

A Gradient Analysis During Training

In this section, we analyze the behavior of gradients throughout training. We fine-tune a LLaMA2-7B model [Touvron et al., 2023] on 10000 randomly selected samples from Tulu V2 [Ivison et al., 2023] for 2 epochs, saving model checkpoints every 10 steps.

For each checkpoint—including the initial and final models—we compute the gradients of 1000 held-out samples from Tulu V2, as well as samples from the target dataset BBH [Suzgun et al., 2022], and project them into an 8192-dimensional space using random Rademacher matrices, following the efficient GPU implementation of [Park et al., 2023], also adopted in [Xia et al., 2024]. For each dataset, we compute the average gradient cosine similarity across checkpoints. As shown in Figure 4, while the gradient directions can change substantially in the early steps, they stabilize quickly during training. This observation justifies the use of a short warm-up phase as both necessary and sufficient. Similar plots for GSM8k [Cobbe et al., 2021] and SQuAD [Rajpurkar et al., 2016] are provided later in the Appendix (Figure 9).

Additionally, for each dataset and checkpoint, we measure the Pearson product-moment correlation between gradient norms and the number of label tokens per sample. As shown in Figure 5, we observe a consistent negative correlation, which supports our decision to normalize gradients prior to distillation.

B Linear Model Study

In this section, we show that a regularization term can be effective in robustifying Objective 6 to small changes in the model parameters θ , when the model is linear and the loss is quadratic.

For a fixed $\epsilon > 0$, define a new objective as below:

$$w^* = \arg \min_w \max_{\|\delta\| \leq \epsilon} f(w; \theta + \delta), \quad (13)$$

minimizing the maximum value of f around a point θ in a neighborhood of radius ϵ . This ensures the weights are stable as long as θ is in this neighborhood. To solve for w , we employ Lemma B.1 below:

Lemma B.1. Assume $\mathcal{L}(\theta; D, w) = \sum_{i=1}^{|D|} w_i (\langle x_i^D, \theta \rangle - y_i^D)^2$, where $D = \{(x_1^D, y_1^D), (x_2^D, y_2^D), \dots, (x_{|D|}^D, y_{|D|}^D)\}$ is a dataset. For datasets S and T , let $\mathbf{H}_T = \nabla_{\theta}^2 \mathcal{L}(\theta; T, \mathbb{1})$, $\mathbf{g}_T = \nabla_{\theta} \mathcal{L}(\theta; T, \mathbb{1})$, $\mathbf{H}_w = \nabla_{\theta}^2 \mathcal{L}(\theta; S, w)$, and $\mathbf{g}_w = \nabla_{\theta} \mathcal{L}(\theta; S, w)$. Define \mathbf{a}_w and \mathbf{B}_w as below:

$$\mathbf{a}_w = -\mathbf{H}_w \mathbf{g}_T - \mathbf{H}_T \mathbf{g}_w + \eta \mathbf{H}_w \mathbf{H}_T \mathbf{g}_w \quad (14)$$

$$\mathbf{B}_w = -\mathbf{H}_T \mathbf{H}_w + \frac{\eta}{2} \mathbf{H}_w \mathbf{H}_T \mathbf{H}_w \quad (15)$$

In the setting above (linear model with quadratic loss), the function f has the property that $\forall \theta, \delta \in \mathbb{R}^d, w \in \mathbb{R}^n$:

$$f(w; \theta + \delta) = f(w; \theta) + \mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta. \quad (16)$$

Proof. First notice that, for simplicity, the loss here is defined as the sum (as opposed to the average) of per-sample losses, which drops the $\frac{1}{|S|}$ terms in the loss, gradient, Hessian, and \mathbf{Q} objects. Recalling the definition of f from 5, we can write $f(w; \theta + \delta) = -\mathbf{p}(\theta + \delta)^T w + \frac{\eta}{2} w^T \mathbf{Q}(\theta + \delta) w$. Since the loss is quadratic in θ , the Hessian is independent of θ , and the derivatives above the second order are zero. Hence, defining $\mathbf{g}_i^D(\theta)$ and \mathbf{H}_i^D as the gradient and Hessian of the sample i in D , we can write:

$$\mathbf{g}_i^D(\theta + \delta) = \mathbf{g}_i^D(\theta) + \mathbf{H}_i^D \delta \quad (17)$$

for any δ with the same dimension as θ . Setting $D = T$ and summing across samples, we can write:

$$\mathbf{g}_T(\theta + \delta) = \mathbf{g}_T(\theta) + \mathbf{H}_T \delta \quad (18)$$

Additionally, setting $D = S$ and taking a weighted sum we can write:

$$\mathbf{G}_S(\theta + \delta) w = \mathbf{G}_S(\theta) w + \mathbf{H}_w \delta \quad (19)$$

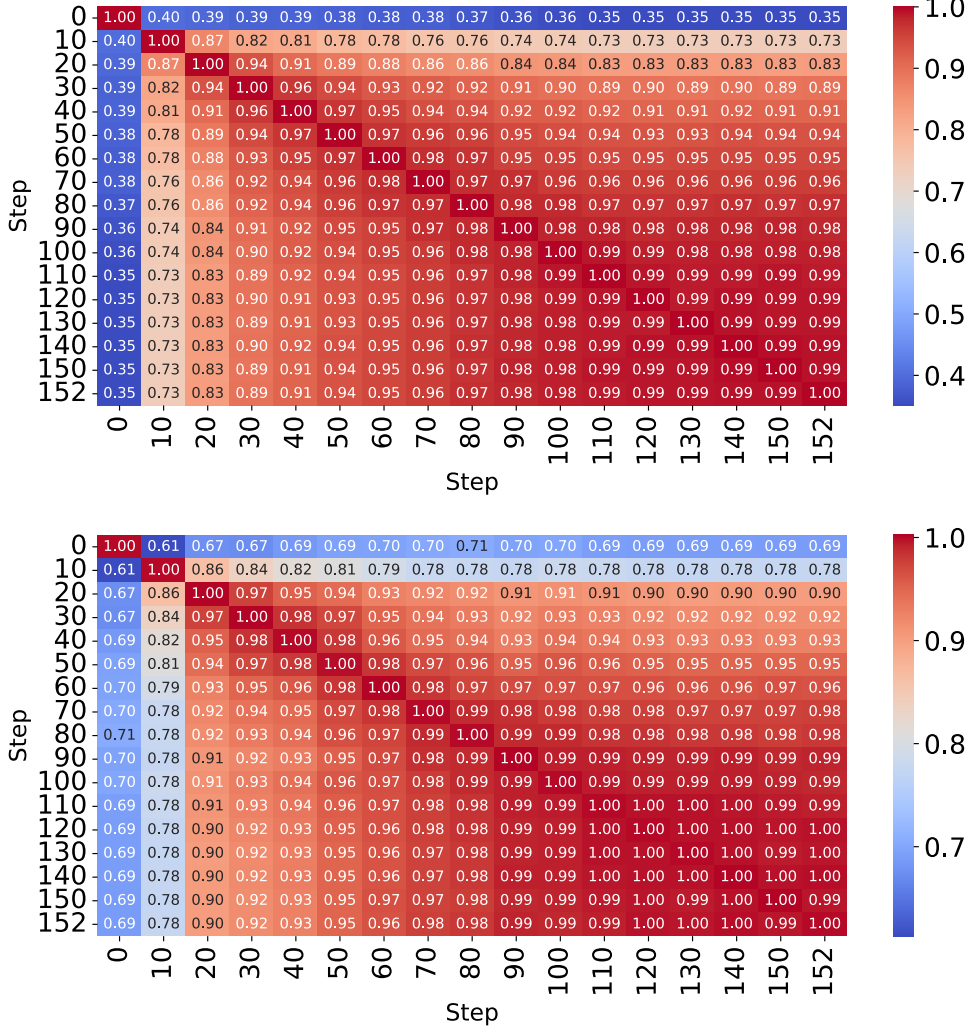


Figure 4: Average gradient cosine similarity on unseen samples from Tulu V2 (top) and BBH (bottom) across checkpoints.

601 Next, we see that,

$$\begin{aligned}
\mathbf{p}(\boldsymbol{\theta} + \boldsymbol{\delta})^T \mathbf{w} &= \mathbf{g}_T(\boldsymbol{\theta} + \boldsymbol{\delta})^T \mathbf{G}_S(\boldsymbol{\theta} + \boldsymbol{\delta}) \mathbf{w} \\
&= (\mathbf{g}_T(\boldsymbol{\theta})^T + \boldsymbol{\delta}^T \mathbf{H}_T) (\mathbf{G}_S(\boldsymbol{\theta}) \mathbf{w} + \mathbf{H}_w \boldsymbol{\delta}) \\
&= \mathbf{p}(\boldsymbol{\theta})^T \mathbf{w} + (\mathbf{g}_T(\boldsymbol{\theta})^T \mathbf{H}_w + \mathbf{g}_w(\boldsymbol{\theta})^T \mathbf{H}_T) \boldsymbol{\delta} + \boldsymbol{\delta}^T \mathbf{H}_T \mathbf{H}_w \boldsymbol{\delta}
\end{aligned} \tag{20}$$

602 And,

$$\begin{aligned}
\mathbf{w}^T \mathbf{Q}(\boldsymbol{\theta} + \boldsymbol{\delta})^T \mathbf{w} &= \mathbf{w}^T \mathbf{G}_S(\boldsymbol{\theta} + \boldsymbol{\delta})^T \mathbf{H}_T \mathbf{G}_S(\boldsymbol{\theta} + \boldsymbol{\delta}) \mathbf{w} \\
&= (\mathbf{w}^T \mathbf{G}_S(\boldsymbol{\theta})^T + \boldsymbol{\delta}^T \mathbf{H}_w) \mathbf{H}_T (\mathbf{G}_S(\boldsymbol{\theta}) \mathbf{w} + \mathbf{H}_w \boldsymbol{\delta}) \\
&= \mathbf{w}^T \mathbf{Q}(\boldsymbol{\theta}) \mathbf{w} + 2 \mathbf{g}_w(\boldsymbol{\theta})^T \mathbf{H}_T \mathbf{H}_w \boldsymbol{\delta} + \boldsymbol{\delta}^T \mathbf{H}_w \mathbf{H}_T \mathbf{H}_w \boldsymbol{\delta}
\end{aligned} \tag{21}$$

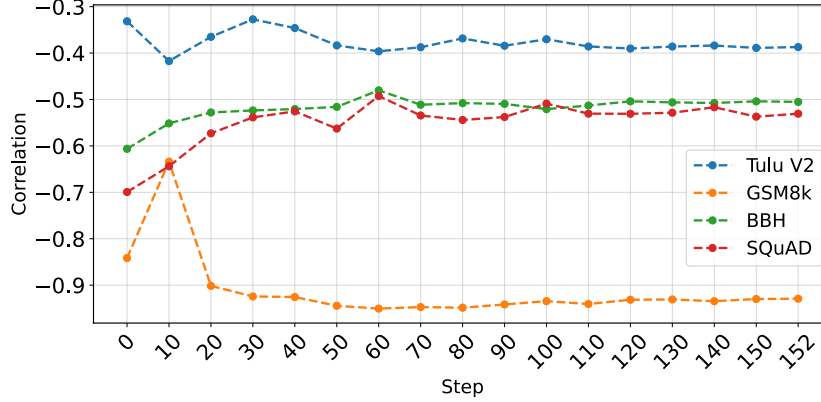


Figure 5: Correlation between gradient norm and number of label tokens, across checkpoints on four datasets.

Putting them together:

$$\begin{aligned}
f(w; \theta + \delta) &= -\mathbf{p}(\theta + \delta)^T \mathbf{w} + \frac{\eta}{2} \mathbf{w}^T \mathbf{Q}(\theta + \delta) \mathbf{w} \\
&= f(w; \theta) - ((\mathbf{g}_T(\theta)^T \mathbf{H}_w + \mathbf{g}_w(\theta)^T \mathbf{H}_T) \delta + \delta^T \mathbf{H}_T \mathbf{H}_w \delta) \\
&\quad + \frac{\eta}{2} (2\mathbf{g}_w(\theta)^T \mathbf{H}_T \mathbf{H}_w \delta + \delta^T \mathbf{H}_w \mathbf{H}_T \mathbf{H}_w \delta) \\
&= f(w; \theta) + (-\mathbf{g}_T(\theta)^T \mathbf{H}_w - \mathbf{g}_w(\theta)^T \mathbf{H}_T + \eta \mathbf{g}_w(\theta)^T \mathbf{H}_T \mathbf{H}_w) \delta \\
&\quad + \delta^T (-\mathbf{H}_T \mathbf{H}_w + \frac{\eta}{2} \mathbf{H}_w \mathbf{H}_T \mathbf{H}_w) \delta \\
&= f(w; \theta) + \mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta
\end{aligned}$$

which concludes the proof. \square

Substituting the result of the [B.1](#) into Objective [I3](#), we can write

$$\begin{aligned}
\mathbf{w}^* &= \arg \min_w \max_{\|\delta\| \leq \epsilon} [f(w; \theta) + \mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta] \\
&= \arg \min_w [f(w; \theta) + \max_{\|\delta\| \leq \epsilon} (\mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta)]
\end{aligned} \tag{22}$$

We maximize $r(\delta) = \mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta$ in the sphere with radius ϵ approximately by taking a single step of size ϵ in the gradient direction, i.e., $\delta^* \approx \epsilon \cdot \frac{r'(\mathbf{0})}{\|r'(\mathbf{0})\|}$. This approximation is standard in the sharpness-aware optimization literature [\[Foret et al., 2020, Peste et al., 2022\]](#), which addresses a similar min-max objective to search for flat minima. Note that $r'(\delta) = \mathbf{a}_w + (\mathbf{B}_w + \mathbf{B}_w^T) \delta$, hence $r'(\mathbf{0}) = \mathbf{a}_w$ and

$$\max_{\|\delta\| \leq \epsilon} (\mathbf{a}_w^T \delta + \delta^T \mathbf{B}_w \delta) \approx \epsilon \cdot \|\mathbf{a}_w\| + \epsilon^2 \cdot \frac{\mathbf{a}_w^T \mathbf{B}_w \mathbf{a}_w}{\|\mathbf{a}_w\|^2}. \tag{23}$$

Substituting into Equation [22](#), we get the following objective:

$$\mathbf{w}^* \approx \arg \min_w [f(w; \theta) + \epsilon \cdot \|\mathbf{a}_w\| + \epsilon^2 \cdot \frac{\mathbf{a}_w^T \mathbf{B}_w \mathbf{a}_w}{\|\mathbf{a}_w\|^2}]. \tag{24}$$

This suggests that the robustness of the weights can be controlled via the hyperparameter ϵ , which determines the strength of the regularization.

We apply this regularization to the running example introduced in Section [3.2](#). As shown in Figure [6](#), using the tuned value $\epsilon = 10^{-4}$ yields better performance than the default weights. However, due to the high computational cost of this regularization term, we use standard L2 regularization for general non-linear models.

C Adam Optimizer

Here we derive Equations 8 and 9, which adapt the vector \mathbf{p} and the matrix \mathbf{Q} to the case of the Adam optimizer. Assume that after a warm-up phase, the first- and second-moment estimates of Adam are \mathbf{m} and \mathbf{v} , respectively. For a new gradient \mathbf{g} , the Adam update rule can be written as:

$$\Delta\theta = -\eta \cdot \frac{\mathbf{m}'}{\sqrt{\mathbf{v}' + \epsilon}} \quad (25)$$

where η is the learning rate, and $\mathbf{m}' = \frac{\beta_1 \mathbf{m} + (1-\beta_1) \mathbf{g}}{1-\beta_1^s}$ and $\mathbf{v}' = \frac{\beta_2 \mathbf{v} + (1-\beta_2) \mathbf{g}^2}{1-\beta_2^s}$ are the updated moment estimates, with (β_1, β_2) being the Adam beta values for first- and second-order estimate updates, and s being the number steps the optimizer has already been trained for.

For a single update, we note that $\beta_2 \mathbf{v} + (1-\beta_2) \mathbf{g}^2 \approx \mathbf{v}$. That is because (1) the value β_2 is typically very close to 1, e.g., 0.995 or 0.999, and (2) due to the warm-up, \mathbf{v} is stabilized and is not expected to change much. This allows us to ignore the dependence of \mathbf{v}' on \mathbf{g} , i.e., $\mathbf{v}' \approx \frac{\mathbf{v}}{1-\beta_2^s}$ simplifying the computations.

Enabled by this, we revisit the Taylor expansion in Equation 2:

$$\begin{aligned} \mathbf{w}^* &= \arg \min_{\mathbf{w}} \mathcal{L}(\mathcal{M}^{\text{Adam}}(\theta; S, \mathbf{w}); T, \mathbb{1}) \\ &= \arg \min_{\mathbf{w}} \mathcal{L}\left(\theta - \frac{\eta}{|S|} \frac{\frac{\beta_1 \mathbf{m} + (1-\beta_1) \mathbf{G}_S(\theta) \mathbf{w}}{1-\beta_1^s}}{\sqrt{\frac{\mathbf{v}}{1-\beta_2^s} + \epsilon}}\right) \\ &= \arg \min_{\mathbf{w}} \mathcal{L}\left(\theta - \frac{\eta}{|S|} \left[\frac{\beta_1 \mathbf{m}}{(1-\beta_1^s)(\sqrt{\frac{\mathbf{v}}{1-\beta_2^s} + \epsilon})} + \frac{(1-\beta_1) \mathbf{G}_S(\theta) \mathbf{w}}{(1-\beta_1^s)(\sqrt{\frac{\mathbf{v}}{1-\beta_2^s} + \epsilon})} \right]\right) \end{aligned} \quad (26)$$

Let $\mathbf{a} = \frac{(1-\beta_1)}{(1-\beta_1^s)(\sqrt{\frac{\mathbf{v}}{1-\beta_2^s} + \epsilon})}$ and $\mathbf{b} = \frac{\beta_1 \mathbf{m}}{(1-\beta_1^s)(\sqrt{\frac{\mathbf{v}}{1-\beta_2^s} + \epsilon})}$. Construct $\mathbf{G}_S^{\text{Adam}}(\theta)$ by element-wise multiplying each column of $\mathbf{G}_S(\theta)$ by \mathbf{a} . We can now continue Equation 26 by:

$$\begin{aligned} \mathbf{w}^* &= \arg \min_{\mathbf{w}} \mathcal{L}\left(\theta - \frac{\eta}{|S|} (\mathbf{b} + \mathbf{G}_S^{\text{Adam}}(\theta) \mathbf{w})\right) \\ &\approx \arg \min_{\mathbf{w}} [\mathcal{L}(\theta; T, \mathbb{1}) - \frac{\eta}{|S|} \mathbf{g}_T^T(\theta) (\mathbf{b} + \mathbf{G}_S^{\text{Adam}}(\theta) \mathbf{w}) + \\ &\quad \frac{\eta^2}{2|S|^2} (\mathbf{b}^T + \mathbf{w}^T \mathbf{G}_S^{\text{Adam}}(\theta)^T) \mathbf{H}_T(\theta) (\mathbf{b} + \mathbf{G}_S^{\text{Adam}}(\theta) \mathbf{w})] \\ &= \arg \min_{\mathbf{w}} -(\mathbf{g}_T^T(\theta) - \frac{\eta}{|S|} \mathbf{b}^T \mathbf{H}_T(\theta)) \mathbf{G}_S^{\text{Adam}}(\theta) \mathbf{w} \\ &\quad + \frac{\eta}{2|S|} \mathbf{w}^T \mathbf{G}_S^{\text{Adam}}(\theta)^T \mathbf{H}_T(\theta) \mathbf{G}_S^{\text{Adam}}(\theta) \mathbf{w} \\ &= \arg \min_{\mathbf{w}} -\mathbf{p}^{\text{Adam}}(\theta)^T \mathbf{w} + \frac{\eta}{2} \mathbf{w}^T \mathbf{Q}^{\text{Adam}}(\theta)^T \mathbf{w} \end{aligned} \quad (27)$$

Where \mathbf{p}^{Adam} and \mathbf{Q}^{Adam} are defined in Equations 8 and 9, respectively.

D Proof of Theorem 4.1

We begin by noting a property of the landmark-based approximation introduced in Section 4.3: it exhibits *rotational equivariance*. That is, if all source and target gradients are rotated by an orthonormal matrix, the resulting landmark-based gradient approximations will also be simply rotated by the same matrix.

In the remainder of this section, we prove two useful lemmas—Lemma D.1 and Lemma D.2. We then state and prove Theorem D.3, which bounds the error in the vector \mathbf{p} for any unbiased approximation that satisfies rotational equivariance. Finally, Corollary D.4 bounds the difference in the resulting sample weights, thereby completing the proof of Theorem 4.1.

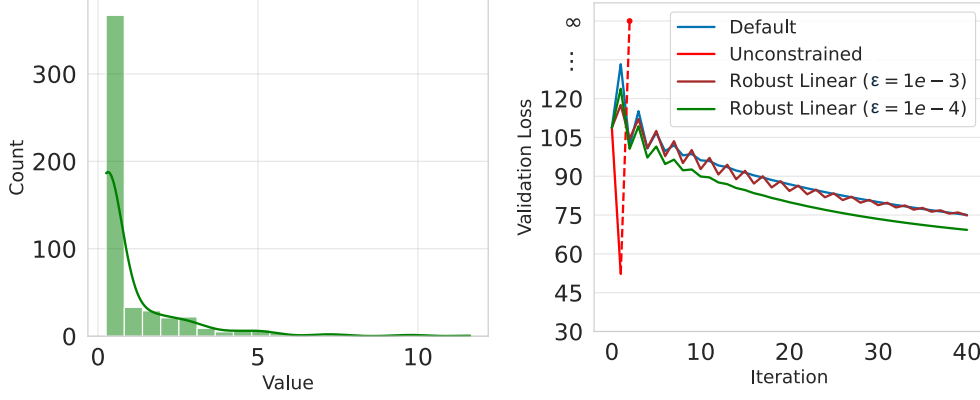


Figure 6: (Left) Distribution of theoretical robust weights for the linear case with $\epsilon = 10^{-4}$, and (Right) validation loss during training with different variants in the running experiment setting.

Lemma D.1. Given unit vectors $\mathbf{g}, \mathbf{t} \in \mathbb{R}^d$, assume $\hat{\mathbf{g}} = \mathbf{g} + \mathbf{e}$ is a noisy approximation to \mathbf{g} . Additionally, assume $\mathbf{e} \in \mathbb{R}^d$ is a zero-mean, isotropic random vector, i.e., $\mathbb{E}[\mathbf{e}] = 0$ and $\text{Cov}(\mathbf{e}) = \sigma^2 \mathbf{I}_d$ for some $\sigma > 0$. Let $S = \langle \hat{\mathbf{g}}, \mathbf{t} \rangle$. Then $\mathbb{E}[S] = \langle \mathbf{g}, \mathbf{t} \rangle$, and $\text{Var}(S) = \frac{\mathbb{E}[\|\hat{\mathbf{g}} - \mathbf{g}\|_2^2]}{d}$.

Proof. The expectation of S follows directly from zero-mean property of \mathbf{e} . To bound its variance, let Σ denote the covariance matrix of \mathbf{e} . Since \mathbf{e} is isotropic, $\Sigma = \sigma^2 \mathbf{I}_d$ for some σ . We can write:

$$\begin{aligned}
 \text{Var}(S) &= \text{Var}(\langle \hat{\mathbf{g}}, \mathbf{t} \rangle) \\
 &= \text{Var}(\langle \mathbf{g}, \mathbf{t} \rangle + \langle \mathbf{e}, \mathbf{t} \rangle) \\
 &= \text{Var}(\langle \mathbf{e}, \mathbf{t} \rangle) \\
 &= \mathbf{t}^T \Sigma \mathbf{t} \\
 &= \sigma^2 \|\mathbf{t}\|^2 \\
 &= \sigma^2
 \end{aligned} \tag{28}$$

Also,

$$\begin{aligned}
 \mathbb{E}[\|\hat{\mathbf{g}} - \mathbf{g}\|^2] &= \mathbb{E}[\|\mathbf{e}\|^2] \\
 &= \text{tr}(\Sigma) \\
 &= d\sigma^2
 \end{aligned} \tag{29}$$

Hence $\sigma^2 = \frac{\mathbb{E}[\|\hat{\mathbf{g}} - \mathbf{g}\|^2]}{d}$, which concludes the proof. \square

Lemma D.2. Assume $\mathbf{x} \in \mathbb{R}^d$ is a random vector from an arbitrary distribution. For any random orthonormal matrix of the form $\mathbf{R} = \mathbf{P}\mathbf{D}$, where

- \mathbf{P} is a random permutation matrix
- \mathbf{D} is a diagonal matrix with i.i.d. Rademacher signs (± 1)

the random vector $\mathbf{y} = \mathbf{R}\mathbf{x}$ is isotropic, i.e., $\text{Cov}(\mathbf{y}) = \sigma^2 \mathbf{I}_d$ for some real value σ .

Proof. We can write:

$$\begin{aligned}
 \text{Cov}(\mathbf{R}\mathbf{x}) &= \mathbb{E}_{\mathbf{P}, \mathbf{D}, \mathbf{x}}[\mathbf{R}\mathbf{x}\mathbf{x}^T \mathbf{R}^T] \\
 &= \mathbb{E}_{\mathbf{P}, \mathbf{D}, \mathbf{x}}[\mathbf{P}\mathbf{D}\mathbf{x}\mathbf{x}^T \mathbf{D}\mathbf{P}^T] \\
 &= \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{\mathbf{P}}[\mathbb{E}_{\mathbf{D}}[\mathbf{P}\mathbf{D}\mathbf{x}\mathbf{x}^T \mathbf{D}\mathbf{P}^T \mid \mathbf{P}, \mathbf{x}] \mid \mathbf{x}]] \\
 &= \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{\mathbf{P}}[\mathbf{P} \mathbb{E}_{\mathbf{D}}[\mathbf{D}\mathbf{x}\mathbf{x}^T \mathbf{D} \mid \mathbf{x}] \mathbf{P}^T \mid \mathbf{x}]]
 \end{aligned} \tag{30}$$

Now note that $\mathbb{E}_{\mathbf{D}}[\mathbf{D}\mathbf{x}\mathbf{x}^T\mathbf{D} \mid \mathbf{x}] = \text{diag}(\mathbf{x}^2)$. Substituting into the expectation over \mathbf{P} , we need to compute $\mathbb{E}_{\mathbf{P}}[\mathbf{P}\text{diag}(\mathbf{x}^2)\mathbf{P}^T \mid \mathbf{x}]$. However, since \mathbf{P} is a random permutation, off-diagonal elements are zero and for the diagonal elements, any element of \mathbf{x}^2 can be picked with equal probability. Hence, the expectation over \mathbf{P} equals $\frac{1}{d}\|\mathbf{x}\|^2\mathbf{I}_d$.

Putting it all together in Equation 30, we get

$$\text{Cov}(\mathbf{R}\mathbf{x}) = \frac{\mathbb{E}[\|\mathbf{x}\|^2]}{d}\mathbf{I}_d, \quad (31)$$

which concludes the proof. \square

Theorem D.3. Let $\mathbf{G} \in \mathbb{R}^{n \times d}$ and $\mathbf{t} \in \mathbb{R}^d$, with \mathbf{t} and each row of \mathbf{G} having unit lengths. Let \mathbf{g}_i denote the i 'th row in \mathbf{G} . Additionally, assume access to a (randomized) mapping function $\mathbf{h} : \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n\} \rightarrow \mathbb{R}^d$, and let $\forall i \in \{1, 2, \dots, n\} : \hat{\mathbf{g}}_i = \mathbf{h}(\mathbf{g}_i; \mathbf{G})$. Additionally, assume $\mathbf{h}(\cdot)$ satisfies:

1. *Unbiased:* $\forall i \in \{1, 2, \dots, n\} : \mathbb{E}[\hat{\mathbf{g}}_i] = \mathbf{g}_i$, i.e., $\mathbf{h}(\cdot)$ is unbiased.
2. *Bounded Average Mean Squared Error:* Let $\delta_i^2 = \mathbb{E}[\|\hat{\mathbf{g}}_i - \mathbf{g}_i\|^2]$. Then:

$$\frac{1}{n} \sum_{i=1}^n \delta_i^2 \leq \Delta^2$$

for some $\Delta^2 \geq 0$.

3. *Rotation Equivariance:* For any orthonormal rotation matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$ and $\forall i \in \{1, 2, \dots, n\} : \mathbf{h}(\mathbf{R}\mathbf{g}_i; \mathbf{G}\mathbf{R}) = \mathbf{R}\hat{\mathbf{g}}_i$.

Construct the vector $\mathbf{p} = [p_1, p_2, \dots, p_n]^T$ such that $p_i = \langle \mathbf{g}_i, \mathbf{t} \rangle$. Similarly define $\hat{\mathbf{p}} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n]^T$, where $\hat{p}_i = \langle \hat{\mathbf{g}}_i, \mathbf{t} \rangle$. Then

$$\mathbb{E}[\|\mathbf{p} - \hat{\mathbf{p}}\|^2] \leq \frac{n\Delta^2}{d} \quad (32)$$

Proof. For all i , let $\mathbf{e}_i = \hat{\mathbf{g}}_i - \mathbf{g}_i$ denote the error. By the *Unibased* assumption, $\mathbb{E}[\mathbf{e}_i] = \mathbf{0}$.

Without loss of generality, we can assume that for all i , the vector \mathbf{e}_i is isotropic, i.e., $\text{Cov}(\mathbf{e}_i)$ is a scalar multiple of the identity matrix. If this is not the case, we take advantage of Lemma D.2 and apply a change of variables: $\mathbf{G} \leftarrow \mathbf{G}\mathbf{R}$ and $\mathbf{t} \leftarrow \mathbf{R}\mathbf{t}$, where $\mathbf{R} = \mathbf{P}\mathbf{D}$, \mathbf{P} is a permutation matrix, and \mathbf{D} is a diagonal matrix with entries chosen uniformly at random from $\{\pm 1\}$. Note that by the *Rotation Equivariance* assumption, this transformation implies $\hat{\mathbf{g}}_i \leftarrow \mathbf{R}\hat{\mathbf{g}}_i$. Under this transformation, the error vectors \mathbf{e}_i are mapped into a space where they become isotropic, and the pairwise dot products and distances remain unchanged as \mathbf{R} is orthonormal.

Now we can directly apply Lemma D.1 for each coordinate i : $\mathbb{E}[\hat{p}_i] = p_i$ and $\text{Var}(\hat{p}_i) = \frac{\mathbb{E}[\|\hat{\mathbf{g}}_i - \mathbf{g}_i\|^2]}{d}$. This means:

$$\begin{aligned} \mathbb{E}[\|\mathbf{p} - \hat{\mathbf{p}}\|^2] &= \sum_{i=1}^n \mathbb{E}[(p_i - \hat{p}_i)^2] \\ &= \sum_{i=1}^n \text{Var}(\hat{p}_i) \\ &= \sum_{i=1}^n \frac{\mathbb{E}[\|\hat{\mathbf{g}}_i - \mathbf{g}_i\|^2]}{d} \\ &\leq \frac{n\Delta^2}{d} \end{aligned}$$

where the last inequality comes from the *Bounded Average Mean Squared Error* assumption. \square

Corollary D.4. In the setting of Theorem D.3 if we define:

$$\mathbf{w}(\mathbf{p}) = \arg \min_{\mathbf{x}} -\mathbf{p}^T \mathbf{x} + \frac{\lambda}{2} \|\mathbf{x}\|_2^2, \quad \text{s.t.} \quad \begin{cases} \mathbf{x} \geq \mathbf{0} \\ \mathbf{x}^T \mathbf{1} = n \end{cases} \quad (33)$$

683 then

$$\mathbb{E}[\|\mathbf{w}(\mathbf{p}) - \mathbf{w}(\hat{\mathbf{p}})\|^2] \leq \frac{n\Delta^2}{\lambda^2 d}. \quad (34)$$

684 *Proof.* Let $F_p(x) = -\mathbf{p}^T \mathbf{x} + \frac{\lambda}{2} \|\mathbf{x}\|_2^2$ and $C = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} \geq 0, \mathbf{x}^T \mathbf{1} = n\}$. Note that the objective
685 above has a unique solution since F_p is λ -strongly convex and C is a convex set independent of \mathbf{p} .

686 By strong convexity, $\forall x, y \in \mathbb{R}^d$:

$$F_p(\mathbf{y}) \geq F_p(\mathbf{x}) + \nabla_x F_p(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{\lambda}{2} \|\mathbf{y} - \mathbf{x}\|^2 \quad (35)$$

687 Set $\mathbf{x} = \mathbf{w} := \mathbf{w}(\mathbf{p})$ and $\mathbf{y} = \hat{\mathbf{w}} := \mathbf{w}(\hat{\mathbf{p}})$. Since \mathbf{w} minimizes F_p over C , $\nabla_x F_p(\mathbf{x})^T (\mathbf{y} - \mathbf{w}) \geq 0$.
688 Hence:

$$F_p(\hat{\mathbf{w}}) \geq F_p(\mathbf{w}) + \frac{\lambda}{2} \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \quad (36)$$

689 Swapping \mathbf{w} and $\hat{\mathbf{w}}$,

$$F_{\hat{p}}(\mathbf{w}) \geq F_{\hat{p}}(\hat{\mathbf{w}}) + \frac{\lambda}{2} \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \quad (37)$$

690 Adding the two equations above:

$$(\mathbf{p} - \hat{\mathbf{p}})^T (\mathbf{w} - \hat{\mathbf{w}}) \geq \lambda \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \quad (38)$$

691 Applying Cauchy-Schwarz on the left hand side, we get

$$\|\mathbf{p} - \hat{\mathbf{p}}\| \cdot \|\mathbf{w} - \hat{\mathbf{w}}\| \geq \lambda \|\mathbf{w} - \hat{\mathbf{w}}\|^2 \quad (39)$$

692 Hence

$$\|\mathbf{w} - \hat{\mathbf{w}}\| \leq \frac{1}{\lambda} \|\mathbf{p} - \hat{\mathbf{p}}\| \quad (40)$$

693 Combining with the result of Theorem D.3:

$$\mathbb{E}[\|\mathbf{w} - \hat{\mathbf{w}}\|^2] \leq \frac{n\Delta^2}{\lambda^2 d}. \quad (41)$$

694

□

695 E Dataset and Model Details

696 This section provides details on the datasets and models used throughout the paper.

697 E.1 Datasets

698 For the datasets, we largely follow the setup of Ivison et al. [2025].

699 **Tulu V2 (ODC-BY License).** The Tulu V2 dataset [Ivison et al., 2023], also known as the Tulu
700 V2 SFT Mixture, is a comprehensive instruction-tuning dataset. Following Ivison et al. [2025], we
701 consider the unfiltered version with 5.8M samples, consisting of 961,322 samples from FLAN v2
702 [Chung et al., 2024], 398,439 samples from FLAN CoT [Chung et al., 2024], 7,707 samples from
703 Open Assistant [Köpf et al., 2023], 15,007 from Dolly [Conover et al., 2023], 52,002 from GPT-4
704 Alpaca [Peng et al., 2023], 20,022 from Code Alpaca [Chaudhary, 2023], 100,054 from ShareGPT,
705 1,030 from LIMA [Zhou et al., 2023b], 142,802 from Wizard Evol-Instruct V2 [Xu et al., 2023],
706 4,111,858 from Open Orca [Lian et al., 2023], 7,535 from SciRIFF [Wadden et al., 2024], and 14
707 from Hardcoded. For more information, we refer the reader to Ivison et al. [2025].

708 **MMLU (MIT License).** The Massive Multitask Language Understanding (MMLU) dataset
709 [Hendrycks et al., 2021a, b] consists of challenging multiple-choice questions from 57 topics, such
710 as abstract algebra, astronomy, machine learning, and more. It includes 5 development samples per
711 category and a total of 14,042 test samples. We use the development samples as our target set and
712 evaluate the final model zero-shot on the test set.

713 **GSM8K (MIT License).** This dataset comprises grade school math questions, with 7.47k training
714 and 1.32k test samples [Cobbe et al., 2021]. We evaluate the models on the test set using 8 examples

in the context (8-shot evaluation) and use the same 8 individual samples as the target set. As is standard, only the final answer to each question is considered.

Big-Bench-Hard (MIT License). This dataset includes questions from 27 challenging tasks, such as causal judgment, multi-step arithmetic, and logic. Following [Suzgun et al., 2022], we perform 3-shot evaluations using the same 3 samples per category (a total of 81) as the target set.

TyDIQA (Apache-2.0 License). TyDIQA is a dataset of 204k question-answering samples across 11 languages [Clark et al., 2020]. For evaluation, we follow [Ivison et al., 2025], which in turn follows [Anil et al., 2023], using 1-shot prompting. We select 9 samples per language for the target set.

Codex (MIT License). This dataset contains 164 Python programming questions [Chen et al., 2021], of which 16 are used as the target set and the remaining as the test set. See [Ivison et al., 2025] for additional evaluation details.

SQuAD (CC BY-SA 4.0 License). The Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2016] contains reading comprehension questions based on Wikipedia articles. We use 500 random samples from the training split as the target set. We perform 3-shot evaluations with three samples randomly selected from the training set.

E.2 Model Licenses

In this paper, we utilize LLaMA 2 [Touvron et al., 2023], LLaMA 3.2 3B [Grattafiori et al., 2024], Qwen 2.5 1.5B [Team, 2024], and Qwen 2.5 3B [Team, 2024] models. These models are distributed under the LLaMA 2 Community License, LLaMA 3.2 Community License, Apache-2.0 License, and Qwen Research License, respectively.

F Embeddings Study

In Section 4.3, we noted that existing embedding functions are insufficient for our landmark-based gradient approximations and introduced the JVP embeddings as an alternative. In this section, we compare different embedding functions in two settings. In all the experiments, the model we consider is Llama-2 7B [Touvron et al., 2023].

Gradient Recovery. First, we randomly take 200k samples from Tulu V2 [Ivison et al., 2023] and embed them using various embedding functions. We then use a small number of landmark gradient samples (selected uniformly at random) to approximate the gradients for all data points, following the method described in Section 4.3. This process is repeated for different numbers of landmarks to evaluate how performance varies with landmark count. We report the average cosine similarity between the approximated gradients and the true gradients (projected into 8192-dimensional space using Rademacher-based projections [Ivison et al., 2025, Park et al., 2023]) for each case.

We evaluate several embedding functions: the RDS+ embeddings from [Ivison et al., 2025], NVIDIA’s NV-Embed-v2 [Lee et al., 2024], GTR-base [Ni et al., 2021], and our proposed JVP-based approach using two random vectors and four transformer blocks.

As a lower bound, we also include a *Trivial* embedding: here, we assume that the gradients for the landmark samples are perfectly recovered, while the gradients for all other samples are treated as completely random.

Figure 7 (Left) presents a comparison of these embedding functions. Our JVP embeddings outperform all other methods, including the more computationally intensive RDS+ and NV-Embed-v2.

Finally, we compute an upper bound by using the true projected gradients as the embedding function and repeating the same experiment. As shown in Figure 7 (Right), this idealized setting quickly achieves high accuracy in gradient approximation—surpassing 0.9 cosine similarity with just over 4096 landmarks. This suggests that the gradients are approximately low-rank, a known phenomenon in LLMs [Hu et al., 2022, Zhao et al., 2024].

End-to-end Selection and Training. We repeat the selection and fine-tuning experiments from Table 1, this time replacing the JVP embeddings with either GTR-base or true gradient embeddings. Table 2 reports the resulting accuracy for each task. Due to the high computational cost of obtaining true gradients, we include only a single random seed for this setting.

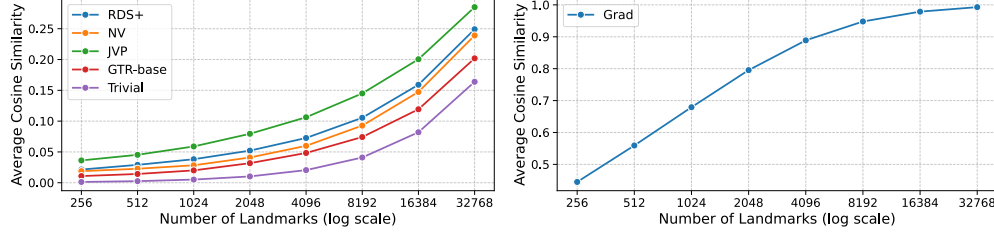


Figure 7: (Left) Gradient direction recovery vs number of landmarks, when different proxy embedding functions are used, and (Right) gradient direction recovery when the actual gradients are used as an ideal embedding.

Table 2: Accuracy (\pm standard deviation) of Llama2-7B across six tasks when using Influence Distillation with different embeddings to select 10k samples from a pool of 200k in the Tulu V2 dataset [Iverson et al., 2023]. The number of landmarks is fixed at 4096.

Model	Embedding	MLLU	GSM8k	BBH	TyDIQA	CODEX	SQuAD	Avg. Δ w/ Uniform
Llama2-7B	GTR-base	46.7 \pm 0.17	18.7 \pm 0.27	42.8 \pm 0.34	52.2 \pm 0.56	29.3 \pm 0.84	82.1 \pm 0.30	45.3
	JVP	48.3 \pm 0.21	20.3 \pm 1.65	43.2 \pm 0.67	53.6 \pm 0.34	29.5 \pm 3.14	83.2 \pm 1.02	46.4
	Grad	48.3	20.2	43.7	51.7	27.7	84.5	46.0

We fix the number of landmarks to 4096 across all experiments. The results show that while GTR-base consistently underperforms, the JVP and true gradient embeddings yield comparable accuracy—falling within each other’s standard deviation in most cases. This indicates that the gradient approximations provided by JVP embeddings are sufficiently accurate for end-to-end training.

Finally, we note that since Figure 7 (Right) demonstrates near-perfect gradient recovery using the Grad embedding, the corresponding row in Table 2 closely mirrors the performance of the LESS method [Xia et al., 2024].

G An Active-Set Solution

In this appendix we derive the solution to the Influence Distillation objective under the assumption that $\eta\mathbf{Q} + \lambda\mathbf{I}$ is positive definite (PD). This setting includes the special first-order case used in the main body of the paper, where $\eta \rightarrow 0$. Concretely, we solve

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} -\mathbf{p}^T \mathbf{w} + \frac{\eta}{2} \mathbf{w}^T \mathbf{Q} \mathbf{w} + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}, \quad s.t. \quad \begin{cases} \mathbf{w} \geq 0 \\ \mathbf{w}^T \mathbf{1} = n \end{cases} \quad (42)$$

where n denotes the dimension of \mathbf{w} and $\eta\mathbf{Q} + \lambda\mathbf{I} \succ \mathbf{0}$.

Introduce the Lagrange multipliers $\tau \in \mathbb{R}$ for the equality constraint and $\alpha \in \mathbb{R}_{\geq 0}^n$ for the non-negativity constraints. The Lagrangian is

$$L(\mathbf{w}, \tau, \alpha) = -\mathbf{p}^T \mathbf{w} + \frac{\eta}{2} \mathbf{w}^T \mathbf{Q} \mathbf{w} + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} - \tau(\mathbf{1}^T \mathbf{w} - n) - \alpha^T \mathbf{w}. \quad (43)$$

Differentiating L with respect to \mathbf{w} and setting it equal to zero yields

$$\eta\mathbf{Q}\mathbf{w} + \lambda\mathbf{w} - \mathbf{p} - \tau\mathbf{1} - \alpha = \mathbf{0}. \quad (44)$$

Let $\mathbf{R} := \eta\mathbf{Q} + \lambda\mathbf{I} \succ \mathbf{0}$. Then

$$\mathbf{R}\mathbf{w} - \mathbf{p} - \tau\mathbf{1} - \alpha = \mathbf{0}. \quad (45)$$

By complementary slackness, $\forall i : \mathbf{w}_i \alpha_i = 0$. Let $A = \{i \mid \mathbf{w}_i = 0\}$ be the active set and B its complement. Restricting (45) to the free indices gives

$$\mathbf{R}_{BB} \mathbf{w}_B = \mathbf{p}_B + \tau \mathbf{1}_B. \quad (46)$$

Because \mathbf{R}_{BB} is a principal sub-matrix of the PD matrix \mathbf{R} , it is itself PD. Hence

$$\mathbf{w}(\tau; B) = \mathbf{R}_{BB}^{-1}(\mathbf{p}_B + \tau \mathbf{1}_B). \quad (47)$$

783 Enforcing $\mathbb{1}^T \mathbf{w} = n$ determines τ :

$$\mathbb{1}_B^T \mathbf{R}_{BB}^{-1} (\mathbf{p}_B + \tau \mathbb{1}_B) = n, \quad (48)$$

784 and therefore

$$\tau^* = \frac{n - \mathbb{1}_B^T \mathbf{R}_{BB}^{-1} \mathbf{p}_B}{\mathbb{1}_B^T \mathbf{R}_{BB}^{-1} \mathbb{1}_B}. \quad (49)$$

785 Substituting τ^* back into $\mathbf{w}(\tau; B)$ gives us the weights on B :

$$\mathbf{w}_B^* = \mathbf{R}_{BB}^{-1} \left(\mathbf{p}_B + \left(\frac{n - \mathbb{1}_B^T \mathbf{R}_{BB}^{-1} \mathbf{p}_B}{\mathbb{1}_B^T \mathbf{R}_{BB}^{-1} \mathbb{1}_B} \right) \mathbb{1}_B \right). \quad (50)$$

786 For indices in the active set A we have $\mathbf{w}_A^* = \mathbf{0}$, giving the final candidate solution $\mathbf{w}^* = (\mathbf{w}_A^*, \mathbf{w}_B^*)$.

787 Optimality requires that the remaining Karush–Kuhn–Tucker (KKT) conditions hold, namely $\forall i \in B$, $\mathbf{w}_i \geq 0$ (primal feasibility) and $\forall j \in A$, $\alpha_j \geq 0$ (dual feasibility). Because the objective is convex ($\mathbf{R} \succ 0$), any partition A, B satisfying these conditions is the global optimum.

790 Examining the coordinates in A in (45) gives

$$\alpha_A = (\mathbf{R}_{AB} \mathbf{w}_B^*)_A - \mathbf{p}_A - \tau^* \mathbb{1}_A. \quad (51)$$

791 Problems of this type are typically solved with a primal–dual active-set algorithm. We start from the feasible point $\mathbf{w} = \mathbb{1}$ (so $A = \emptyset$, $B = \{1, \dots, n\}$) and repeat:

- 793 1. Solve for \mathbf{w}_B^* via (50).
- 794 2. If any component of \mathbf{w}_B^* is negative, move its index to A .
- 795 3. Compute α_A ; if any component is negative, move its index back to B .

796 Each move strictly decreases the objective, and with only finitely many index sets the algorithm terminates once all components of \mathbf{w}_B and α_A are non-negative.

798 **The Special Case of $\eta \rightarrow 0$.** This setting corresponds to the first-order Influence Distillation variant used throughout the main body of the paper. In this case, we demonstrate that as λ increases, the solution \mathbf{w}^* becomes denser—that is, it contains more non-zero elements. This observation is leveraged in Section 4.4 for tuning the parameter λ .

802 When $\eta \rightarrow 0$, we can write $\mathbf{R} = \lambda \mathbf{I}$, which implies $\mathbf{R}_{BB}^{-1} = \frac{1}{\lambda} \mathbf{I}$ and $\mathbf{R}_{AB} = \mathbf{0}$. Substituting these into Equations 49, 50, and 51, we obtain:

$$\tau^* = \frac{n\lambda - \mathbb{1}_B^T \mathbf{p}_B}{|B|} \quad (52)$$

$$\mathbf{w}_B^* = \frac{1}{\lambda} (\mathbf{p} + \tau^* \mathbb{1})_B \quad (53)$$

$$\alpha_A = -(\mathbf{p} + \tau^* \mathbb{1})_A \quad (54)$$

804 Since both \mathbf{w}_B and α must be non-negative, the last two equations imply that the active set B must satisfy $B = \{i : \mathbf{p}_i \geq \tau^*\}$, i.e., B is necessarily a set of top- k elements from \mathbf{p} for some k .

806 Consider two values $\lambda_1 < \lambda_2$, and let B_1 and B_2 denote their optimal supports with sizes k_1 and k_2 , and $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ their respective optimal weight vectors; similarly, let $\alpha^{(1)}$ and $\alpha^{(2)}$ denote their associated dual variables. Suppose for contradiction that $k_2 < k_1$. Note that B_1 consists of the indices of the top k_1 elements in \mathbf{p} , while $B_2 \subset B_1$ includes the top k_2 elements of \mathbf{p} . Let s_{k_1} and s_{k_2} represent the sums of the top k_1 and k_2 elements in \mathbf{p} , respectively. Define j as the index of the k_1 -th largest element in \mathbf{p} . Since $j \in B_1$, we have $\mathbf{w}_j^{(1)} \geq 0$, and since $j \notin B_2$, it follows that

812 $\alpha_j^{(2)} \geq 0$. Therefore,

$$\begin{aligned} \mathbf{w}_j^{(1)} &\geq 0 \\ \Rightarrow \mathbf{p}_j + \frac{n\lambda_1 - s_{k_1}}{k_1} &\geq 0 \\ \Rightarrow n\lambda_1 &\geq s_{k_1} - k_1\mathbf{p}_j = \sum_{i \in B_1} (\mathbf{p}_i - \mathbf{p}_j) \end{aligned}$$

813 and

$$\begin{aligned} \alpha_j^{(2)} &\geq 0 \\ \Rightarrow \mathbf{p}_j + \frac{n\lambda_2 - s_{k_2}}{k_2} &\leq 0 \\ \Rightarrow n\lambda_2 &\leq s_{k_2} - k_2\mathbf{p}_j = \sum_{i \in B_2} (\mathbf{p}_i - \mathbf{p}_j) \end{aligned}$$

814 Observe that $\sum_{i \in B_2} (\mathbf{p}_i - \mathbf{p}_j) \leq \sum_{i \in B_1} (\mathbf{p}_i - \mathbf{p}_j)$ by definition of \mathbf{p}_j , leading to the inequality
815 $n\lambda_2 \leq n\lambda_1$, which contradicts our initial assumption that $\lambda_1 < \lambda_2$.

816 This contradiction confirms that as the regularization parameter λ increases, the solution becomes
817 progressively denser. Specifically, at $\lambda = 0$, the solution concentrates all weight on the largest
818 element of \mathbf{p} to minimize the objective, whereas in the limit as $\lambda \rightarrow \infty$, the regularization dominates,
819 resulting in $\mathbf{w} = \mathbb{1}$.

820 H First- vs Second-Order Influence Distillation

821 Recall the robust Objective [7](#)

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} f(\mathbf{w}; \boldsymbol{\theta}) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \quad s.t. \quad \begin{cases} \mathbf{w} \geq 0 \\ \mathbf{w}^T \mathbb{1} = |S| \end{cases} \quad (55)$$

822 where,

$$f(\mathbf{w}; \boldsymbol{\theta}) = -\mathbf{p}(\boldsymbol{\theta})^T \mathbf{w} + \frac{\eta}{2} \mathbf{w}^T \mathbf{Q}(\boldsymbol{\theta}) \mathbf{w} \quad (56)$$

823 In this section, we compare the first-order term $T_1 = \mathbf{p}(\boldsymbol{\theta})^T \mathbf{w}$ with the second-order term $T_2 =$
824 $\frac{\eta}{2} \mathbf{w}^T \mathbf{Q}(\boldsymbol{\theta}) \mathbf{w}$. To do so, we sample 128 random examples from the Tulu V2 dataset [\[Iverson et al.,](#)
825 [2023\]](#) as the source dataset, and 4 examples from either GSM8k [\[Cobbe et al., 2021\]](#) or MMLU
826 [\[Hendrycks et al., 2021a, b\]](#) as the target dataset.

827 We compute the vectors \mathbf{p} and the matrices \mathbf{Q} exactly for the Qwen-2.5 1.5B model [\[Team, 2024\]](#),
828 using Hessian-vector products to obtain \mathbf{Q} . We then evaluate both T_1 and T_2 using default weights
829 $\mathbf{w} = \mathbb{1}$ and a range of learning rates. To measure the relative contribution of the second-order term,
830 we report the ratio $\left| \frac{T_2}{T_1} \right|$.

831 As shown in Figure [8](#), the second-order term is generally negligible for practical learning rates
832 ($\eta \leq 10^{-4}$), indicating that the first-order approximation is sufficient in this setting.

833 I Projection Details

834 While in some of our lower-cost experiments we employ Rademacher-based projections—including
835 projecting JVP embeddings to a 4096-dimensional space using this method, as supported on GPUs
836 by [Park et al. \[2023\]](#)—we find that projecting the landmark gradients with Rademacher projections
837 becomes a computational bottleneck. To address this, we instead use a combination of pre-masking
838 and Randomized Hadamard Transform-based projections, as described below.

839 **Hadamard-based Projection.** Given a high-dimensional gradient vector \mathbf{g} , we first pad it with
840 zeros to the nearest power of two, 2^k . Then, we apply a random sign (± 1) to each element. The
841 signed vector is reshaped into a matrix \mathbf{X} of dimensions $m = 2^{\lceil \frac{k}{2} \rceil}$ and $n = 2^{\lfloor \frac{k}{2} \rfloor}$. We then apply

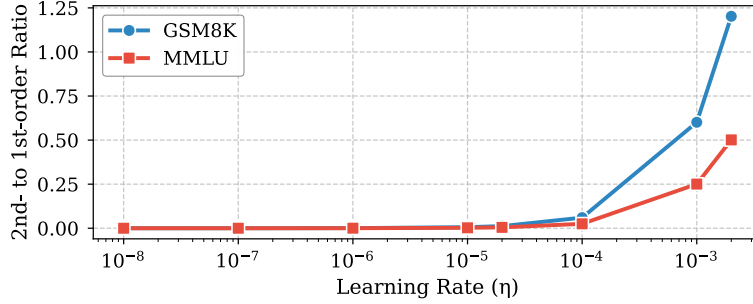


Figure 8: Ratio of second- to first-order terms for Qwen-2.5 1.5B across learning rates on two target datasets.

Table 3: Accuracy (\pm standard deviation) of Llama2-7B across six tasks when using Influence Distillation to select 10k samples from a pool of 200k in the Tulu V2 dataset [Iverson et al., 2023], with and without loss weighting during training. The number of landmarks is fixed at 8192.

Model	Embedding	MMLU	GSM8k	BBH	TyDIQA	CODEX	SQuAD	Avg. Δ w/ Uniform
Llama2-7B	Weighted	47.8 ± 0.16	19.5 ± 0.06	42.3 ± 0.26	52.2 ± 1.38	27.0 ± 2.53	84.4 ± 0.48	+1.48
	Not Weighted	48.2 ± 0.35	19.6 ± 0.79	42.4 ± 0.14	52.7 ± 1.67	29.3 ± 1.27	83.4 ± 0.86	+1.93

842 Hadamard transforms from both sides: $\mathbf{H}_m^T \mathbf{X} \mathbf{H}_n$. The resulting matrix is flattened, and a random
843 subset of its entries is selected as the projected vector.

844 Importantly, both the random sign patterns and the final index subset are generated once and reused
845 across all projected vectors. This ensures consistency and enables meaningful comparison. The
846 left and right Hadamard transforms are highly efficient and provide strong mixing across rows and
847 columns.

848 **Pre-masking.** Although efficient GPU implementations of the Hadamard transform exist [Agarwal
849 et al., 2024, Dao, 2023], they support transforms up to dimension $2^{15} = 32,768$. This allows us to
850 efficiently project vectors of up to $2^{30} = 1,073,741,824$ elements—just over one billion. However,
851 the full gradients of large language models (LLMs) can exceed this size.

852 To address this, we apply *pre-masking*: we randomly select one billion elements from the gradient
853 vector before projection. For LLaMA-2 7B [Touvron et al., 2023], we select these elements from the
854 `down_proj` matrices, which we find to represent the overall gradients well. For smaller models, we
855 randomly sample one billion elements from the entire gradient vector.

856 J Weighted Training Loss

857 In this section, we investigate the effect of incorporating the weights derived by Influence Distillation
858 into the training loss. Specifically, we conduct an experiment using LLaMA-2 7B [Touvron et al.,
859 2023], with a pool size of 200k and 8192 landmarks sampled from Tulu V2 [Iverson et al., 2023].
860 During training, we scale the loss of each selected sample by its corresponding weight.

861 Table 3 compares this weighted training setup with a baseline where the weights of the selected
862 samples are ignored. The results show that incorporating weights during training does not improve
863 performance—and in some cases, it may even degrade it. This may be due to some samples having
864 near-zero weights, effectively pruning them from the training process.

865 K Differed Figures

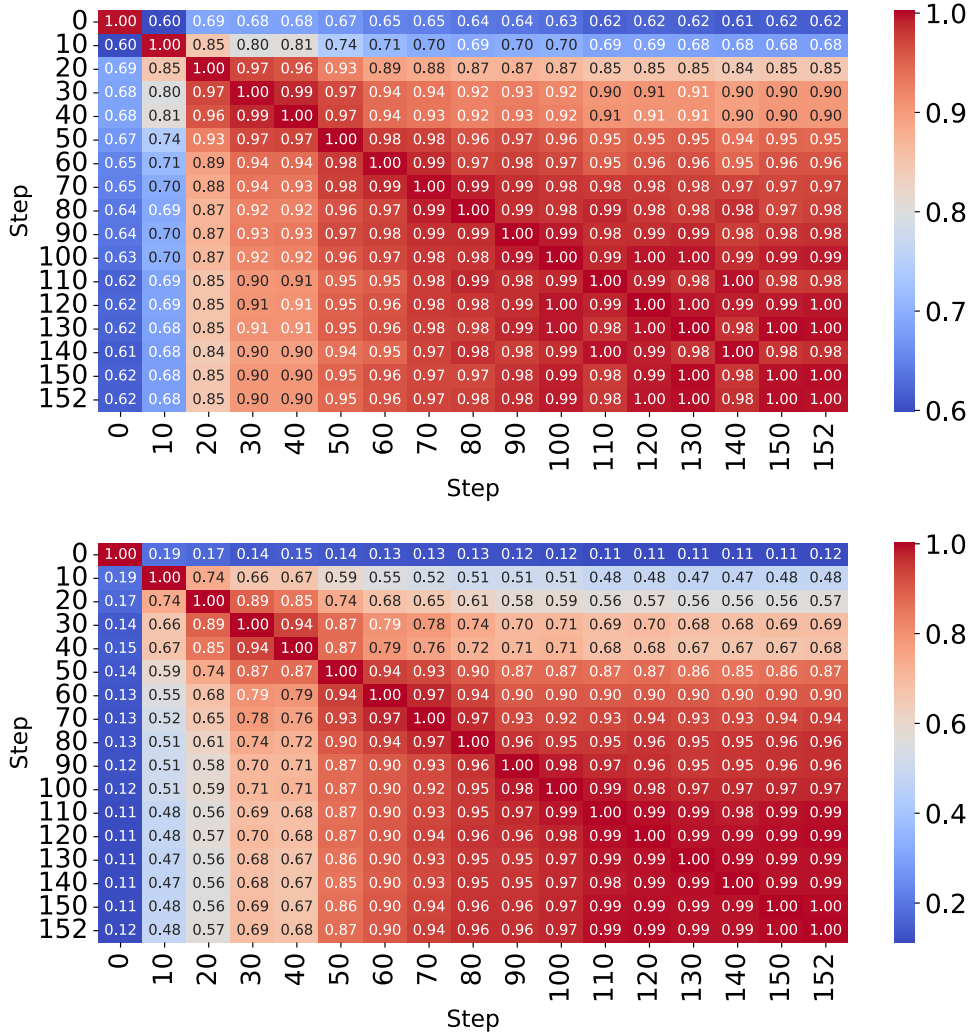


Figure 9: Average gradient cosine similarity on unseen samples from GSM8k (top) and SQuAD (bottom) across checkpoints.