C.3. By Λ-smoothness (descent lemma), $f(\theta - \frac{1}{\Lambda}\nabla f(\theta)) \le f(\theta) - \frac{1}{2\Lambda}\|\nabla f(\theta)\|^2$. Combined with $f(\theta^*) \le f(\theta - \frac{1}{\Lambda}\nabla f(\theta))$: $f(\theta^*) \le f(\theta) - \frac{1}{2\Lambda}\|\nabla f(\theta)\|^2$, which rearranges to the claim. □

Proof of Theorem 5.4. By Lemma C.3, $\|g_i^{(t)}\|^2 \le 2\Lambda_i(L_i^{(t)} - L^*) = 2\Lambda_i c_i r_i^t$ (Assumption 3.2). By Lemma C.1 at $N = 2$, $\Delta^2 L_i^{(t)} = c_i r_i^{t-2}(1 - r_i)^2$, so $c_i r_i^t = (r_i/(1 - r_i))^2 \Delta^2 L_i^{(t)}$. Substituting gives $\|g_i^{(t)}\|^2 \le 2\Lambda_i(r_i/(1 - r_i))^2 \Delta^2 L_i^{(t)}$, and taking square roots gives Equation (10). □

Corollary C.4 (Probabilistic bridge under observation noise)**.** *Under the assumptions of Theorem 5.4 and Assumption 3.4, for $k > 0$ and $\delta' := \delta + k\sqrt{6}\,\sigma$: $\Pr\left(\hat{S}_i < \delta \text{ and } \|g_i^{(t)}\| \ge \varepsilon_i(\delta')\right) \le 2\exp(-k^2/2)$.*

Remark C.5 (Scope of the bridge). Two caveats apply. First, $\varepsilon_i(\delta) \propto r_i/(1 - r_i)$ grows without bound as $r_i \to 1$, so the bound is uninformative for near-stationary, as opposed to near-converged, instances. Second, Corollary C.4 bounds a *joint*, not a conditional, probability, since $\hat{S}_i$ and $\Delta^2\xi_{i,t}$ share the same noise.

Proof of Corollary C.4. $\Delta^2 L_i^{(t)} = \hat{S}_i \mp \Delta^2\xi_{i,t}$, so $\hat{S}_i < \delta$ with $|\Delta^2\xi_{i,t}| \le k\sqrt{6}\sigma$ implies $\Delta^2 L_i^{(t)} < \delta'$, hence $\|g_i^{(t)}\| < \varepsilon_i(\delta')$ by Theorem 5.4. So the event in the statement is contained in $\{|\Delta^2\xi_{i,t}| > k\sqrt{6}\sigma\}$, bounded by Corollary 4.4 at $2\exp(-k^2/2)$. □

D Actual-Trajectory Analysis: Proofs and Discussion

This appendix proves Theorems 5.1 and 5.2, recovers the per-instance multi-step bound of our earlier analysis as a special case, and discusses what would remove the exponential factor. Throughout, $\theta^{(t)}$ is full-data gradient descent, $\hat{\theta}^{(t)}$ is the TG-IES trajectory (Equation (6)), A_t and P_t are the algorithm's own evolving active and passive sets, and ω_t is the omitted-gradient residual of Equation (7). As in the main text, the analysis idealizes each epoch as one full-batch step.

Proof of Theorem 5.1. Let $e_t := \hat{\theta}^{(t)} - \theta^{(t)}$, with $e_{t_0} = 0$. By Equation (7),

$$e_{t+1} = e_t - \eta_t\big(g(\hat{\theta}^{(t)}) - g(\theta^{(t)})\big) + \eta_t\,\omega_t(\hat{\theta}^{(t)}).$$

By Assumption 3.1, $\|g(\hat{\theta}^{(t)}) - g(\theta^{(t)})\| \le \beta\|e_t\|$, so the triangle inequality and the residual bound give

$$\|e_{t+1}\| \le (1 + \beta\eta_t)\|e_t\| + \eta_t\rho_t. \tag{14}$$

Unrolling Equation (14) from t_0 to $t_0 + H$ with $\|e_{t_0}\| = 0$ gives Equation (8), where an empty product equals 1. For constant η and $\rho_t \le \rho$, the right side of Equation (8) is at most $\eta\rho\sum_{s=0}^{H-1}(1 + \beta\eta)^{H-1-s} = \eta\rho\,\frac{(1+\beta\eta)^H - 1}{\beta\eta}$, which is the constant-step bound stated in Theorem 5.1. No convexity is used. □

Proof of Theorem 5.2. Write $g_t := g(\hat{\theta}^{(t)})$ and $\omega_t := \omega_t(\hat{\theta}^{(t)})$, so that $g_t = \hat{g}^{(t)} + \omega_t$ by Equation (7). The descent lemma for the β-smooth loss L, applied to the step $\hat{\theta}^{(t+1)} = \hat{\theta}^{(t)} - \eta_t \hat{g}^{(t)}$, gives

$$L(\hat{\theta}^{(t+1)}) \le L(\hat{\theta}^{(t)}) - \eta_t \langle g_t, \hat{g}^{(t)} \rangle + \frac{\beta \eta_t^2}{2} \|\hat{g}^{(t)}\|^2.$$

By Cauchy–Schwarz and the relative-error condition, $\langle g_t, \hat{g}^{(t)} \rangle = \|\hat{g}^{(t)}\|^2 + \langle \omega_t, \hat{g}^{(t)} \rangle \ge \|\hat{g}^{(t)}\|^2 - \|\omega_t\| \, \|\hat{g}^{(t)}\| \ge (1-\nu)\|\hat{g}^{(t)}\|^2$. Substituting gives Equation (9). The coefficient $1 - \nu - \beta\eta_t/2$ is positive exactly when $\eta_t < 2(1-\nu)/\beta$. □

The per-instance route as a special case. An earlier version of this analysis bounded the same trajectory gap by assuming a gradient floor for every passive instance. That result follows from Theorem 5.1 through the triangle inequality in Equation (11).

Corollary D.1 (Per-instance multi-step bound)**.** *Under Assumption 3.1 with a constant step size η, suppose every passive instance satisfies $\|g_i(\hat{\theta}^{(t)})\| < \varepsilon_i$ at every step along the actual TG-IES trajectory. Let $\bar{\varepsilon}_t := \frac{1}{N} \sum_{i \in P_t} \varepsilon_i$ and $\varepsilon_{\max} := \max_i \varepsilon_i$. Then, with $t_0 = 0$,*

$$\|\hat{\theta}^{(t)} - \theta^{(t)}\| \;\le\; \eta \sum_{s=0}^{t-1} (1+\eta\beta)^{t-1-s} \, \bar{\varepsilon}_s \;\le\; \frac{\varepsilon_{\max}}{\beta} \Big[(1+\eta\beta)^t - 1 \Big]. \tag{15}$$

Proof. $\|\omega_s(\hat{\theta}^{(s)})\| \le \frac{1}{N} \sum_{i \in P_s} \|g_i(\hat{\theta}^{(s)})\| < \bar{\varepsilon}_s$, so Equation (8) applies with $\rho_s = \bar{\varepsilon}_s$. The second inequality uses $\bar{\varepsilon}_s \le (|P_s|/N)\, \varepsilon_{\max} \le \varepsilon_{\max}$. □

The aggregate condition of Theorem 5.1 is weaker than the per-instance condition of Corollary D.1. It constrains only the norm of the *summed* passive gradient, so it can hold even when many individual passive gradients exceed a per-instance floor, for example when those gradients partly cancel. The reactivation probe (Section S) shows that the per-instance condition fails for about half of the sampled passive instances at any snapshot, which is why the main text states the aggregate version. The probe does not measure ω_t, however, so the aggregate condition remains an assumption.

Remark D.2 (What would remove the exponential factor). The factor $(1+\beta\eta)^H$ in Theorem 5.1 arises because smoothness alone lets the gradient-descent map expand parameter gaps. The absolute divergence measured in Section 8 instead *plateaus*, which is qualitatively more consistent with a contractive map. If L were additionally μ-strongly convex on a convex region containing both trajectories and $\eta < 2/\beta$, the map $\theta \mapsto \theta - \eta\, g(\theta)$ would be a contraction with factor $\kappa = \max(|1-\eta\mu|, |1-\eta\beta|) < 1$. Replacing $(1+\beta\eta)$ by κ in Equation (14) gives the horizon-independent bound $\|\hat{\theta}^{(t)} - \theta^{(t)}\| \le \eta\rho/(1-\kappa)$, which matches the observed plateau qualitatively. Local contraction of this kind is plausible in restricted regimes (late training, over-parameterized models, near a broad minimum), but we have not verified it along the trajectories of our architectures, and we flag it as the natural next step.

E THE SINGLE-REMOVAL REFERENCE CERTIFICATE

Our earlier analysis bounded the effect of removing a single instance under a shared-reference idealization. We keep it here because it yields the closed-form horizon K_{safe} used to initialize K^*. It is no longer the main stability result: Theorem 5.1 is stated for the actual trajectories and contains this certificate as its first-order ($\beta \to 0$) limit for a single passive instance.

Theorem E.1 (Single-removal certificate, shared-reference trajectory)**.** *Suppose instance i satisfies $\|g_i^{(t)}\| < \varepsilon$ for all $t \in [T-K, T]$. Consider two runs that share the same reference trajectory $\{\theta^{(t)}\}_{t=T-K}^{T}$, both use Equation* (6) *evaluated on it, and differ only in whether $i \in A_t$. Then $\big\|\theta_{\text{full}}^{(T)} - \theta_{\text{rem}}^{(T)}\big\| \le K\eta\varepsilon/N$. If the test loss is additionally β_{test}-Lipschitz, then $\big|L_{\text{test}}(\theta_{\text{full}}^{(T)}) - L_{\text{test}}(\theta_{\text{rem}}^{(T)})\big| \le \beta_{\text{test}} K\eta\varepsilon/N$.*

Corollary E.2 (Conditional reference horizon)**.** *Given a tolerance $\tau > 0$ on the test-loss deviation, any exclusion window $K \le K_{\text{safe}} := \lfloor N\tau/(\beta_{\text{test}}\, \eta\, \varepsilon) \rfloor$ keeps the deviation in Theorem E.1 at most τ. The floor, not the ceiling, is required, since rounding up would violate the bound.*

Remark E.3 (Why this certificate is conditional and local). Theorem E.1 compares two runs evaluated on the *same* reference trajectory. In an actual run, the two trajectories separate from the moment i is removed, and the separation feeds back through the gradient. Theorem 5.1 accounts for this feedback through the factor $(1+\beta\eta)$, and Theorem E.1 is its $\beta \to 0$ limit for a single passive instance. The certificate is also local: it covers one removal, not the repeated and overlapping removals of Algorithm 1.

Remark E.4 (K_{safe} is not the algorithm's patience). K_{safe} bounds a *closed* exclusion window counted backward from a fixed endpoint, whereas the patience K^* of Algorithm 1 governs an *open-ended* exclusion that begins after K^* consecutive passing epochs. Nothing here shows that initializing K^* from K_{safe} keeps that open-ended exclusion within the budget of Corollary E.2. In addition, Theorem E.1 requires $\|g_i^{(t)}\| < \varepsilon$, whereas the algorithm thresholds the noisy signal $\hat{S}_i < \delta$; Theorem 5.4 connects the two only under Assumption 5.3. We therefore call K_{safe} a conditional reference horizon and K^* a theory-informed persistence rule.

Proof of Theorem E.1. At each $t \in [T-K, T]$, both runs use Equation (6) on the same reference trajectory, so their update directions differ only by the term $\frac{1}{N} g_i^{(t)}$. Hence each step adds at most $\frac{\eta}{N}\|g_i^{(t)}\| \le \frac{\eta\varepsilon}{N}$ to the gap, and summing over K steps (triangle inequality) gives $\|\theta_{\text{full}}^{(T)} - \theta_{\text{rem}}^{(T)}\| \le K\eta\varepsilon/N$. The test-loss bound follows from β_{test}-Lipschitzness. □

Proof of Corollary E.2. $\beta_{\text{test}} K\eta\varepsilon/N \le \tau \iff K \le N\tau/(\beta_{\text{test}}\eta\varepsilon)$. The largest integer satisfying this is $K_{\text{safe}} = \lfloor N\tau/(\beta_{\text{test}}\eta\varepsilon)\rfloor$; a ceiling would round up past the constraint. □

F Explicit Constants for Proposition C.2

Recall from Assumption B.1 that $\mathrm{Var}(c) \le (c_{\max} - c_{\min})^2/4 =: V_c$. One concrete choice of the constant in Proposition C.2 is

$$C(t, \bar{r}, c_{\min}, c_{\max}) := \max_{N \in \{1,2\}} \left(\frac{c_{\max} L_N}{c_{\min} f_N(\bar{r})} + \frac{c_{\max} L_N}{\sqrt{V_c}\, c_{\min} f_N(\bar{r}) + 10^{-12}} \right),$$

where $f_1(r) = r^{t-1}(1-r)$, $f_2(r) = r^{t-2}(1-r)^2$, $L_N = \sup_{r \in [\bar{r}-\delta_r, \bar{r}+\delta_r]} |f_N'(r)|$. The 10^{-12} guards against division by zero when all c_i are equal ($V_c = 0$).

G Counterexample: Global CV Monotonicity Can Fail

A global claim $\mathrm{CV}(\Delta^2 L) \le \mathrm{CV}(\Delta^1 L)$ (without Assumption 3.5) can fail under wide rate dispersion. Take $c_1 = c_2 = 1$, $L^* = 0$, $r_1 = 0.01$, $r_2 = 0.99$, $t = 10$:

$$\Delta^1(L_1^{(10)}) \approx -10^{-18},\ \Delta^1(L_2^{(10)}) \approx -0.0091, \qquad \Delta^2(L_1^{(10)}) \approx 10^{-16},\ \Delta^2(L_2^{(10)}) \approx 9.2{\times}10^{-5}.$$

The cross-instance magnitude ratio is order 10^{16} at $N{=}1$ and order 10^{11} at $N{=}2$: dispersion (CV across the two instances) *increases*, not decreases, from $N{=}1$ to $N{=}2$ here, confirming Assumption 3.5 is load-bearing, not a technicality.

H Extended Practical Implementation Notes

Recalibration schedule. δ and K^* are recalibrated once, at epoch 30, then held fixed, rather than repeating every Δ epochs: repeating was found counterproductive, since a shrinking noise floor $\hat{\sigma}$ re-tightens δ over time and every recalibration resets in-progress K^*-window counts, together strangling savings. This is an empirical finding, not a theorem (Section 9).

Gradient-norm floor estimate. ε_0 is a low percentile (default 20th) of a cheap last-layer proxy $\|\mathrm{softmax}(z_i) - \mathrm{onehot}(y_i)\| \cdot \|\mathrm{feat}_i\|$ over a subsample of the active set (capped at 8192 instances), cutting recalibration cost roughly $6\times$ versus a full backward pass without materially changing the percentile estimate.

Consistent forward-pass discipline. Following Yuan et al. (2025)'s stated Algorithm 1 rather than the shipped implementation, all instances (active and passive) are forwarded in the same `train()`

mode every epoch (passive under `no_grad`), avoiding a train/eval batch-normalization mismatch (Ioffe & Szegedy, 2015).

A rounding-direction correction, with no effect on reported results. Corollary E.2 rounds K_{safe} down; the implementation generating every run in this paper used a ceiling instead, caught only while finalizing this version. Checking every run's logged K^* trajectory: in every case K^* reaches $\text{max_kstar} = 15$ immediately upon recalibration, i.e., the uncapped value already exceeds the clamp before rounding is applied, so ceiling vs. floor changes nothing here. The implementation now uses the floor consistent with Corollary E.2. The same check shows that the operational patience is $K^* = K_{\max} = 15$ in every reported TG-IES run, which is why the main text describes K^* as a theory-informed persistence rule rather than a derived safety horizon.

I Full Zero-Shot Transfer Table

Table 5: TG-IES, every distinct (dataset, architecture, optimizer/schedule) cell in this paper, same four fixed calibration constants throughout. Mean ± std over seeds where $n>1$; asterisked rows are single-seed by design (Section K's matched-epoch schedule comparison).

Cell	n	Acc. (%)	Saved (%)
CIFAR-10 / ResNet-18 / SGD-exp (primary)	3	94.85 ± 0.28	53.63 ± 0.30
CIFAR-10 / ResNet-18 / SGD-fixed	3	91.91 ± 0.18	29.27 ± 0.30
CIFAR-10 / ResNet-18 / SGD-linear	1	94.85^*	32.05^*
CIFAR-10 / ResNet-18 / SGD-step	1	94.44^*	43.52^*
CIFAR-10 / ResNet-18 / SGD-cosine	1	94.73^*	23.15^*
CIFAR-10 / ResNet-18 / Adam	3	92.44 ± 0.10	30.18 ± 0.62
CIFAR-10 / VGG-16 / SGD-exp	3	93.11 ± 0.14	54.43 ± 0.12
CIFAR-10 / DenseNet-121 / SGD-exp	3	95.28 ± 0.09	54.53 ± 0.66
CIFAR-100 / ResNet-18 / SGD-exp	3	76.01 ± 0.08	57.76 ± 0.04
CIFAR-10 / ViT-Tiny / AdamW-wc	3	83.71 ± 0.47	25.52 ± 0.26
CIFAR-100 / ViT-Tiny / AdamW-wc	3	58.56 ± 0.47	26.28 ± 0.14

The savings figure varies substantially across cells (23–58%), as it should: schedule-, dataset-, and architecture-dependent mechanisms (Sections J, K and Q) directly cause this variation. What does not vary is the calibration procedure: this table was assembled with no per-cell constant lookup, because there is none to look up.

J Second Optimizer: Adam

Table 6: ResNet-18, CIFAR-10, Adam, mean ± std, 3 seeds.

Method	Test Acc. (%)	Backprop Saved (%)
Baseline	92.91 ± 0.06	0.03 ± 0.00
IES (shipped)	92.61 ± 0.10	56.02 ± 0.22
IES (literal)	92.42 ± 0.10	$\mathbf{60.09 \pm 0.29}$
TG-IES (ours)	92.44 ± 0.10	30.18 ± 0.62

Adam (Kingma & Ba, 2014) is a second, more pronounced instance of the fixed-LR pattern of Section 7: both IES variants save more (56.0–60.1%) than TG-IES's 30.2%, at accuracy within 0.02–0.17 points of TG-IES's (Section V: the accuracy gap vs. IES (literal) is not distinguishable from zero at $n = 3$, while the saved-% deficit is consistent in sign across all 3 seeds). Adam and fixed-LR SGD share the structural property distinguishing them from every cell where TG-IES leads: neither has the exponential schedule's large, steadily-decaying learning rate. We report this as an open empirical pattern, not an explained one; Section K proposes and checks a candidate mechanism against the logged record.

K Toward a Mechanism: The Drift Term Under Non-Decaying Learning Rates

Theorem 4.5 decomposes $\Delta^2 L_i^{(t)} = \eta_t D_i^{(t)} + O(\eta_t^2) + R_{i,t}$, $D_i^{(t)} := \langle g_i^{(t-2)}, g^{(t-2)} \rangle - \langle g_i^{(t-1)}, g^{(t-1)} \rangle$. Under the exponential schedule, $\eta_t \to 0$ drives the whole drift term toward zero regardless of $D_i^{(t)}$'s own decay; under a non-decaying schedule (fixed-LR SGD, or Adam's lack of

scalar-η decay), only $D_i^{(t)} \to 0$ on its own can shrink it, which Assumption 3.2's geometric picture gives no particular reason to expect on any fixed timetable. This predicts δ should calibrate smaller wherever η is smaller, and fewer instances should reach K^*=15 consecutive low-signal epochs. Both hold in the logged data: at epoch 31, δ recalibrates to 0.01597 (exponential, $\eta_{31}\approx$0.0282) vs. 0.00247 (fixed-LR, 6.5× smaller) and 0.000257 (Adam, 62× smaller); by training's end the exponential run excludes 87.4% of instances vs. 42.0% (fixed-LR) and 45.6% (Adam), matching the backprop-saved gap.

Direct instrumentation of $D_i^{(t)}$. A diagnostic run computing $g^{(t)}$, every $L_i^{(t)}$, and $\langle g_i^{(t)}, g^{(t)} \rangle$ for 300 fixed instances, baseline training, all three schedules, *complicates* the "drift dominates" reading: $\eta_t D_i^{(t)}$ correlates only weakly with the observed $\Delta^2 L_i^{(t)}$ (Pearson $r = 0.34, 0.37, 0.23$ for exponential/fixed/Adam), and its magnitude does not dominate the combined curvature-plus-remainder term (ratio 0.69×, 0.32×, and effectively 0× respectively) – the opposite of a literal "drift dominates" reading, and it weakens further in the last training quarter (Figure 4). We read this as refining, not refuting, Theorem 4.5: the $O(\eta)$-vs-$O(\eta^2)$ ordering is asymptotic as $\eta \to 0$, and at this network's actual step sizes (η up to 0.1) the curvature term is evidently not negligible in absolute terms. The qualitative argument above only requires drift to *scale* with η_t, which this does not contradict; but $\Delta^2 L_i^{(t)}$ being *well-approximated by* $\eta_t D_i^{(t)}$ alone is not supported, and Section 4.3's framing is corrected accordingly.

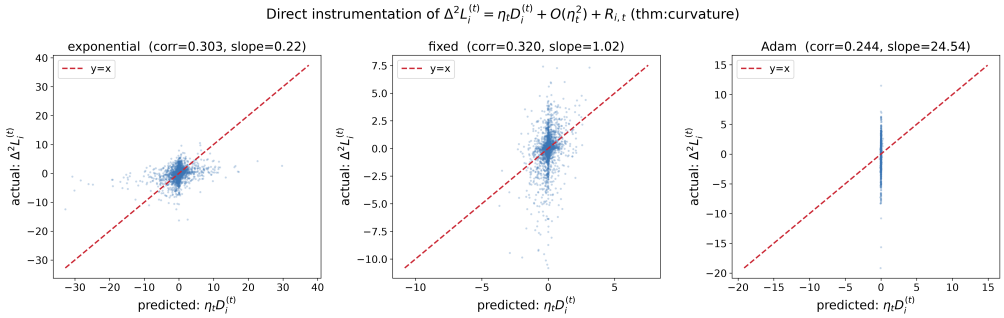

Figure 4: Direct instrumentation of $\Delta^2 L_i^{(t)} = \eta_t D_i^{(t)} + O(\eta_t^2) + R_{i,t}$, fixed 300-instance subset, baseline training, all valid epochs, one panel per schedule. Dashed line: $y = x$. The drift term correlates weakly with the observed signal and does not dominate the residual under any schedule.

L Full New-Baselines Comparison and Fixed-Compute-Budget Analysis

We implement and run four baselines not previously benchmarked against IES: InfoBatch (Qin et al., 2024) (r=0.5 soft-pruning below epoch-mean loss, $1/(1-r)$ gradient rescaling, annealing to the full dataset for the final 12.5% of epochs); BLS-InfoBatch (Zhou et al., 2026) (same schedule, EMA-smoothed score, α=0.7); AlignPrune (Qin et al., 2026) (Dynamic Alignment Score against the population-mean trajectory, since no clean-labeled reference subset exists here); and Selective Backprop (Jiang et al., 2019) (percentile-rank accept probability $p^{\beta_{\text{sb}}-1}$, β_{sb}=3, on the persistent last-observed-loss score, no gradient rescaling and no annealing back to the full dataset, since SB is an explicitly biased sampler by design rather than InfoBatch's unbiased correction). We do not compare against DPPO (Zhu et al., 2026) (a different, RL setting).

On VGG-16 (seed 0) and CIFAR-100 (seed 0), the same ranking as the primary cell holds among the first three: TG-IES leads at 54.4%/57.8% saved against InfoBatch's 38.9%/35.2%, BLS-InfoBatch's 36.4%/30.9%, and AlignPrune's 22.7%/22.5%, all within 1–2 accuracy points. AlignPrune saves the least of these methods consistently, expected rather than a weakness of our implementation: its Dynamic Alignment Score identifies samples *misaligned* with the population trend (a label-noise signal), not samples converged *with* it (a mastery signal); on clean CIFAR labels most of what it would need to prune to match TG-IES are exactly the well-aligned instances its score is built to keep.

Selective Backprop: the other extreme of the trade-off. On the primary cell (3 seeds), SB saves $66.72 \pm 0.01\%$ of backpropagation instances – more than every other method in this paper, TG-IES included (53.63%) – but at $93.31 \pm 0.11\%$ accuracy, a 1.72-point cost relative to baseline

(95.03%), the largest accuracy cost of any method compared anywhere in this paper (next largest: TG-IES's own 0.87 points on CIFAR-100, Table 4). SB's savings are also the most seed-stable of any method measured (±0.01 points, matching InfoBatch's ±0.01). This is not a failure of our implementation: SB's probabilistic accept/reject rule has no noise-calibrated threshold and no persistence rule (Section 2), so nothing in it stops pruning before accuracy degrades. Read together with Table 2, SB and TG-IES mark the two ends of a Pareto-like frontier on this cell: the most raw compute saved for the most accuracy given up, versus a small, bounded accuracy cost for a smaller but still leading (among methods with any comparable accuracy) compute saving.

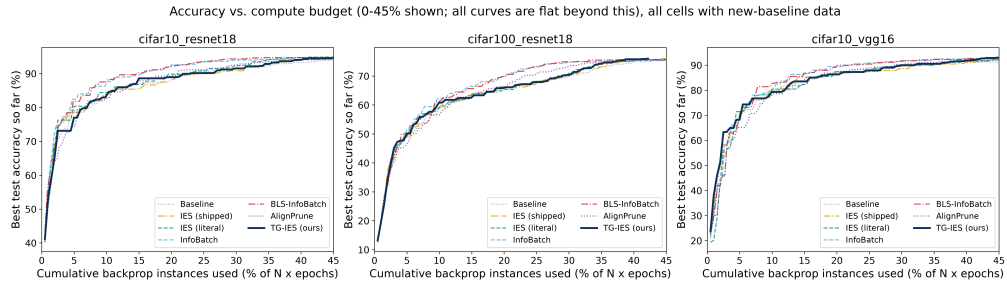

Figure 5: Best-so-far test accuracy vs. cumulative compute budget (% of $N\times$total-epochs), all three cells with new-baseline data. A method's curve stops at its own final compute point.

Accuracy under a fixed compute budget. At a common compute budget rather than each method's own stopping point, the result is a genuine trade-off, consistent across all three cells: at low budgets (20–30%), InfoBatch and BLS-InfoBatch lead TG-IES by 2.5–4.3 points, since they skip both forward and backward on a pruned instance and so get more model updates per unit of measured compute early in training. This narrows to 0.1–0.8 points by 40% and reverses on CIFAR-100 (+0.12 points for TG-IES). The practical implication is budget-dependent: a practitioner constrained to 20–30% of full compute would currently do better with InfoBatch-family methods; the own-stopping-point comparison of Table 2 describes a different, later operating regime. We report both framings rather than hide the trade-off.

M Empirical Validation of the Noise-Independence Assumption

Assumption 3.3 assumes $\xi_{i,t}$ independent across t; we test this directly. Detrending each instance's epoch-101–200 loss trajectory (ResNet-18/CIFAR-10, seed 0) with an OLS linear fit and pooling residuals across instances estimates lag-k autocorrelation directly (Figure 6).

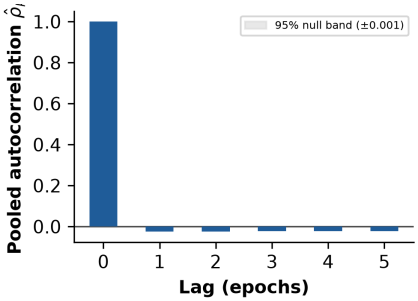

Figure 6: Pooled lag-k autocorrelation of detrended per-instance loss residuals, epochs 101–200, ResNet-18/CIFAR-10 seed 0 ($n\approx$1.9M pairs/lag). Shaded band: 95% null interval for zero correlation.

An earlier pass detrended with a width-5 moving average and measured $\hat{\rho}_1{=}-0.298$, $\hat{\rho}_2{=}-0.348$, which looked like strong evidence against Assumption 3.3; we verified by direct calculation that pure white noise under that same filter produces $\rho_1{=}-0.30$, $\rho_2{=}-0.35$ *exactly* – the entire signal was the detrending filter's own artifact. Redone with OLS linear detrending (whose own simulated white-noise artifact is $\hat{\rho}_1{\approx}-0.021$, $\hat{\rho}_2{\approx}-0.019$), the real data gives $\hat{\rho}_1{=}-0.025$, $\hat{\rho}_2{=}-0.023$ – statistically distinguishable from zero at $n\approx$1.9M pairs but small and close to the detrending method's own bias, evidence *for*, not against, Assumption 3.3 once the method's bias is accounted for. Taking the measured values at face value regardless, the correction to Theorem 4.2 at $N{=}2$ is a factor of 6.15 against the assumed 6.0 – a 2.5% correction, requiring δ to grow under 1.3% to keep the same nominal false-exclusion rate. This does not materially change any result in Section 7.

N GRADIENT-NORM-FLOOR PERCENTILE SENSITIVITY

The calibration policy of Section 6 fixes the percentile behind ε_0 (and hence the reference horizon for K^*) at the 20th throughout; we sweep $\{5, 10, 20, 35, 50, 80\}$ on ResNet-18 and VGG-16 (primary recipe, seed 0), holding everything else fixed.

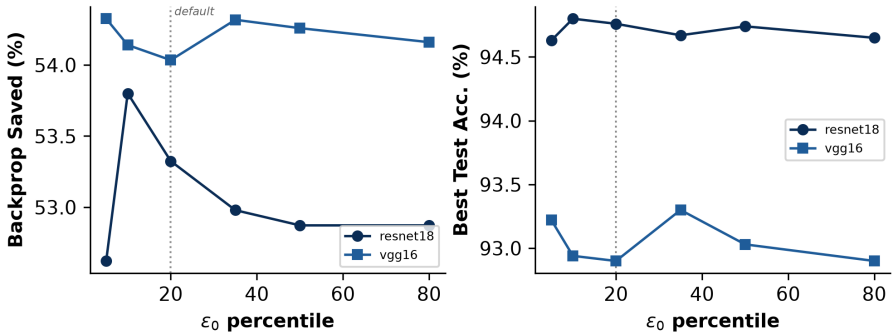

Figure 7: Backprop saved (left) and best test accuracy (right) vs. gradient-norm-floor percentile, ResNet-18 and VGG-16, CIFAR-10, seed 0. Dotted line: this paper's fixed default (20th percentile).

Both curves are essentially flat across a 16× range: ResNet-18 savings vary by 1.18 points, accuracy by 0.17; VGG-16 by 0.30 and 0.40 points respectively. Neither curve is monotonic, itself informative: there is no directional trend to correct for by tuning further. The gap between architectures at any fixed percentile is larger than the gap the percentile choice induces within either, consistent with the percentile being a fixed, low-sensitivity policy input rather than one needing per-architecture retuning within this range. This shows that the current value is not fragile; it does not show that a derived value would not be preferable in principle (Remark E.4).

O MEASURED WALL-CLOCK TIME, PEAK MEMORY, AND ESTIMATED ENERGY

Table 7: Measured efficiency, primary setting, single RTX 5090. Wall-clock: mean over 3 seeds from the real 200-epoch runs. Peak Mem/Util/Energy: short instrumented runs, seed 0; Energy is an estimate. Lower is better throughout.

Method	Wall-clk (min)	Peak Mem (MiB)	Util (%)	Energy (Wh)
Baseline	15.66	1162	67.7	98.2
IES (shipped)	12.86	1216	62.8	78.1
IES (literal)	12.26	1184	59.0	70.1
TG-IES	**11.66**	1454	**48.9**	**59.9**

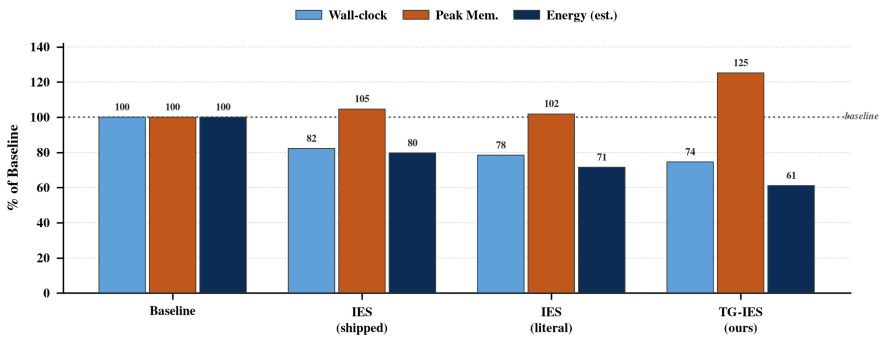

Figure 8: Wall-clock time, peak GPU memory, and estimated energy, each as % of no-removal baseline. Wall-clock and energy fall furthest for TG-IES; peak memory is the one metric that rises above baseline, only for TG-IES.

Wall-clock is the per-epoch time already logged by every 200-epoch run; peak memory, mean utilization, and mean power were collected separately via out-of-process `nvidia-smi` polling on fresh 8-epoch instrumented repetitions; estimated energy multiplies each method's measured

mean power by its real 200-epoch wall-clock. TG-IES is fastest and most estimated-energy-efficient (25.5% faster, $\sim 39\%$ lower energy than baseline) but has the *highest* peak memory (1454 MiB, $\sim 25\%$ above baseline), attributable specifically to its one-time recalibration pass: estimating the gradient-norm-floor percentile forwards a subsample in batches of 256 (four times the training batch size), and PyTorch's caching allocator does not release the resulting larger reservation for the rest of the run. This is a genuine memory/compute trade-off specific to TG-IES's calibration mechanism.

P Additional Figures for Finding 2 and the Mastery-to-Gradient Bridge

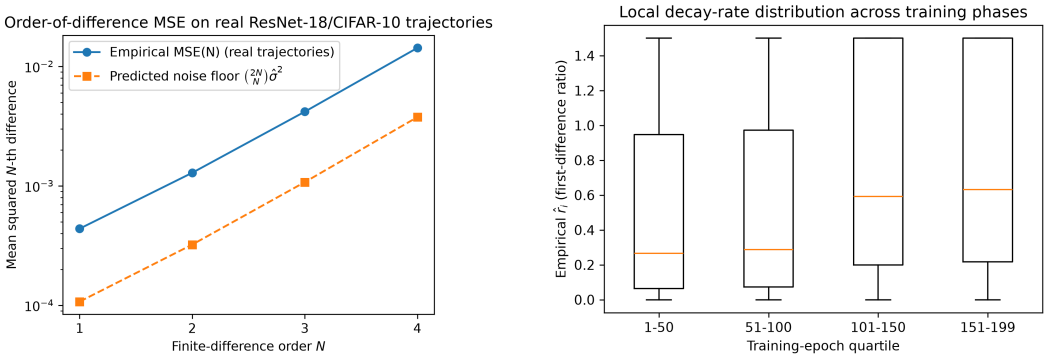

Figure 9: Left: empirical MSE(N) vs. predicted noise floor, naive population-average test (all instances, epochs 101–200). This pooled test does not isolate Proposition 4.3's SNR band and finds $N{=}1$ lowest; the band-restricted test (main text), which evaluates each instance at the epoch its signal first drops below the calibrated $\delta{=}0.01597$, finds $N{=}2$ lowest (7.5×10^{-5} pooled over 46,779 instances), as predicted. Right: empirical $\hat{r}_i$ pooled per training-epoch quartile; rate rises and spread narrows in the second half of training (mean $\hat{r}$ from 0.53 to 0.76; CV from 1.05 to 0.75), so Assumption 3.2 fits late training better.

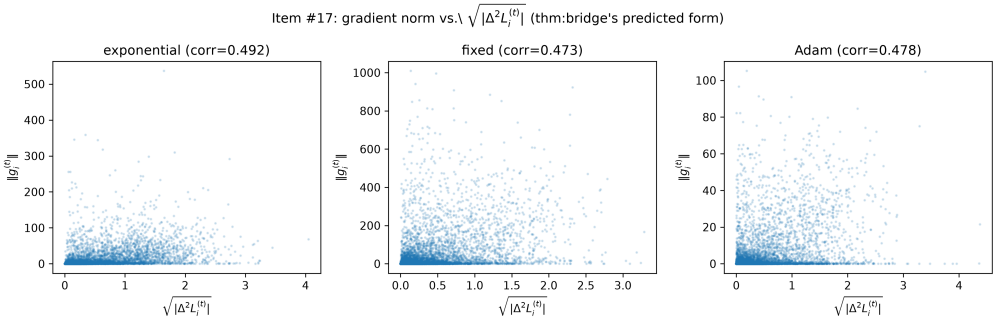

Figure 10: Per-instance gradient norm $\|g_i^{(t)}\|$ vs. $\sqrt{|\Delta^2 L_i^{(t)}|}$, all three LR schedules, primary architecture/dataset (300-instance subset). Theorem 5.4 predicts a positive, roughly linear relationship; correlations shown per panel.

Q ViT-Tiny: Full Results and Caveats

Every architecture in the main matrix is convolutional; here we test whether TG-IES's calibration – derived under no architectural assumption beyond a per-instance loss and gradient – transfers to a Vision Transformer (Dosovitskiy et al., 2021), trained from scratch at CIFAR scale with the same four fixed calibration constants as every convolutional cell. We use a CIFAR-native ViT-Tiny: patch-4 on the 32×32 input (64 tokens plus CLS), 6 pre-norm blocks, 192-dim, 3 heads, 2.69M parameters – not the ImageNet-oriented patch-16 configuration, whose tokenization would yield only 4 tokens at this resolution. AdamW (lr 10^{-3}, weight decay 0.05), 10-epoch linear warmup then cosine decay over 200 epochs; we validated the baseline alone before running the full comparison.

Four findings. The savings advantage transfers and grows with difficulty: 1.53× IES (literal) on CIFAR-10, 2.10× on CIFAR-100, tight seed variance on savings specifically. Noise-adaptive threshold calibration, not the fixed-δ methods' hand-tuned constant, is the mechanism: δ rescales 186× (CIFAR-10) and 333× (CIFAR-100) with no constant retuned by hand. Absolute savings are

Table 8: ViT-Tiny (CIFAR-native, patch-4), CIFAR-10 and CIFAR-100, AdamW warmup-cosine, mean ± std, 3 seeds.

	CIFAR-10		CIFAR-100	
Method	Acc. (%)	Saved (%)	Acc. (%)	Saved (%)
Baseline	84.07 ± 0.35	0.03 ± 0.00	59.25 ± 0.24	0.03 ± 0.00
IES (shipped)	83.92 ± 0.18	13.13 ± 0.21	59.19 ± 0.22	8.38 ± 0.22
IES (literal)	83.86 ± 0.48	16.68 ± 0.32	58.77 ± 0.24	12.51 ± 0.21
TG-IES (ours)	83.71 ± 0.47	$\mathbf{25.52 \pm 0.26}$	58.56 ± 0.47	$\mathbf{26.28 \pm 0.14}$

much lower than CNNs ($\sim 26\%$ vs. $\sim 54.5\%$ DenseNet-121). There is a small accuracy cost on transformers unlike CNNs (0.36–0.69 points, widest seed spread of any method), suggesting instance removal destabilizes a data-starved, from-scratch ViT more than a CNN.

Caveats. K^* was pinned at the configured ceiling (15) in every ViT run: Corollary E.2's closed form asked for more patience than allowed, so these numbers reflect the theory clamped by configuration; lifting the cap is untested. Savings arrive late (exclusions ramp steeply only in the second half of training). δ recalibrates once, at epoch 30 as in every CNN cell; whether this timing is also right for a transformer's different loss dynamics is untested. Absolute accuracy is well below the CNN cells, expected for a from-scratch transformer on 50k images with crop-and-flip augmentation only.

R K^*-SWEEP: FULL TRAJECTORY FIGURE AND DISCUSSION

Table 9: K^*-sweep, primary cell, mean ± std, 3 seeds, δ noise-calibrated. Matched baseline: $95.08 \pm 0.14\%$ acc.

K^* (fixed)	Acc. (%)	Saved (%)
1	94.55 ± 0.16	$\mathbf{66.98 \pm 0.62}$
2	94.56 ± 0.08	65.75 ± 0.27
4	94.62 ± 0.10	63.24 ± 0.44
8	94.80 ± 0.09	58.85 ± 0.57
15	94.80 ± 0.09	53.38 ± 0.58
30	$\mathbf{94.87 \pm 0.09}$	44.13 ± 0.40

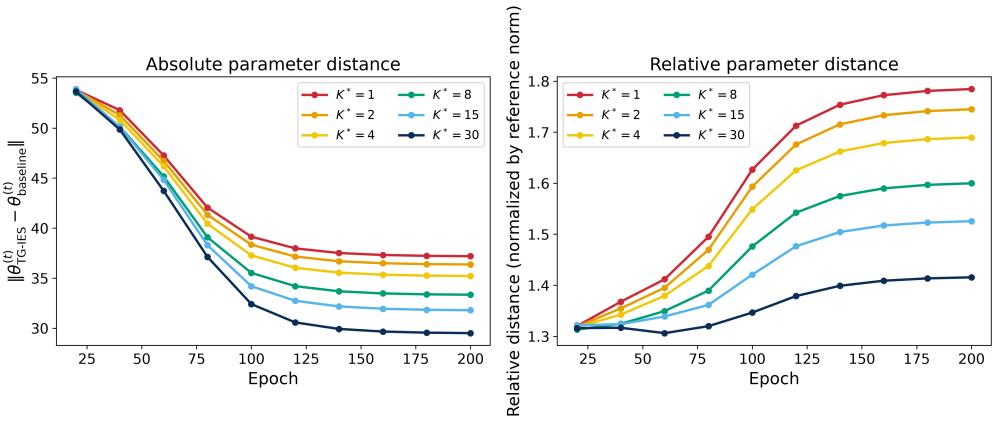

Figure 11: Parameter-space distance between TG-IES's actual trajectory and the shared-seed baseline reference vs. epoch, one curve per fixed K^*, 3-seed mean. Left: absolute distance. Right: normalized by the baseline's own parameter norm at that epoch.

Absolute parameter distance from baseline *decreases* over training for every K^* tested, from ≈ 54 at epoch 20 to 29–37 by epoch 200, plateauing over the last third of training. At every snapshot epoch, larger K^* gives a smaller distance, with no crossings. This is the direction suggested by Theorem 5.1: longer patience delays exclusion and keeps the passive set, and plausibly the residual ω_t, smaller. The *normalized* distance (right panel) instead rises and plateaus higher, because the reference parameter norm itself shrinks faster (weight decay) than the absolute gap does. We report both panels, since this experiment does not settle which normalization is the right one for assessing multi-step deviation. We read the plateau as empirical behavior consistent with a bounded aggregate residual. It is not a

validation of Theorem 5.1, whose residual bound is not measured here, and the plateau is also not predicted by that theorem's smoothness-only bound, which grows with the horizon (Remark D.2).

S Reactivation Probe: Does the Excluded Set Actually Satisfy the Gradient-Floor Hypothesis?

The single-removal certificate (Theorem E.1), the reference horizon used to initialize K^* (Corollary E.2), and the per-instance multi-step bound (Corollary D.1) all rest on a per-instance hypothesis: every currently excluded instance i satisfies $\|g_i^{(t)}\| < \varepsilon_0$, where ε_0 is the cheap last-layer proxy percentile estimated once at epoch 30 (Section H). The other checks in this paper (Section 5.3's empirical check, Section P) use the same cheap proxy or a correlational test, not the *true* full-model gradient norm on the actual excluded set during an otherwise unmodified run. We test the hypothesis directly: `--reactivation_probe_every 20` adds, every 20 epochs, an individual batch-size-1 forward+backward pass on a random subsample of 300 currently-passive instances (noise-calibrated δ, primary cell, 3 seeds, tagged `_reactprobe`), computing each one's true $\|g_i^{(t)}\|$ and checking it against the run's own calibrated ε_0. This instrumentation is purely diagnostic: it does not feed back into the algorithm's actual removal or reactivation decisions, so the probed runs' accuracy and savings ($94.67 \pm 0.11\%$ acc., $53.66 \pm 0.10\%$ saved) match the unprobed primary-cell TG-IES numbers (Table 2) closely, confirming the probe itself does not perturb training.

Table 10: Reactivation probe, primary cell, 3-seed mean of `frac_exceed_eps0` (fraction of a 300-instance random subsample of the passive set whose true gradient norm exceeds the run's calibrated ε_0) at each probed epoch.

Epoch	40	60	80	100	120	140	160	180	200
Frac. $> \varepsilon_0$	0.52	0.48	0.39	0.46	0.48	0.46	0.47	0.50	0.50

Finding: roughly half the excluded set exceeds the calibrated floor at every check, with no decreasing trend. Pooling all 9 probed epochs across 3 seeds (27 (epoch, seed) points, 300 probed instances each), the mean fraction of the passive set exceeding ε_0 is 47.4%, ranging 34–59% epoch-to-epoch with no consistent decrease over training (Table 10) – if anything, the fraction is lowest at epoch 80 and drifts back up to $\approx 50\%$ by epoch 200, the opposite of what a hypothesis that becomes *more* valid as training progresses would predict. This contradicts a literal reading of the per-instance hypothesis: at any given check, roughly half of the instances the algorithm currently excludes have a true gradient norm above the floor that the per-instance analysis assumes. The picture is not uniformly bad, however: each seed's *median* probed gradient norm (0.025–0.029 across seeds and epochs) sits close to that seed's own ε_0 (0.027–0.030), while the *mean* probed gradient norm is 2–$10\times$ higher (0.07–0.29) – a heavy right tail, not a uniformly shifted distribution. So the calibrated ε_0 is a reasonable estimate of a *typical* (median) excluded instance's gradient floor, but a systematically loose one for the population as a whole, since roughly half the excluded set sits above it at any moment.

Why this motivates the aggregate formulation. On the same primary cell, the trajectory-divergence measurement of Section 8 finds that the absolute parameter gap from the full-data baseline *plateaus* rather than growing, although the per-instance violation rate never approaches zero. The inexact-gradient view of Section 5.2 locates the quantity that matters for this comparison: by Theorem 5.1, multi-step deviation depends on the passive set only through the aggregate residual $\omega_t = \frac{1}{N}\sum_{i \in P_t} g_i$, not through any individual $\|g_i\|$. The per-instance floor is thus a sufficient but unnecessarily strong way to control ω_t (Corollary D.1), and this probe shows that it is too strong to be treated as an empirical property of the passive set. The probe does *not*, however, measure ω_t, so it does not establish the aggregate condition required by Theorem 5.1. We therefore keep that condition as an explicit assumption (Section 9) rather than claiming it has been verified, and we treat the plateau as behavior consistent with a bounded aggregate residual, not as proof of one.

T Extended Future-Work List

(1) *Scale.* All experiments are CIFAR-scale; TinyImageNet or ImageNet-100 would test whether calibration (in particular the gradient-norm-floor proxy behind K^*) transfers to higher resolution and more classes. (2) *Architecture family, partly closed.* Section Q adds one CIFAR-native ViT-Tiny;

a larger transformer (ViT-Small, DeiT-Tiny) or one at a resolution where patch-16 tokenization is meaningful remains untried, as does raising `--max_kstar`. (3) ***Optimizer/schedule sensitivity.*** Sections 7 and J show TG-IES trailing simpler IES variants on settings without the exponential schedule; Section K's candidate mechanism is only partly confirmed by direct instrumentation, and a controlled ablation calibrating δ, K^* under a matched noise/gradient-norm profile across schedules remains unrun. The bridge's qualitative direction ($\|g_i^{(t)}\|$ vs. $\sqrt{|\Delta^2 L_i^{(t)}|}$) is now confirmed across all three schedules; only the *exact* bound $\varepsilon_i(\delta)$, needing per-instance r_i, Λ_i, remains open. (4) RHO-LOSS (Mindermann et al., 2022) and RL-style dynamic pruning (Zhu et al., 2026) are not compared. (5) ***The aggregate residual.*** Theorems 5.1 and 5.2 are conditional on a bound for ω_t, which the current implementation neither measures nor controls. Measuring $\|\omega_t\|$ during training, bounding it from observable quantities, and turning such a bound into a residual-aware reactivation rule would each require instrumentation beyond the present study. A contraction-based version of Theorem 5.1 that removes the $(1+\beta\eta)^H$ factor (Remark D.2) is the corresponding theoretical step. We view these as the most important remaining gaps, ahead of (1)–(4).

U Additional Experimental Details

A reproducibility correction. An earlier version reported TG-IES collapsing to 1.8% savings on VGG-16. We traced this to a stale configuration that repeated the recalibration step. Re-run with the one-time recalibration used everywhere else, VGG-16 gives the largest saving of any method on that setting (54.43%, Table 4). The fixed-LR ResNet-18 gap to IES (literal) (29.27% vs. 49.42%) is genuine and persists after re-running. The failure mode that the stale configuration had suggested, a patience rule incompatible with the threshold, is real, and it is demonstrated by the controlled ablation of Section 8 rather than by this corrected data point.

Full per-run results. Table 11 lists every IES-family run underlying the main tables and Sections J and Q (96 runs total). Two rows (ResNet-18/F and VGG-16/E, TG-IES, seed 0) supersede values from an earlier version, traced to the reproducibility issue above; stale originals are retained in the released code's `results/` directory with a `_STALE_preverify` suffix for audit purposes.

Table 11: All 96 IES-family runs. E = SGD exponential LR, F = SGD fixed LR, C100 = CIFAR-100, ViT/C10 and ViT/C100 = ViT-Tiny on CIFAR-10/CIFAR-100. Color on TG-IES rows compares against the other methods for that same seed.

Setting	Method	Seed	Acc. (%)	Saved (%)
ResNet-18/E	Baseline	0	94.96	0.03
ResNet-18/E	Baseline	1	95.18	0.03
ResNet-18/E	Baseline	2	94.95	0.03
ResNet-18/E	IES (shipped)	0	94.61	45.36
ResNet-18/E	IES (shipped)	1	94.74	46.09
ResNet-18/E	IES (shipped)	2	94.79	45.53
ResNet-18/E	IES (literal)	0	94.92	48.59
ResNet-18/E	IES (literal)	1	94.66	49.00
ResNet-18/E	IES (literal)	2	94.97	49.47
ResNet-18/E	TG-IES	0	94.61	53.48
ResNet-18/E	TG-IES	1	95.16	53.44
ResNet-18/E	TG-IES	2	94.77	53.98
ResNet-18/F	Baseline	0	92.28	0.03
ResNet-18/F	Baseline	1	92.21	0.03
ResNet-18/F	Baseline	2	92.13	0.03
ResNet-18/F	IES (shipped)	0	91.78	44.71
ResNet-18/F	IES (shipped)	1	91.69	44.55
ResNet-18/F	IES (shipped)	2	91.61	44.23
ResNet-18/F	IES (literal)	0	91.77	49.60
ResNet-18/F	IES (literal)	1	91.66	49.38
ResNet-18/F	IES (literal)	2	91.87	49.27
ResNet-18/F	TG-IES	0	91.71	29.57
ResNet-18/F	TG-IES	1	91.96	29.27
ResNet-18/F	TG-IES	2	92.07	28.98
VGG-16/E	Baseline	0	93.25	0.03
VGG-16/E	Baseline	1	93.21	0.03
VGG-16/E	Baseline	2	93.44	0.03
VGG-16/E	IES (shipped)	0	93.33	38.67

continued on next page

Table 11 – continued from previous page

Setting	Method	Seed	Acc. (%)	Saved (%)
VGG-16/E	IES (shipped)	1	93.27	37.46
VGG-16/E	IES (shipped)	2	93.34	36.99
VGG-16/E	IES (literal)	0	93.04	41.85
VGG-16/E	IES (literal)	1	93.35	41.45
VGG-16/E	IES (literal)	2	93.07	39.74
VGG-16/E	TG-IES	0	92.96	54.36
VGG-16/E	TG-IES	1	93.15	54.56
VGG-16/E	TG-IES	2	93.23	54.37
DenseNet-121/E	Baseline	0	95.37	0.03
DenseNet-121/E	Baseline	1	95.50	0.03
DenseNet-121/E	Baseline	2	95.22	0.03
DenseNet-121/E	IES (shipped)	0	95.37	48.19
DenseNet-121/E	IES (shipped)	1	95.47	49.79
DenseNet-121/E	IES (shipped)	2	95.25	47.83
DenseNet-121/E	IES (literal)	0	95.26	51.96
DenseNet-121/E	IES (literal)	1	95.24	52.27
DenseNet-121/E	IES (literal)	2	95.23	50.59
DenseNet-121/E	TG-IES	0	95.24	54.73
DenseNet-121/E	TG-IES	1	95.22	55.06
DenseNet-121/E	TG-IES	2	95.38	53.79
ResNet-18/Adam	Baseline	0	92.98	0.03
ResNet-18/Adam	Baseline	1	92.88	0.03
ResNet-18/Adam	Baseline	2	92.86	0.03
ResNet-18/Adam	IES (shipped)	0	92.67	56.04
ResNet-18/Adam	IES (shipped)	1	92.50	55.80
ResNet-18/Adam	IES (shipped)	2	92.66	56.23
ResNet-18/Adam	IES (literal)	0	92.49	59.80
ResNet-18/Adam	IES (literal)	1	92.47	60.39
ResNet-18/Adam	IES (literal)	2	92.30	60.10
ResNet-18/Adam	TG-IES	0	92.33	30.86
ResNet-18/Adam	TG-IES	1	92.49	29.64
ResNet-18/Adam	TG-IES	2	92.51	30.04
ResNet-18/C100	Baseline	0	76.46	0.03
ResNet-18/C100	Baseline	1	77.27	0.03
ResNet-18/C100	Baseline	2	76.91	0.03
ResNet-18/C100	IES (shipped)	0	76.64	11.74
ResNet-18/C100	IES (shipped)	1	76.69	11.99
ResNet-18/C100	IES (shipped)	2	77.29	12.19
ResNet-18/C100	IES (literal)	0	76.33	18.18
ResNet-18/C100	IES (literal)	1	76.92	18.32
ResNet-18/C100	IES (literal)	2	77.09	18.98
ResNet-18/C100	TG-IES	0	75.97	57.81
ResNet-18/C100	TG-IES	1	75.96	57.73
ResNet-18/C100	TG-IES	2	76.11	57.75
ViT-Tiny/C10	Baseline	0	84.25	0.03
ViT-Tiny/C10	Baseline	1	84.30	0.03
ViT-Tiny/C10	Baseline	2	83.67	0.03
ViT-Tiny/C10	IES (shipped)	0	84.13	13.37
ViT-Tiny/C10	IES (shipped)	1	83.83	12.97
ViT-Tiny/C10	IES (shipped)	2	83.81	13.06
ViT-Tiny/C10	IES (literal)	0	84.32	17.05
ViT-Tiny/C10	IES (literal)	1	83.90	16.48
ViT-Tiny/C10	IES (literal)	2	83.37	16.52
ViT-Tiny/C10	TG-IES	0	84.26	25.80
ViT-Tiny/C10	TG-IES	1	83.47	25.30
ViT-Tiny/C10	TG-IES	2	83.41	25.45
ViT-Tiny/C100	Baseline	0	59.19	0.03
ViT-Tiny/C100	Baseline	1	59.51	0.03
ViT-Tiny/C100	Baseline	2	59.05	0.03
ViT-Tiny/C100	IES (shipped)	0	58.96	8.20
ViT-Tiny/C100	IES (shipped)	1	59.40	8.62
ViT-Tiny/C100	IES (shipped)	2	59.20	8.33
ViT-Tiny/C100	IES (literal)	0	58.54	12.28
ViT-Tiny/C100	IES (literal)	1	59.02	12.70
ViT-Tiny/C100	IES (literal)	2	58.75	12.55
ViT-Tiny/C100	TG-IES	0	58.97	26.23
ViT-Tiny/C100	TG-IES	1	58.05	26.43
ViT-Tiny/C100	TG-IES	2	58.66	26.17

Hardware and reproducibility. All runs used a single NVIDIA RTX 5090 GPU; Section O reports precise wall-clock, memory, utilization, and estimated energy. All four methods are implemented in a single training entry point taking `--method {baseline,ies_shipped,ies_alg1,tgies}` so data pipeline, model, and optimizer code are shared exactly across methods; a single script drives the full experimental matrix and supports resuming any interrupted run.

V Statistical Analysis of Seed-Level Comparisons

Every $\pm$ figure elsewhere in this paper is a sample standard deviation over 3 seeds, not a significance test. This section reports one directly. `setup_seed(seed)` is called once per run, before model initialization and data loading, so all four methods sharing a seed share the same initial weights and data-shuffling order – a genuine paired design, analyzed via paired per-seed differences rather than treating (method, seed) runs as independent. Eight settings have 3 seeds: the primary cell, both secondary CNN cells, Adam, CIFAR-100, DenseNet-121, and both ViT-Tiny cells; the three LR-sweep cells of Section K remain single-seed by design.

For each 3-seed setting we compare TG-IES against each other method on best test accuracy, backprop-saved %, and FLOP-saved %, reporting: the paired mean difference and its sample standard deviation; a 95% CI via the exact t-distribution at 2 degrees of freedom ($t_{0.975,2} = 4.303$); Cohen's d_z (paired effect size); and an exact two-sided sign test (floor $p = 2/8 = 0.25$ at $n = 3$, reached whenever all seeds agree in sign, regardless of effect size).

Table 12: Paired per-seed statistics, TG-IES vs. each other method, every 3-seed setting ($n = 3$). CI: 95%, $t(\mathrm{df}{=}2)$. d_z: Cohen's paired effect size. p_{sign}: exact two-sided sign test (floor 0.25 at $n = 3$).

Metric	Comparison (TG-IES −)	Mean Δ	95% CI	d_z	p_{sign}
Primary: ResNet-18, CIFAR-10, SGD (exponential LR)					
Acc. (%)	Baseline	−0.18	[−0.59, 0.23]	−1.11	0.250
Acc. (%)	IES (shipped)	0.13	[−0.48, 0.75]	0.54	1.000
Acc. (%)	IES (literal)	0.00	[−1.09, 1.09]	−0.01	1.000
Backprop Saved (%)	Baseline	53.60	[52.86, 54.34]	179.2	0.250
Backprop Saved (%)	IES (shipped)	7.97	[6.57, 9.37]	14.2	0.250
Backprop Saved (%)	IES (literal)	4.61	[4.00, 5.22]	18.8	0.250
FLOP Saved (%)	Baseline	35.71	[35.22, 36.21]	179.2	0.250
FLOP Saved (%)	IES (shipped)	5.31	[4.38, 6.24]	14.2	0.250
FLOP Saved (%)	IES (literal)	3.07	[2.67, 3.48]	18.8	0.250
Secondary: ResNet-18, CIFAR-10, SGD (fixed LR)					
Acc. (%)	Baseline	−0.29	[−0.93, 0.35]	−1.14	0.250
Acc. (%)	IES (shipped)	0.22	[−0.45, 0.89]	0.82	1.000
Acc. (%)	IES (literal)	0.15	[−0.31, 0.61]	0.79	1.000
Backprop Saved (%)	Baseline	29.24	[28.51, 29.98]	98.8	0.250
Backprop Saved (%)	IES (shipped)	−15.23	[−15.42, −15.03]	−194.1	0.250
Backprop Saved (%)	IES (literal)	−20.14	[−20.48, −19.80]	−148.1	0.250
FLOP Saved (%)	Baseline	19.49	[18.99, 19.97]	98.8	0.250
FLOP Saved (%)	IES (shipped)	−10.14	[−10.27, −10.02]	−194.1	0.250
FLOP Saved (%)	IES (literal)	−13.42	[−13.65, −13.20]	−148.1	0.250
Secondary: VGG-16, CIFAR-10, SGD (exponential LR)					
Acc. (%)	Baseline	−0.19	[−0.48, 0.10]	−1.60	0.250
Acc. (%)	IES (shipped)	−0.20	[−0.57, 0.17]	−1.36	0.250
Acc. (%)	IES (literal)	−0.04	[−0.50, 0.42]	−0.22	1.000
Backprop Saved (%)	Baseline	54.40	[54.11, 54.69]	470.5	0.250
Backprop Saved (%)	IES (shipped)	16.72	[14.46, 18.98]	18.4	0.250
Backprop Saved (%)	IES (literal)	13.42	[10.69, 16.13]	12.3	0.250
FLOP Saved (%)	Baseline	36.23	[36.04, 36.43]	470.5	0.250
FLOP Saved (%)	IES (shipped)	11.14	[9.63, 12.64]	18.4	0.250
FLOP Saved (%)	IES (literal)	8.94	[7.12, 10.75]	12.3	0.250
Adam: ResNet-18, CIFAR-10					
Acc. (%)	Baseline	−0.46	[−0.87, −0.06]	−2.84	0.250
Acc. (%)	IES (shipped)	−0.17	[−0.58, 0.24]	−1.01	0.250
Acc. (%)	IES (literal)	0.02	[−0.44, 0.48]	0.13	1.000

continued on next page

Table 12 – continued from previous page

Metric	Comparison (TG-IES −)	Mean Δ	95% CI	d_z	p_{sign}
Backprop Saved (%)	Baseline	30.15	[28.61, 31.69]	48.6	0.250
Backprop Saved (%)	IES (shipped)	−25.85	[−27.28, −24.41]	−44.7	0.250
Backprop Saved (%)	IES (literal)	−29.91	[−32.18, −27.65]	−32.8	0.250
FLOP Saved (%)	Baseline	20.09	[19.06, 21.11]	48.6	0.250
FLOP Saved (%)	IES (shipped)	−17.22	[−18.18, −16.26]	−44.7	0.250
FLOP Saved (%)	IES (literal)	−19.93	[−21.44, −18.42]	−32.8	0.250
CIFAR-100: ResNet-18, SGD (exponential LR)					
Acc. (%)	Baseline	−0.87	[−1.90, 0.16]	−2.09	0.250
Acc. (%)	IES (shipped)	−0.86	[−1.55, −0.17]	−3.09	0.250
Acc. (%)	IES (literal)	−0.77	[−1.64, 0.11]	−2.18	0.250
Backprop Saved (%)	Baseline	57.73	[57.64, 57.83]	1522.5	0.250
Backprop Saved (%)	IES (shipped)	45.79	[45.15, 46.43]	178.2	0.250
Backprop Saved (%)	IES (literal)	39.27	[38.16, 40.37]	88.3	0.250
FLOP Saved (%)	Baseline	38.47	[38.40, 38.53]	1522.5	0.250
FLOP Saved (%)	IES (shipped)	30.51	[30.08, 30.94]	178.2	0.250
FLOP Saved (%)	IES (literal)	26.17	[25.43, 26.90]	88.3	0.250
DenseNet-121: CIFAR-10, SGD (exponential LR)					
Acc. (%)	Baseline	−0.08	[−0.64, 0.47]	−0.37	1.000
Acc. (%)	IES (shipped)	−0.08	[−0.57, 0.40]	−0.43	1.000
Acc. (%)	IES (literal)	0.04	[−0.21, 0.28]	0.37	1.000
Backprop Saved (%)	Baseline	54.49	[52.85, 56.13]	82.5	0.250
Backprop Saved (%)	IES (shipped)	5.92	[4.36, 7.49]	9.4	0.250
Backprop Saved (%)	IES (literal)	2.92	[2.32, 3.53]	12.0	0.250
FLOP Saved (%)	Baseline	36.32	[35.22, 37.41]	82.5	0.250
FLOP Saved (%)	IES (shipped)	3.95	[2.90, 4.99]	9.4	0.250
FLOP Saved (%)	IES (literal)	1.95	[1.54, 2.35]	12.0	0.250
ViT-Tiny: CIFAR-10, AdamW (warmup-cosine)					
Acc. (%)	Baseline	−0.36	[−1.43, 0.71]	−0.84	1.000
Acc. (%)	IES (shipped)	−0.21	[−0.94, 0.52]	−0.71	1.000
Acc. (%)	IES (literal)	−0.15	[−0.77, 0.47]	−0.61	1.000
Backprop Saved (%)	Baseline	25.49	[24.84, 26.13]	98.5	0.250
Backprop Saved (%)	IES (shipped)	12.38	[12.25, 12.52]	228.0	0.250
Backprop Saved (%)	IES (literal)	8.83	[8.62, 9.05]	101.1	0.250
FLOP Saved (%)	Baseline	17.06	[16.63, 17.49]	98.5	0.250
FLOP Saved (%)	IES (shipped)	8.29	[8.20, 8.38]	228.0	0.250
FLOP Saved (%)	IES (literal)	5.91	[5.77, 6.06]	101.1	0.250
ViT-Tiny: CIFAR-100, AdamW (warmup-cosine)					
Acc. (%)	Baseline	−0.69	[−2.36, 0.98]	−1.03	0.250
Acc. (%)	IES (shipped)	−0.63	[−2.33, 1.07]	−0.92	1.000
Acc. (%)	IES (literal)	−0.21	[−1.97, 1.55]	−0.30	1.000
Backprop Saved (%)	Baseline	26.25	[25.90, 26.59]	191.0	0.250
Backprop Saved (%)	IES (shipped)	17.89	[17.60, 18.18]	154.5	0.250
Backprop Saved (%)	IES (literal)	13.77	[13.36, 14.17]	85.0	0.250
FLOP Saved (%)	Baseline	17.57	[17.34, 17.79]	191.0	0.250
FLOP Saved (%)	IES (shipped)	11.98	[11.78, 12.17]	154.5	0.250
FLOP Saved (%)	IES (literal)	9.21	[8.94, 9.48]	85.0	0.250

What this does and does not support. On the primary cell, TG-IES's backprop- and FLOP-saved advantage over every other method is in the same direction in all 3 seeds for every comparison (hence $p_{\text{sign}} = 0.25$, the floor), with CIs entirely on one side of zero; large d_z values (12–470) reflect small seed-to-seed variability relative to a large, consistent savings effect. The same pattern holds on VGG-16, CIFAR-100 ($d_z = 1522.5$, the largest in this table), DenseNet-121, and both ViT-Tiny cells. On ResNet-18(fixed-LR) and Adam, the sign flips as Sections 7 and J report qualitatively, with CIs again excluding zero entirely. Accuracy differences are small (|mean Δ| ≤ 0.87 points) with CIs mostly straddling zero; the exceptions (Adam vs. Baseline, CIFAR-100 vs. IES shipped) are real, if small, effects at this sample size. A CI containing zero is not evidence the true difference *is* zero, and a large d_z with $p_{\text{sign}} = 0.25$ is not conventionally "significant" – it is the most a 3-seed exact sign test can report regardless of effect size; we do not describe any result here as statistically significant at a threshold this design cannot reach.