LOSS-AWARE OPTIMIZATION FOR MODEL MERGING IN A LOW-DIMENSIONAL SUBSPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging aims to combine multiple task-specific models into a single model without joint retraining. Many existing methods construct merged updates from geometric or structural information, leaving their relation to downstream task losses largely indirect. To address this limitation, we formulate model merging as local optimization of the average supervised loss around a strong merged anchor. Using calibration subsets, we linearize each task output at the anchor and apply a second-order Taylor expansion to the resulting tangent loss, yielding a quadratic surrogate based on generalized Gauss–Newton curvature. This curvature captures the sensitivity of each task loss to perturbations along different parameter directions, distinguishing directions that are costly to change from those that can accommodate larger deviations. We further restrict optimization to a low-dimensional subspace spanned by structured components of merged updates from strong existing methods. This restriction retains the structural information encoded by these updates, regularizes the optimization by constraining the search space, and reduces the quadratic problem to a low-dimensional linear system with a closed-form solution. Our method can be integrated with a broad range of existing merging methods and jointly leverage updates from multiple sources. Extensive experiments on benchmarks demonstrate notable improvements over strong baselines across model architectures and task collections.

1 INTRODUCTION

Fine-tuning a shared pre-trained model effectively adapts it to different downstream tasks. As specialized models proliferate, however, storing and deploying one model per task becomes increasingly costly. Model merging addresses this problem by combining multiple fine-tuned models into a single model without jointly retraining it on the combined task data (Wortsman et al., 2022; Ilharco et al., 2023; Ramé et al., 2023). The main challenge is to preserve useful task knowledge while limiting loss increases from interference among task updates.

Many methods construct merged updates from parameter geometry or model structure by resolving magnitude and sign conflicts, exploiting sparsity or spectral components, or aligning intermediate representations (Yadav et al., 2023; Yu et al., 2024; Gargiulo et al., 2025; Cheng et al., 2025; Wang et al., 2024; Davari & Belilovsky, 2024; Deep et al., 2024; Huang et al., 2024). Although effective, these criteria only indirectly reflect downstream losses; even data dependent variants typically optimize representation alignment or other surrogates rather than the average supervised loss.

Curvature-based methods describe loss geometry more explicitly but generally construct a new merged model from statistics evaluated separately at individual fine-tuned solutions (Matena & Raffel, 2022; Daheim et al., 2024). Because full parameter curvature is expensive to represent and optimize, practical implementations often use simplified structures such as diagonal approximations. Meanwhile, methods operating in parameter, representation, and spectral spaces capture complementary structure for combining task updates that purely curvature-based constructions may overlook (Singh & Jaggi, 2020; Ainsworth et al., 2023; Akiba et al., 2025).

To address these limitations, we use a strong merged model as an anchor and locally approximate the average supervised loss around it. This common reference allows the losses and directional sensitivities of all tasks to be evaluated jointly. Figure 1 illustrates how curvature distinguishes sensitive from flat directions to balance conflicting task preferences.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

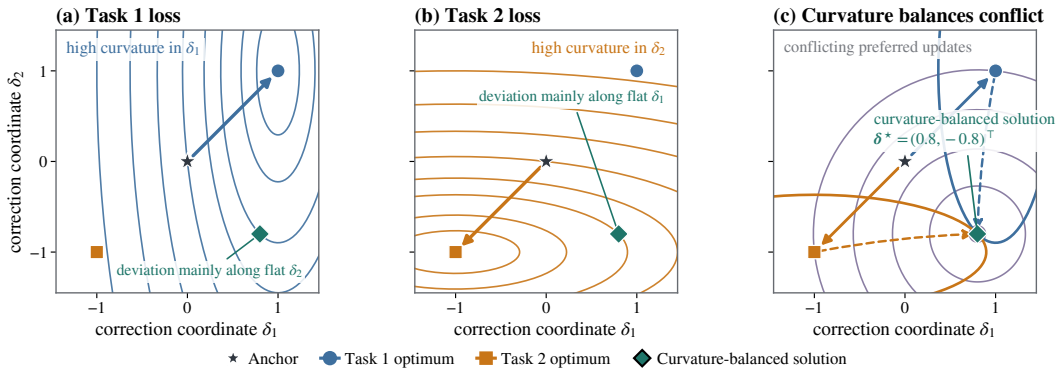


Figure 1: Curvature balances conflicting task preferences. For $F_t(\boldsymbol{\delta}) = (\boldsymbol{\delta} - \boldsymbol{\delta}_t^\star)^\top \boldsymbol{G}_t (\boldsymbol{\delta} - \boldsymbol{\delta}_t^\star)/2$, let $\boldsymbol{\delta}_1^\star = (1,1)^\top$, $\boldsymbol{\delta}_2^\star = (-1,-1)^\top$, $\boldsymbol{G}_1 = \mathrm{diag}(9,1)$, and $\boldsymbol{G}_2 = \mathrm{diag}(1,9)$. The average loss is minimized at $\boldsymbol{\delta}^\star = (0.8, -0.8)^\top$, with smaller deviations along each task's sensitive coordinate and larger ones along flatter directions.

We construct the local objective by linearizing each task output at the anchor and applying a second-order Taylor expansion to the resulting tangent loss (Jacot et al., 2018; Lee et al., 2019). Compared with directly expanding the original nonlinear loss, this construction replaces the exact parameter Hessian with the generalized Gauss–Newton curvature and avoids second derivatives of the nonlinear model output. When the loss is convex in its prediction argument, this curvature is positive semidefinite, yielding a convex quadratic surrogate rather than a potentially indefinite one.

Although convex, the surrogate remains a full parameter space problem. Solving the associated system is computationally expensive, and estimating its gradient and curvature from finite calibration data can yield a rank deficient or ill-conditioned empirical system. We therefore optimize the surrogate in a subspace spanned by a small number r of basis directions. This restriction reduces the number of free coefficients and the risk of overfitting, while yielding an $r \times r$ linear system with a closed-form solution. We construct the basis from merged updates produced by strong existing methods. Updates from multiple approaches can be included simultaneously, allowing the subspace to combine complementary information. The local loss objective determines how these directions are combined, while the subspace provides structural regularization and computational efficiency.

Our main contributions are summarized as follows:

- We formulate model merging as local optimization of the average supervised loss around a strong merged anchor and derive, using tangent linearization and generalized Gauss–Newton curvature, a quadratic surrogate that is convex when each task loss is convex in its prediction argument.
- We construct the optimization subspace from merged updates produced by strong existing methods, reducing the problem to an $r \times r$ linear system without forming a curvature matrix over all model parameters.
- Extensive experiments demonstrate broad applicability to existing merging methods and notable improvements over strong baselines across multiple model architectures and task collections. These gains remain stable with calibration subsets of moderate size, while the additional computational cost remains manageable.

2 Related Work

2.1 Merging based on parameters and structure

Weight averaging directly averages fine-tuned parameters (Wortsman et al., 2022; Ramé et al., 2023), whereas Task Arithmetic scales, adds, or negates updates from a shared pre-trained initialization (Ilharco et al., 2023). Subsequent methods modify these updates to reduce interference (Wang et al., 2024; Davari & Belilovsky, 2024; Deep et al., 2024; Huang et al., 2024). TIES-Merging removes

small updates and resolves sign conflicts, while DARE randomly drops and rescales update entries before merging (Yadav et al., 2023; Yu et al., 2024). WUDI-Merging treats the task vector in each linear layer as a proxy for its input subspace and suppresses interference measured within this space (Cheng et al., 2025). Spectral methods exploit the matrix structure of layer updates. TSV-Merge decomposes updates into singular directions to reduce interference among principal components (Gargiulo et al., 2025), while related methods align informative singular subspaces before merging (Stoica et al., 2025; Marczak et al., 2025). These methods require no task data, but their structural criteria do not directly reflect downstream losses.

2.2 Merging with auxiliary data

When calibration or proxy data are available, model behavior can guide merging beyond model parameters (Jordan et al., 2023; Stoica et al., 2024; Lu et al., 2024; Oh et al., 2025). At the output level, RegMean fits merged linear layers to match individual model outputs, whereas AdaMerging learns task or layer coefficients by minimizing prediction entropy (Jin et al., 2023; Yang et al., 2024b). At the representation level, Representation Surgery reduces feature discrepancies between merged and individual models, while ESM constructs essential subspaces from principal directions of activation changes (Yang et al., 2024a; Li et al., 2026). These methods use functional information from data, but optimize output, entropy, or representation proxies rather than supervised task losses.

2.3 Model linearization and loss geometry

Existing model merging methods use linearization during fine-tuning to obtain more compatible task updates, applying a tangent parameterization to either the full model or modules used for parameter efficient adaptation (Ortiz-Jiménez et al., 2023; Tang et al., 2024). Thus, linearization shapes task-specific updates before merging rather than approximating the loss around an existing merged model.

Other work combines fine-tuned models using local loss geometry, including Fisher-weighted averaging and matching along directions to which task losses are locally sensitive (Matena & Raffel, 2022; Tam et al., 2024), as well as gradient mismatch or Hessian geometry (Daheim et al., 2024; Lu et al., 2026). These methods evaluate local quantities separately at the fine-tuned solutions.

Our method instead applies the tangent approximation after fine-tuning, centered at a merged anchor. This preserves the compromise already encoded by the merged model, while task gradients and generalized Gauss–Newton curvatures characterize the effects of a shared correction. The resulting surrogate locally approximates the average loss, enabling joint optimization of the task errors.

3 Methodology

3.1 Tangent loss at the anchor

Let $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$ denote a model backbone parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, and let $\boldsymbol{\theta}_0$ denote its pre-trained parameters. Consider T downstream tasks indexed by $[T] = \{1, \dots, T\}$. For each $t \in [T]$, fine-tuning the pre-trained backbone on task t yields parameters $\boldsymbol{\theta}_t$. The task update is $\boldsymbol{\Delta}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$ for $t \in [T]$. Our goal is to construct a single merged backbone that performs well across all T tasks without joint retraining on their combined data.

For task t, let $(\boldsymbol{X}, Y) \sim P_t$, where $\boldsymbol{X} \in \mathcal{X}$ is the input, $Y \in \mathcal{Y}_t$ is the corresponding target, and P_t is their joint distribution. During merging, each task retains a fixed output head $h_t : \mathbb{R}^d \to \mathbb{R}^{q_t}$, where q_t is the output dimension. The resulting model output is $\boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}) = h_t\big(f_{\boldsymbol{\theta}}(\boldsymbol{X})\big)$. Let $\ell_t : \mathbb{R}^{q_t} \times \mathcal{Y}_t \to \mathbb{R}$ denote the loss on an individual example, such as mean squared error or cross-entropy. The population loss for task t is

$$F_t(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t} \ell_t\big(\boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}), Y\big).$$

To evaluate a model jointly across tasks, we define the average population loss $\overline{F}(\boldsymbol{\theta}) = T^{-1} \sum_{t=1}^{T} F_t(\boldsymbol{\theta})$. Directly optimizing $\overline{F}$ over all model parameters would amount to joint retraining. We instead construct a local surrogate around an anchor model. Let $\boldsymbol{\Delta}_A$ denote an anchor update and define the corresponding anchor model as $\boldsymbol{\theta}_A = \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_A$. For example, $\boldsymbol{\Delta}_A$ may be

produced by an existing model merging method. Starting from $\boldsymbol{\theta}_A$, we seek a perturbation $\boldsymbol{\delta} \in \mathbb{R}^p$ that reduces the average task loss. For task t, the loss at the perturbed model is

$$F_t(\boldsymbol{\theta}_A + \boldsymbol{\delta}) = \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t} \ell_t(\boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}_A + \boldsymbol{\delta}), Y)\,.$$

We linearize the model output at the anchor using the tangent approximation associated with the neural tangent kernel:

$$\boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}_A + \boldsymbol{\delta}) \approx \boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}_A) + \boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{\delta}, \tag{1}$$

where the parameter Jacobian at the anchor is $\boldsymbol{J}_{t,A}(\boldsymbol{X}) = \nabla_{\boldsymbol{\theta}} \boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_A}$. Substituting (1) into the task loss gives the tangent loss

$$\widetilde{F}_t(\boldsymbol{\delta}) := \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t} \ell_t(\boldsymbol{z}_t(\boldsymbol{X}; \boldsymbol{\theta}_A) + \boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{\delta}, Y)\,,$$

which approximates $F_t(\boldsymbol{\theta}_A + \boldsymbol{\delta})$ by linearizing the dependence of the model output on $\boldsymbol{\delta}$ at $\boldsymbol{\theta}_A$. By construction, $\widetilde{F}_t(\mathbf{0}) = F_t(\boldsymbol{\theta}_A)$ and $\nabla_{\boldsymbol{\delta}} \widetilde{F}_t(\mathbf{0}) = \nabla_{\boldsymbol{\theta}} F_t(\boldsymbol{\theta}_A)$. Thus, the tangent loss agrees with the original task loss in value and first derivative at the anchor.

3.2 Quadratic approximation and local merging objective

Directly minimizing the tangent loss would generally require an iterative solver. We instead obtain a local quadratic objective by expanding $\widetilde{F}_t$ around $\boldsymbol{\delta} = \mathbf{0}$:

$$\widetilde{F}_t(\boldsymbol{\delta}) = F_t(\boldsymbol{\theta}_A) + \boldsymbol{g}_{t,A}^\top \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{G}_{t,A}\boldsymbol{\delta} + r_t(\boldsymbol{\delta}), \tag{2}$$

where

$$\begin{aligned}
\boldsymbol{g}_{t,A} &= \nabla_{\boldsymbol{\delta}} \widetilde{F}_t(\boldsymbol{\delta})\Big|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t}\left[\boldsymbol{J}_{t,A}(\boldsymbol{X})^\top \boldsymbol{s}_{t,A}(\boldsymbol{X}, Y)\right], \\
\boldsymbol{G}_{t,A} &= \nabla^2_{\boldsymbol{\delta}} \widetilde{F}_t(\boldsymbol{\delta})\Big|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t}\left[\boldsymbol{J}_{t,A}(\boldsymbol{X})^\top \boldsymbol{W}_{t,A}(\boldsymbol{X}, Y)\boldsymbol{J}_{t,A}(\boldsymbol{X})\right],
\end{aligned}$$

with $\boldsymbol{s}_{t,A}(\boldsymbol{X}, Y) = \nabla_{\boldsymbol{z}} \ell_t(\boldsymbol{z}, Y)|_{\boldsymbol{z}=\boldsymbol{z}_t(\boldsymbol{X};\boldsymbol{\theta}_A)}$ and $\boldsymbol{W}_{t,A}(\boldsymbol{X}, Y) = \nabla^2_{\boldsymbol{z}} \ell_t(\boldsymbol{z}, Y)|_{\boldsymbol{z}=\boldsymbol{z}_t(\boldsymbol{X};\boldsymbol{\theta}_A)}$. Here, $\boldsymbol{s}_{t,A}$ and $\boldsymbol{W}_{t,A}$ are the gradient and Hessian of the individual loss with respect to the model output. Accordingly, $\boldsymbol{g}_{t,A}^\top \boldsymbol{\delta}$ describes the first-order loss change, $\boldsymbol{\delta}^\top \boldsymbol{G}_{t,A}\boldsymbol{\delta}/2$ describes its local curvature, and $r_t(\boldsymbol{\delta})$ collects terms beyond quadratic order.

The matrix $\boldsymbol{G}_{t,A}$ is the generalized Gauss–Newton (GGN) curvature (Martens, 2020). It maps the curvature of the loss in the output space to the parameter space through $\boldsymbol{J}_{t,A}$ and does not involve second derivatives of the model output. In contrast, directly expanding the original task loss gives

$$\nabla^2_{\boldsymbol{\theta}} F_t(\boldsymbol{\theta}_A) = \boldsymbol{G}_{t,A} + \mathbb{E}_{(\boldsymbol{X},Y)\sim P_t}\left[\sum_{k=1}^{q_t} s_{t,A,k}(\boldsymbol{X}, Y)\nabla^2_{\boldsymbol{\theta}} z_{t,k}(\boldsymbol{X}; \boldsymbol{\theta}_A)\right].$$

The additional term captures nonlinear model curvature, requires second model derivatives, and may make the exact Hessian indefinite. It is absent from the tangent model because its output is affine in $\boldsymbol{\delta}$. If ℓ_t is convex in its prediction argument, then $\boldsymbol{W}_{t,A}$ and $\boldsymbol{G}_{t,A}$ are positive semidefinite, making the quadratic surrogate convex. GGN matrix–vector products can be evaluated as $\boldsymbol{v} \mapsto \boldsymbol{J}_{t,A}^\top \boldsymbol{W}_{t,A}\boldsymbol{J}_{t,A}\boldsymbol{v}$ without forming the full curvature matrix or differentiating the model output twice.

Figure 2 illustrates this distinction in a two-dimensional squared loss example. The direct Taylor surrogate inherits negative curvature, whereas the tangent surrogate remains convex and is minimized near a local minimum of the original loss.

Averaging (2) across tasks and omitting the remainder terms give the local merging objective

$$Q_A(\boldsymbol{\delta}) := \frac{1}{T}\sum_{t=1}^{T}\left\{\boldsymbol{g}_{t,A}^\top \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{G}_{t,A}\boldsymbol{\delta}\right\}. \tag{3}$$

3.3 Optimization in a low-dimensional subspace

The objective $Q_A(\boldsymbol{\delta})$ is defined over all p model parameters. For modern networks, storing the aggregate curvature requires $\mathcal{O}(p^2)$ memory, and solving the dense system requires $\mathcal{O}(p^3)$ operations.

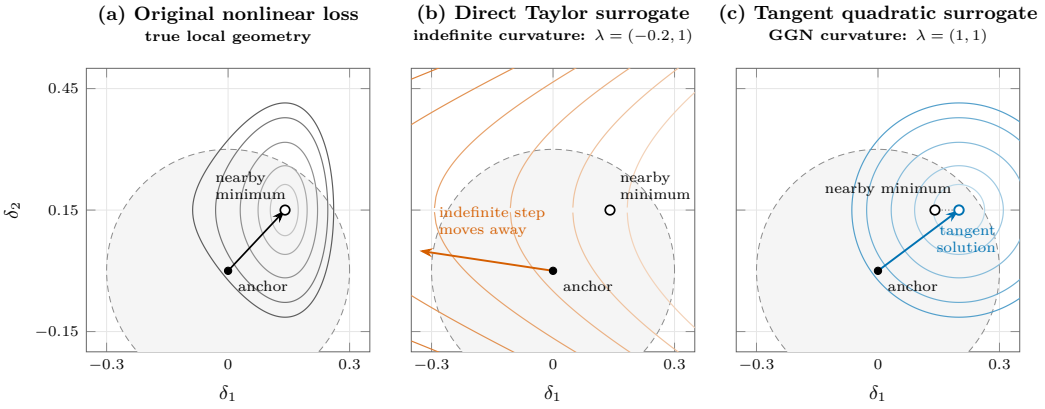


Figure 2: Local loss geometry for $\boldsymbol{z}(\boldsymbol{\delta}) = (\delta_1 + 3\delta_1^2, \delta_2)^\top$, target $\boldsymbol{y} = (0.2, 0.15)^\top$, and half the squared Euclidean loss. At $\boldsymbol{\delta} = \mathbf{0}$, the exact Hessian is $\mathrm{diag}(-0.2, 1)$, whereas the GGN is $\boldsymbol{I}_2$. The indefinite direct Taylor surrogate has a stationary point at $(-1, 0.15)^\top$, while the tangent surrogate is minimized at $(0.2, 0.15)^\top$, near the original loss minimum $(0.141, 0.15)^\top$. The dashed circle denotes $\|\boldsymbol{\delta}\|_2 \leq 0.3$.

Moreover, estimating p correction coefficients from limited calibration data may fit estimation noise and generalize poorly.

We therefore restrict the correction to a prescribed subspace. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r) \in \mathbb{R}^{p \times r}$ ($r \ll p$) collect the allowed parameter directions. In our implementation, for each merged update produced by a strong existing method, we separate selected Transformer block components and group all remaining parameter entries into one source-specific Rest direction. This preserves the structure encoded by these methods while allowing the selected block components and the Rest direction of each source to be adjusted separately. We parameterize the correction as

$$\boldsymbol{\delta} = \boldsymbol{B}\boldsymbol{\eta}, \qquad \boldsymbol{\theta}(\boldsymbol{\eta}) = \boldsymbol{\theta}_A + \boldsymbol{B}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} \in \mathbb{R}^r.$$

Thus, $\boldsymbol{\eta} = \mathbf{0}$ recovers the anchor. The restriction reduces the free coefficients from p to r and provides structural regularization through the basis directions. Substituting $\boldsymbol{\delta} = \boldsymbol{B}\boldsymbol{\eta}$ into (3) gives

$$Q_A(\boldsymbol{B}\boldsymbol{\eta}) = \frac{1}{T}\sum_{t=1}^{T}\left\{\boldsymbol{g}_{t,A}^\top \boldsymbol{B}\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\eta}^\top \boldsymbol{B}^\top \boldsymbol{G}_{t,A}\boldsymbol{B}\boldsymbol{\eta}\right\}.$$

When ℓ_t is convex in its prediction argument, each $\boldsymbol{G}_{t,A}$ is positive semidefinite, and hence the restricted objective is convex. The projected gradient and curvature can be computed directly as

$$\boldsymbol{B}^\top \boldsymbol{g}_{t,A} = \mathbb{E}_{P_t}\left[\big(\boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{B}\big)^\top \boldsymbol{s}_{t,A}(\boldsymbol{X}, Y)\right],$$

$$\boldsymbol{B}^\top \boldsymbol{G}_{t,A}\boldsymbol{B} = \mathbb{E}_{P_t}\left[\big(\boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{B}\big)^\top \boldsymbol{W}_{t,A}(\boldsymbol{X}, Y)\big(\boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{B}\big)\right].$$

Thus, only the directional Jacobians $\boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{B}$ and an $r \times r$ curvature matrix are required. The curvature storage and dense solution costs are reduced to $\mathcal{O}(r^2)$ and $\mathcal{O}(r^3)$, respectively.

For implementation, let $\mathcal{C}_t = \{(\boldsymbol{X}_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$ denote the calibration set for task t, and write $\boldsymbol{J}_{t,A,i} = \boldsymbol{J}_{t,A}(\boldsymbol{X}_{t,i})$, $\boldsymbol{s}_{t,A,i} = \boldsymbol{s}_{t,A}(\boldsymbol{X}_{t,i}, Y_{t,i})$, and $\boldsymbol{W}_{t,A,i} = \boldsymbol{W}_{t,A}(\boldsymbol{X}_{t,i}, Y_{t,i})$. The empirical gradient and curvature in coefficient space are

$$\widehat{\boldsymbol{g}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\big(\boldsymbol{J}_{t,A,i}\boldsymbol{B}\big)^\top \boldsymbol{s}_{t,A,i}, \qquad \widehat{\boldsymbol{G}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\big(\boldsymbol{J}_{t,A,i}\boldsymbol{B}\big)^\top \boldsymbol{W}_{t,A,i}\big(\boldsymbol{J}_{t,A,i}\boldsymbol{B}\big). \quad (4)$$

The inner averages assign equal weight to examples within each task, while the outer average assigns equal weight to all tasks regardless of their calibration set sizes.

Because $\widehat{\boldsymbol{G}}$ may be singular or ill-conditioned when estimated from finite calibration data, we add ridge regularization in coefficient space:

$$\widehat{Q}_\lambda(\boldsymbol{\eta}) = \widehat{\boldsymbol{g}}^\top \boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\eta}^\top \widehat{\boldsymbol{G}}\boldsymbol{\eta} + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_2^2.$$

Algorithm 1 Local loss optimization for model merging

Require: Pre-trained parameters $\boldsymbol{\theta}_0$; anchor update $\boldsymbol{\Delta}_A$; directions $\{\boldsymbol{b}_j\}_{j=1}^r$; calibration sets $\mathcal{C}_t = \{(\boldsymbol{X}_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$; task heads and losses $\{(h_t, \ell_t)\}_{t=1}^T$; ridge candidates $\mathcal{K} \subset (0, \infty)$; validation data

Ensure: Merged update $\boldsymbol{\Delta}_M$ and model $\boldsymbol{\theta}_M$

$\boldsymbol{\theta}_A \leftarrow \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_A$, $\boldsymbol{B} \leftarrow (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r)$, $\widehat{\boldsymbol{g}} \leftarrow \boldsymbol{0}_r$, $\widehat{\boldsymbol{G}} \leftarrow \boldsymbol{0}_{r \times r}$
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, n_t$ **do**
    Evaluate $\boldsymbol{s}_{t,A,i}$ and $\boldsymbol{W}_{t,A,i}$ at $\boldsymbol{z}_t(\boldsymbol{X}_{t,i}; \boldsymbol{\theta}_A)$; compute $\boldsymbol{D}_{t,A,i} \leftarrow \boldsymbol{J}_{t,A}(\boldsymbol{X}_{t,i})\boldsymbol{B}$
    $\widehat{\boldsymbol{g}} \leftarrow \widehat{\boldsymbol{g}} + (Tn_t)^{-1}\boldsymbol{D}_{t,A,i}^\top \boldsymbol{s}_{t,A,i}$
    $\widehat{\boldsymbol{G}} \leftarrow \widehat{\boldsymbol{G}} + (Tn_t)^{-1}\boldsymbol{D}_{t,A,i}^\top \boldsymbol{W}_{t,A,i}\boldsymbol{D}_{t,A,i}$
  end for
end for
Verify $\operatorname{tr}(\widehat{\boldsymbol{G}}) > 0$
for $\kappa \in \mathcal{K}$ **do**
  $\lambda_\kappa \leftarrow \kappa \operatorname{tr}(\widehat{\boldsymbol{G}})/r$; solve $\big(\widehat{\boldsymbol{G}} + \lambda_\kappa \boldsymbol{I}_r\big)\widehat{\boldsymbol{\eta}}_\kappa^\star = -\widehat{\boldsymbol{g}}$
end for
Select $\kappa^\star$ by validating $\boldsymbol{\theta}_A + \boldsymbol{B}\widehat{\boldsymbol{\eta}}_\kappa^\star$ and set $\widehat{\boldsymbol{\eta}}^\star \leftarrow \widehat{\boldsymbol{\eta}}_{\kappa^\star}^\star$
$\boldsymbol{\Delta}_M \leftarrow \boldsymbol{\Delta}_A + \boldsymbol{B}\widehat{\boldsymbol{\eta}}^\star$
$\boldsymbol{\theta}_M \leftarrow \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_M$
return $\boldsymbol{\Delta}_M, \boldsymbol{\theta}_M$

Assuming $\operatorname{tr}(\widehat{\boldsymbol{G}}) > 0$, we set $\lambda_\kappa = \kappa \operatorname{tr}(\widehat{\boldsymbol{G}})/r$ for $\kappa > 0$. Thus $\lambda_\kappa > 0$, while κ remains relative to the average curvature in coefficient space. The value of κ is selected on the validation split. The minimizer satisfies

$$\left(\widehat{\boldsymbol{G}} + \lambda_\kappa \boldsymbol{I}_r\right) \widehat{\boldsymbol{\eta}}_\kappa^\star = -\widehat{\boldsymbol{g}}.$$

The regularizer acts on $\boldsymbol{\eta}$, shrinking the correction toward the anchor rather than the merged update toward zero. Let $\widehat{\boldsymbol{\eta}}^\star$ denote the solution. The anchor update $\boldsymbol{\Delta}_A$ includes the scale of its source method, so no additional global scaling is applied. The resulting merged update and model are

$$\boldsymbol{\Delta}_M = \boldsymbol{\Delta}_A + \boldsymbol{B}\widehat{\boldsymbol{\eta}}^\star, \qquad \boldsymbol{\theta}_M = \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_M.$$

Algorithm 1 summarizes our method.

4 Experiments

4.1 Fully fine-tuned vision models

The evaluation covers suites of 8, 14, and 20 image classification tasks across three CLIP variants with ViT-B/32, ViT-B/16, and ViT-L/14 image encoders (Radford et al., 2021). The task-specific fully fine-tuned checkpoints were originally provided by TALL-masks (Wang et al., 2024) and distributed through the TSV-M repository (Gargiulo et al., 2025). Their output heads remain fixed throughout merging and evaluation.

Following the protocol commonly adopted for this benchmark (Gargiulo et al., 2025; Li et al., 2026), we use the validation and test splits implemented in TSV-M. The validation split selects the regularization parameter κ; the test split is evaluated once thereafter. We sample 2,000 labeled examples per task from the training split to form calibration sets for estimating the local gradient and curvature. Further dataset and split details are provided in Appendix C.

We compare with Weight Averaging (Wortsman et al., 2022), Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), Consensus TA (Wang et al., 2024), TSV-M (Gargiulo et al., 2025), Iso-C and Iso-CTS (Marczak et al., 2025), and ESM (Li et al., 2026). We also report the zero-shot model and individual fine-tuned models as lower and upper reference points, respectively. Performance is measured by macro accuracy and normalized macro accuracy. The latter averages, across tasks, the ratio of merged model accuracy to the corresponding fine-tuned accuracy. Both metrics are reported as percentages.

Table 1: Multi-task merging results on three CLIP backbones. Entries for merged models report mean test accuracy, with mean normalized accuracy as a parenthesized subscript; both are percentages. The zero-shot and fine-tuned rows report accuracy only. Our methods are shown in bold.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Zero-shot	48.3	57.2	56.1	55.3	61.3	59.7	64.7	68.2	65.2
Fine-tuned	92.8	90.9	91.3	94.6	92.8	93.2	95.8	94.3	94.7
Weight Averaging	$66.3_{(72.1)}$	$64.3_{(71.1)}$	$61.0_{(67.5)}$	$72.2_{(76.6)}$	$69.5_{(74.8)}$	$65.3_{(70.4)}$	$79.6_{(83.2)}$	$76.7_{(81.1)}$	$71.6_{(75.6)}$
Task Arithmetic	$70.8_{(76.5)}$	$65.3_{(72.1)}$	$60.5_{(66.8)}$	$75.4_{(79.6)}$	$70.5_{(75.9)}$	$65.8_{(70.8)}$	$84.9_{(88.7)}$	$79.4_{(84.0)}$	$74.0_{(78.1)}$
TIES-Merging	$75.1_{(81.0)}$	$68.0_{(74.8)}$	$63.4_{(69.9)}$	$79.7_{(84.3)}$	$73.2_{(78.7)}$	$68.2_{(73.3)}$	$86.9_{(90.7)}$	$79.5_{(84.1)}$	$75.7_{(79.8)}$
Consensus TA	$75.0_{(80.8)}$	$70.4_{(77.4)}$	$65.4_{(72.0)}$	$79.4_{(83.9)}$	$74.4_{(79.9)}$	$69.8_{(74.9)}$	$86.3_{(90.1)}$	$82.2_{(86.9)}$	$79.0_{(83.2)}$
TSV-M	$85.9_{(92.3)}$	$80.1_{(87.9)}$	$77.1_{(84.3)}$	$89.0_{(93.9)}$	$84.6_{(91.0)}$	$80.6_{(86.5)}$	$93.0_{(97.0)}$	$89.2_{(94.4)}$	$87.7_{(92.5)}$
Iso-C	$86.3_{(92.9)}$	$80.3_{(88.1)}$	$75.5_{(82.5)}$	$90.6_{(95.6)}$	$84.8_{(91.1)}$	$79.6_{(85.4)}$	$94.2_{(98.3)}$	$89.3_{(94.5)}$	$87.6_{(92.2)}$
Iso-CTS	$86.2_{(92.8)}$	$81.7_{(89.7)}$	$78.1_{(85.5)}$	$91.1_{(96.1)}$	$86.4_{(92.8)}$	$82.4_{(88.4)}$	$94.7_{(98.8)}$	$91.0_{(96.3)}$	$90.1_{(94.9)}$
LossOpt (I)	$\mathbf{87.9_{(94.7)}}$	$\mathbf{83.8_{(92.1)}}$	$\mathbf{81.0_{(88.7)}}$	$\mathbf{91.8_{(96.9)}}$	$\mathbf{87.7_{(94.3)}}$	$\mathbf{84.4_{(90.5)}}$	$\mathbf{94.8_{(98.9)}}$	$\mathbf{91.4_{(96.8)}}$	$\mathbf{90.7_{(95.6)}}$
ESM	$88.4_{(95.3)}$	$83.7_{(92.0)}$	$81.3_{(88.9)}$	$91.8_{(97.0)}$	$87.4_{(94.1)}$	$84.9_{(91.1)}$	$94.8_{(98.9)}$	$91.3_{(96.8)}$	$90.4_{(95.3)}$
LossOpt (T+I+E)	$\mathbf{89.1_{(95.9)}}$	$\mathbf{85.4_{(93.9)}}$	$\mathbf{83.4_{(91.3)}}$	$\mathbf{92.4_{(97.6)}}$	$\mathbf{88.6_{(95.4)}}$	$\mathbf{86.4_{(92.7)}}$	$\mathbf{95.0_{(99.1)}}$	$\mathbf{91.9_{(97.4)}}$	$\mathbf{91.6_{(96.7)}}$

4.2 Anchor and subspace construction

ViT-B/32 and ViT-B/16 contain 12 Transformer blocks, whereas ViT-L/14 contains 24; all blocks are numbered from 1. Let $\mathcal{S}$ denote the source merging methods. For each $s \in \mathcal{S}$, let $\boldsymbol{\Delta}_s$ be its merged update and $\mathcal{J}_s$ index the blocks used as individual directions. For $j \in \mathcal{J}_s$, define $\boldsymbol{b}_{s,j} = \mathcal{P}_j(\boldsymbol{\Delta}_s)$, where $\mathcal{P}_j$ retains only the entries in block j. All remaining entries form the Rest direction $\boldsymbol{b}_{s,\mathrm{rem}} = \boldsymbol{\Delta}_s - \sum_{j\in\mathcal{J}_s} \boldsymbol{b}_{s,j}$. Collecting these directions as the columns of $\boldsymbol{B}$, the correction is

$$\boldsymbol{\delta} = \boldsymbol{B}\boldsymbol{\eta} = \sum_{s\in\mathcal{S}} \left(\sum_{j\in\mathcal{J}_s} \eta_{s,j}\boldsymbol{b}_{s,j} + \eta_{s,\mathrm{rem}}\boldsymbol{b}_{s,\mathrm{rem}} \right).$$

The coefficients jointly scale the selected blocks and Rest direction from each source.

The single source configuration uses the complete Iso-CTS update as its anchor. Preliminary validation ablations indicated that the final six blocks generally had the greatest effect on accuracy, so these blocks are selected for all three CLIP variants. The multi-source configuration combines Task Arithmetic, Iso-CTS, and ESM directions using the complete ESM update as its anchor. ViT-B/32 and ViT-B/16 use the final six blocks for all sources. For ViT-L/14, Task Arithmetic and Iso-CTS use the final six blocks, whereas ESM uses blocks 9–14. Further block selection and validation settings are reported in Appendix C.

4.3 Main results

We call our method LossOpt, denoting loss-aware optimization in a low-dimensional subspace. Its single- and multi-source configurations are denoted by LossOpt (I) and LossOpt (T+I+E), where T, I, and E represent Task Arithmetic, Iso-CTS, and ESM, respectively. Table 1 reports all nine combinations of backbone and task suite. The results of existing methods in Table 1 are taken directly from the values reported in the ESM paper (Li et al., 2026).

LossOpt (T+I+E) outperforms the strong ESM baseline in all nine settings. The LossOpt (I) configuration also improves its Iso-CTS anchor throughout and generally achieves performance comparable to ESM despite using directions from only one source. For both configurations, the gains increase with the number of tasks for every backbone, indicating that loss-aware correction becomes more beneficial as task interference grows. The largest gains occur on ViT-B/32, whose lower anchor accuracy leaves greater room for correction.

4.4 Effect of calibration set size

We assess sensitivity to the amount of labeled calibration data using LossOpt (T+I+E) on ViT-B/32 with fourteen tasks. Table 2 reports runtime and test performance. All runtimes are measured on a

Table 2: Effect of calibration set size for LOSSOPT (T+I+E) on ViT-B/32 with fourteen tasks. Times are reported as `m:ss`, and both test metrics are percentages.

	Time (`m:ss`)				Test performance	
Examples/task	Deriv./solve	Validation	Test	Core total	Acc.	Norm.
500	7:44	4:41	6:48	19:13	85.392	93.894
1000	10:32	4:58	6:49	22:19	85.316	93.813
2000	20:01	4:53	6:39	31:33	85.376	93.877
3000	21:46	4:44	6:47	33:17	85.383	93.886

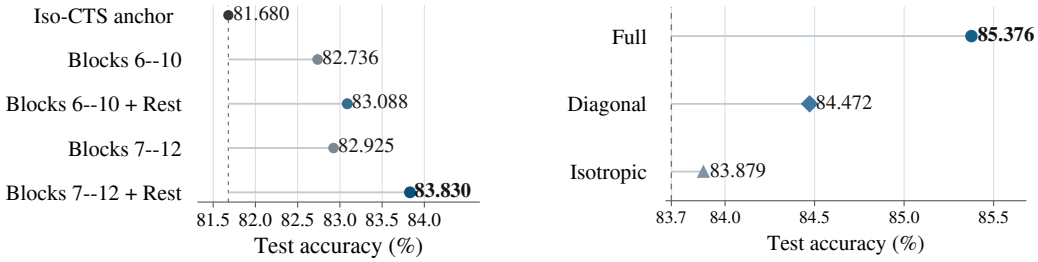


Figure 3: Ablation studies on ViT-B/32 with fourteen tasks. Left: basis direction ablation of LOSSOPT (I), where Rest contains all Iso-CTS update entries outside the listed Transformer blocks. Right: curvature ablation for LOSSOPT (T+I+E).

single NVIDIA GeForce RTX 4090 GPU. Deriv./solve includes derivative estimation and the linear solve. Validation covers model selection. Test includes constructing the final merged model and evaluating it, while Core total sums the recorded stages. Increasing the calibration budget mainly raises Deriv./solve time, whereas relatively small sets already yield stable test performance.

4.5 ABLATION STUDIES

We conduct two ablation studies on ViT-B/32 with fourteen tasks. First, the basis direction ablation evaluates how the choice of directions affects the LOSSOPT (I) configuration. Second, within LOSSOPT (T+I+E), the curvature ablation compares the full curvature matrix with two simplified approximations. The isotropic approximation replaces the projected GGN with $\mathrm{tr}(\widehat{\boldsymbol{G}})\boldsymbol{I}_r/r$, while the diagonal approximation retains its diagonal entries. The results show that separating influential blocks while retaining the remaining source updates as the Rest direction improves performance. The full curvature performs best, demonstrating the value of interactions among basis directions.

4.6 LANGUAGE MODEL MERGING

We further evaluate LOSSOPT with RoBERTa-Base on eight GLUE tasks following the WUDI protocol (Cheng et al., 2025): CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. Only the shared encoder parameters are merged; the task-specific heads remain fixed. The final 10% of each training set is used for validation, and $1{,}000$ randomly sampled training examples per task for calibration. We average each task's standard metric normalized by its corresponding single task model score.

Because GLUE includes both classification and regression tasks with substantially different objective scales, we normalize each task objective by its value at the anchor:

$$\widetilde{\mathcal{L}}(\boldsymbol{\eta}) = \frac{1}{8}\left[\sum_{m\neq\text{STS-B}} \frac{\mathrm{CE}_m(\boldsymbol{\eta})}{\mathrm{CE}_m(\mathbf{0})} + \frac{1-r_{\text{STS-B}}(\boldsymbol{\eta})}{1-r_{\text{STS-B}}(\mathbf{0})}\right],$$

where CE_m denotes the cross-entropy loss for classification task m. STS-B denotes the Semantic Textual Similarity Benchmark, the regression task, and $r_{\text{STS-B}}$ is its Pearson correlation. For STS-B, we compute the Hessian of its dataset level Pearson loss on the tangent surrogate by automatic

Table 3: RoBERTa-Base merging results. Mean normalized scores are reported as percentages.

Method	Score	Method	Score	Method	Score
Zero-shot	41.70	TIES-Merging	64.70	Iso-CTS	82.24
Fine-tuned	100.00	Task Arithmetic w/ DARE	63.70	ESM	89.11
Weight Averaging	52.60	TIES-Merging w/ DARE	65.60	**LOSSOPT (W+I+E)**	**90.85**
Task Arithmetic	67.80	WUDI-Merging	85.34		

Table 4: Ablation of basis sources for LOSSOPT on RoBERTa-Base. W, I, and E denote WUDI-Merging, Iso-CTS, and ESM, respectively. Dimension is the number of basis directions.

Basis sources	E	W+E	I+E	W+I	**W+I+E**
Dimension	7	14	14	14	21
Mean norm. score (%)	89.70	89.97	90.04	89.81	**90.85**

differentiation and use it in the same second-order Taylor model; this exception is outside the per-example GGN positive semidefinite (PSD) and convexity guarantees.

Table 3 shows that LOSSOPT (W+I+E) achieves a mean normalized score of 90.85, outperforming the strongest baseline, ESM, by 1.74 points. Table 4 further ablates the basis sources. ESM directions alone improve the anchor, while adding WUDI-Merging or Iso-CTS yields further gains. Combining all three sources performs best, exceeding the strongest configuration using two sources by 0.81 points and demonstrating their complementarity.

To assess the quadratic approximation, we compare LOSSOPT with direct tangent loss optimization using the same ESM anchor and 21-dimensional W+I+E subspace. Figure 4 shows that direct optimization achieves a higher validation score, whereas LOSSOPT achieves a higher test score, suggesting that the quadratic approximation provides implicit local regularization in addition to a closed-form solution.

5 CONCLUSION

We introduced a loss-aware framework for improving an existing merged model by locally optimizing the average supervised loss. Linearizing task outputs at a merged anchor and quadratically expanding the tangent losses yields a GGN-based quadratic surrogate that is convex for losses convex in their prediction argument. Restricting the correction to directions from strong merged updates reduces the problem to a small regularized linear system and allows directions from multiple methods to be optimized jointly. Experiments show that the correction improves strong single source anchors, with complementary source directions generally providing further gains.

The method uses labeled calibration examples to estimate the local gradient and curvature. Since the search is restricted to directions from existing merged updates, improvements may be limited when the anchor is weak or useful corrections lie outside the selected subspace. Future work could use calibration data to select correction directions adaptively.

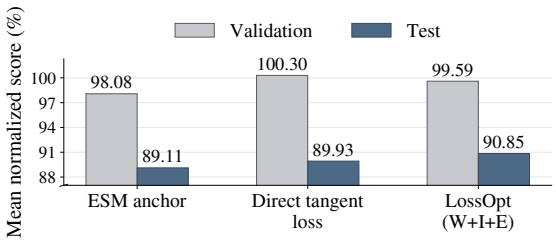


Figure 4: Comparison between direct tangent loss and LOSSOPT (W+I+E) on RoBERTa-Base.

AI USE STATEMENT

Generative AI tools were used to assist with English language editing, preparation of illustrative figures, code development, and checking mathematical derivations. The research problem, core method, experimental design and execution, and final scientific conclusions were developed and determined by the authors. All AI-assisted material was reviewed and verified by the authors, who take full responsibility for the final content of the paper.

ETHICS STATEMENT

We are not aware of any ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

Appendix C documents the datasets and splits, checkpoint sources, calibration set construction, model selection settings, evaluation protocol, and hardware used in our experiments to support reproducibility.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7:195–204, 2025. doi: 10.1038/s42256-024-00975-8. URL https://www.nature.com/articles/s42256-024-00975-8.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10121–10143. PMLR, 2025. URL https://proceedings.mlr.press/v267/cheng25h.html.

Nico Daheim, Thomas Möllenhoff, Edoardo M. Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=D7KJmfEDQP.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *Computer Vision – ECCV 2024*, pp. 270–287. Springer, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/9596_ECCV_2024_paper.php.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. DELLA-Merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024. URL https://arxiv.org/abs/2406.11617.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18695–18705, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. EMR-Merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/dda5cac5272a9bcd4bc73d90bc725ef1-Abstract-Conference.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Xisen Jin, Xiang Ren, Daniel Preoţiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: Renormalizing permuted activations for interpolation repair. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gU5sJ6ZggcX.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model merging in the essential subspace. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31145–31154, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_Model_Merging_in_the_Essential_Subspace_CVPR_2026_paper.html.

Juanwu Lu, Anand Bhaskar, Brian Axelrod, Ekaterina Tolstaya, and Tristan Emrich. Model merging on loss landscape: A geometry perspective. *arXiv preprint arXiv:2605.26693*, 2026. URL https://arxiv.org/abs/2605.26693.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems*, volume 37, pp. 78905–78935, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8fcd17eb91bae20d9826786d7d6be799-Abstract-Conference.html.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost Van De Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43177–43199. PMLR, 2025. URL https://proceedings.mlr.press/v267/marczak25a.html.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL https://jmlr.org/papers/v21/17-678.html.

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/70c26937fbf3d4600b69a129031b66ec-Abstract-Conference.html.

Changdae Oh, Yixuan Li, Kyungwoo Song, Sangdoo Yun, and Dongyoon Han. DaWin: Training-free dynamic weight interpolation for robust adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Abstract-Conference.html.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d28077e5ff52034cd35b4aa15320caea-Abstract-Conference.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28656–28679, 2023. URL https://proceedings.mlr.press/v202/rame23a.html.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb2697869f56484404c8ceee2985b01d-Abstract.html.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. ZipIt! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LEYUkvdUhq.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the KnOTS. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=67X93aZHII.

Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=qNGo6ghWFB.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iynRvVVAmH.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50268–50287, 2024. URL https://proceedings.mlr.press/v235/wang24k.html.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115, 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56332–56356. PMLR, 2024a. URL https://proceedings.mlr.press/v235/yang24t.html.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=nZP6NgD3QY.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

A Theoretical properties

A.1 Descent of the original loss

Define the aggregate gradient and curvature in coefficient space as

$$\boldsymbol{g} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{B}^\top \boldsymbol{g}_{t,A}, \qquad \boldsymbol{G} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{B}^\top \boldsymbol{G}_{t,A}\boldsymbol{B}.$$

Then $Q_A(\boldsymbol{B}\boldsymbol{\eta}) = \boldsymbol{g}^\top\boldsymbol{\eta} + \boldsymbol{\eta}^\top\boldsymbol{G}\boldsymbol{\eta}/2$. Their empirical counterparts are denoted by $\widehat{\boldsymbol{g}}$ and $\widehat{\boldsymbol{G}}$.

We now show that the correction obtained from the regularized tangent quadratic surrogate is a strict descent direction for the original nonlinear average loss at the anchor whenever it is nonzero.

Proposition A.1 (Descent of the original loss)**.** *Suppose $\overline{F}(\boldsymbol{\theta}) = T^{-1}\sum_{t=1}^{T} F_t(\boldsymbol{\theta})$ is differentiable at $\boldsymbol{\theta}_A$ and $\boldsymbol{G} \succeq \boldsymbol{0}$. For a fixed $\lambda > 0$, let $\boldsymbol{\eta}_\lambda^\star$ minimize $Q_A(\boldsymbol{B}\boldsymbol{\eta}) + \lambda\|\boldsymbol{\eta}\|_2^2/2$. Then there exists $\gamma_0 > 0$ such that*

$$\overline{F}\big(\boldsymbol{\theta}_A + \gamma\boldsymbol{B}\boldsymbol{\eta}_\lambda^\star\big) \le \overline{F}(\boldsymbol{\theta}_A),$$

for every $\gamma \in (0, \gamma_0]$. The inequality is strict whenever $\boldsymbol{g} \neq \boldsymbol{0}$.

Proof. Since $\boldsymbol{G} \succeq \boldsymbol{0}$ and $\lambda > 0$, the regularized objective is strictly convex, with unique minimizer

$$\boldsymbol{\eta}_\lambda^\star = -(\boldsymbol{G} + \lambda\boldsymbol{I}_r)^{-1}\boldsymbol{g}.$$

The tangent and original losses agree to first-order at the anchor, so $\boldsymbol{B}^\top\nabla_{\boldsymbol{\theta}}\overline{F}(\boldsymbol{\theta}_A) = \boldsymbol{g}$. Therefore,

$$\frac{\mathrm{d}}{\mathrm{d}\gamma}\overline{F}\big(\boldsymbol{\theta}_A + \gamma\boldsymbol{B}\boldsymbol{\eta}_\lambda^\star\big)\Big|_{\gamma=0} = \boldsymbol{g}^\top\boldsymbol{\eta}_\lambda^\star = -\boldsymbol{g}^\top(\boldsymbol{G} + \lambda\boldsymbol{I}_r)^{-1}\boldsymbol{g} \le 0,$$

with strict inequality if $\boldsymbol{g} \neq \boldsymbol{0}$. In this case, differentiability gives

$$\overline{F}\big(\boldsymbol{\theta}_A + \gamma\boldsymbol{B}\boldsymbol{\eta}_\lambda^\star\big) - \overline{F}(\boldsymbol{\theta}_A) = -\gamma\boldsymbol{g}^\top(\boldsymbol{G} + \lambda\boldsymbol{I}_r)^{-1}\boldsymbol{g} + o(\gamma),$$

which is negative for all sufficiently small $\gamma > 0$. If $\boldsymbol{g} = \boldsymbol{0}$, then $\boldsymbol{\eta}_\lambda^\star = \boldsymbol{0}$, and the loss remains unchanged. □

The same argument applies to the empirical calibration loss, whose gradient with respect to $\boldsymbol{\eta}$ at $\boldsymbol{\eta} = \boldsymbol{0}$ is $\widehat{\boldsymbol{g}}$. If $\mathrm{tr}(\widehat{\boldsymbol{G}}) > 0$ and $\kappa > 0$, then $\lambda_\kappa > 0$. Since $\widehat{\boldsymbol{G}} \succeq \boldsymbol{0}$, the matrix $\widehat{\boldsymbol{G}} + \lambda_\kappa\boldsymbol{I}_r$ is positive definite, and the solution

$$\widehat{\boldsymbol{\eta}}_\kappa^\star = -\big(\widehat{\boldsymbol{G}} + \lambda_\kappa\boldsymbol{I}_r\big)^{-1}\widehat{\boldsymbol{g}},$$

defines a strict descent direction $\boldsymbol{B}\widehat{\boldsymbol{\eta}}_\kappa^\star$ for the original nonlinear calibration loss whenever $\widehat{\boldsymbol{g}} \neq \boldsymbol{0}$. If $\widehat{\boldsymbol{g}} = \boldsymbol{0}$, the correction is zero.

A.2 Cost of subspace restriction

The subspace restriction reduces computational cost and can improve statistical stability, but excludes some parameter directions. To quantify the resulting approximation, we compare the restricted solution with the unrestricted minimizer of the same quadratic objective.

Proposition A.2 (Exact decomposition of restriction and regularization costs)**.** *Let* $\boldsymbol{A} = T^{-1}\sum_{t=1}^{T}\boldsymbol{G}_{t,A} \succeq \boldsymbol{0}$*, and suppose* $\boldsymbol{\delta}^\star$ *minimizes* $Q_A(\boldsymbol{\delta})$ *over the full parameter space. Let* $\boldsymbol{\eta}_0^\star$ *minimize* $Q_A(\boldsymbol{B\eta})$ *over the prescribed subspace, and, for* $\lambda > 0$*, let* $\boldsymbol{\eta}_\lambda^\star$ *minimize* $Q_A(\boldsymbol{B\eta}) + \lambda\|\boldsymbol{\eta}\|_2^2/2$*. Then*

$$Q_A(\boldsymbol{B\eta}_\lambda^\star) - Q_A(\boldsymbol{\delta}^\star) = \frac{1}{2}\left\|\boldsymbol{B\eta}_0^\star - \boldsymbol{\delta}^\star\right\|_{\boldsymbol{A}}^2 + \frac{1}{2}\left\|\boldsymbol{B}\left(\boldsymbol{\eta}_\lambda^\star - \boldsymbol{\eta}_0^\star\right)\right\|_{\boldsymbol{A}}^2,$$

where $\|\boldsymbol{v}\|_{\boldsymbol{A}}^2 = \boldsymbol{v}^\top\boldsymbol{A}\boldsymbol{v}$ *is the curvature seminorm induced by* $\boldsymbol{A}$*.*

Proof. Since Q_A is quadratic with Hessian $\boldsymbol{A}$ and $\boldsymbol{\delta}^\star$ is a minimizer,

$$Q_A(\boldsymbol{\delta}) - Q_A(\boldsymbol{\delta}^\star) = \frac{1}{2}\left\|\boldsymbol{\delta} - \boldsymbol{\delta}^\star\right\|_{\boldsymbol{A}}^2.$$

The first-order condition for the unregularized restricted problem gives

$$\boldsymbol{B}^\top\boldsymbol{A}\left(\boldsymbol{B\eta}_0^\star - \boldsymbol{\delta}^\star\right) = \boldsymbol{0}.$$

Moreover,

$$\boldsymbol{B\eta}_\lambda^\star - \boldsymbol{\delta}^\star = \boldsymbol{B\eta}_0^\star - \boldsymbol{\delta}^\star + \boldsymbol{B}\left(\boldsymbol{\eta}_\lambda^\star - \boldsymbol{\eta}_0^\star\right).$$

The cross term in the expansion is zero because

$$\left(\boldsymbol{B\eta}_0^\star - \boldsymbol{\delta}^\star\right)^\top\boldsymbol{A}\boldsymbol{B}\left(\boldsymbol{\eta}_\lambda^\star - \boldsymbol{\eta}_0^\star\right) = 0.$$

Expanding the squared curvature seminorm and substituting it into the exact quadratic identity proves the result. □

The first term is the exact increase in the quadratic objective caused by restricting the correction to $\mathrm{span}(\boldsymbol{B})$, whereas the second is the additional gap induced by regularization. The first term is zero if $\boldsymbol{\delta}^\star \in \mathrm{span}(\boldsymbol{B})$ or, more generally, if some feasible correction differs from $\boldsymbol{\delta}^\star$ only along $\mathrm{Null}(\boldsymbol{A})$. The second term may remain positive because regularization shrinks the correction toward the anchor. Both terms are measured in the GGN curvature geometry, so deviations along high curvature directions contribute more than those along flat directions.

B Comparison of loss optimization strategies

This section compares four ways of optimizing the supervised loss around a merged anchor. The anchor $\boldsymbol{\theta}_A$, subspace matrix $\boldsymbol{B}$, and calibration data are fixed throughout, so the comparison isolates the effects of model linearization and loss approximation. Table 5 summarizes the four strategies, which differ in whether the model output is linearized and whether the outer task loss is quadratically expanded. We discuss these strategies in detail below.

For a coefficient vector $\boldsymbol{\eta} \in \mathbb{R}^r$, define the original average loss restricted to the prescribed subspace as

$$\mathcal{L}(\boldsymbol{\eta}) = \frac{1}{T}\sum_{t=1}^{T} F_t(\boldsymbol{\theta}_A + \boldsymbol{B\eta}).$$

Let

$$\boldsymbol{D}_{t,A}(\boldsymbol{X}) = \boldsymbol{J}_{t,A}(\boldsymbol{X})\boldsymbol{B},$$

denote the directional Jacobian at the anchor. Replacing the nonlinear model output with its tangent approximation gives

$$\widetilde{\mathcal{L}}(\boldsymbol{\eta}) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{(\boldsymbol{X},Y)\sim P_t}\ell_t\left(\boldsymbol{z}_t(\boldsymbol{X};\boldsymbol{\theta}_A) + \boldsymbol{D}_{t,A}(\boldsymbol{X})\boldsymbol{\eta}, Y\right).$$

Table 5: Comparison of loss optimization strategies around the same anchor and within the same subspace. Rows specify the treatment of the model output, while columns specify the treatment of the outer task loss.

	Outer loss unexpanded	Outer loss quadratically expanded
Original nonlinear output	**Direct original loss** *Derivatives:* repeated full network forward and backward passes for gradient evaluation. *Optimization:* iterative and generally nonconvex.	**Original loss quadratic** *Derivatives:* gradient and projected exact Hessian constructed once at the anchor using second-order network derivatives. *Optimization:* one linear solve; curvature may be indefinite.
Tangent linearized output	**Direct tangent loss** *Derivatives:* directional Jacobians computed once at the anchor. *Optimization:* iterative in coefficient space using outer loss gradients.	**LOSSOPT** *Derivatives:* directional Jacobians and GGN computed once at the anchor. *Optimization:* one regularized linear solve.

The original and tangent losses agree in value and gradient at the anchor:

$$\mathcal{L}(\mathbf{0}) = \widetilde{\mathcal{L}}(\mathbf{0}), \qquad \nabla_{\boldsymbol{\eta}}\mathcal{L}(\mathbf{0}) = \nabla_{\boldsymbol{\eta}}\widetilde{\mathcal{L}}(\mathbf{0}) = \boldsymbol{g}.$$

Their second derivatives differ. The Hessian of the tangent loss is the projected GGN curvature

$$\boldsymbol{G} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{P_t}\left[\boldsymbol{D}_{t,A}(\boldsymbol{X})^\top \boldsymbol{W}_{t,A}(\boldsymbol{X}, Y)\boldsymbol{D}_{t,A}(\boldsymbol{X})\right],$$

whereas the Hessian of the original loss is

$$\nabla^2_{\boldsymbol{\eta}}\mathcal{L}(\mathbf{0}) = \boldsymbol{G} + \boldsymbol{R}, \tag{5}$$

with

$$\boldsymbol{R} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{P_t}\left[\sum_{k=1}^{q_t} s_{t,A,k}(\boldsymbol{X}, Y)\boldsymbol{B}^\top \nabla^2_{\boldsymbol{\theta}} z_{t,k}(\boldsymbol{X}; \boldsymbol{\theta}_A)\boldsymbol{B}\right].$$

The matrix $\boldsymbol{R}$ is the curvature contributed by the nonlinear dependence of the model output on its parameters. These identities clarify the differences among the following four strategies.

Direct optimization of the original loss The most direct strategy solves

$$\underset{\boldsymbol{\eta}\in\mathbb{R}^r}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{\eta}) + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_2^2. \tag{6}$$

This retains both the nonlinear network and the exact supervised loss. It therefore avoids local approximation error in the objective. However, restricting the update to a low-dimensional subspace does not eliminate the cost of evaluating the nonlinear network. Each optimization iteration still requires forward and backward passes through the full model, and the resulting objective generally remains nonconvex. Its solution also depends on the iterative optimizer, learning rate, stopping rule, and initialization.

Quadratic expansion of the original loss Alternatively, one may directly expand the original composite loss at the anchor:

$$q_{\mathrm{H}}(\boldsymbol{\eta}) = \mathcal{L}(\mathbf{0}) + \boldsymbol{g}^\top\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\eta}^\top(\boldsymbol{G} + \boldsymbol{R})\boldsymbol{\eta} + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_2^2. \tag{7}$$

This surrogate yields a linear stationarity equation, but computing $\boldsymbol{R}$ requires second-order differentiation through the nonlinear network. Even without explicitly forming the full parameter Hessian, constructing its projection onto the subspace requires exact Hessian–vector products involving the model output.

Moreover, $\boldsymbol{R}$ has no fixed sign because it is weighted by the loss gradient in the output space. Consequently, the exact projected Hessian $\boldsymbol{G} + \boldsymbol{R}$ may be indefinite at a merged anchor. An illustrative example is shown in Figure 2. The regularized stationary point is a unique minimizer only when $\boldsymbol{G} + \boldsymbol{R} + \lambda \boldsymbol{I}_r$ is positive definite; otherwise, it may be a saddle point or the quadratic objective may be unbounded below. Ensuring positive definiteness can require a regularization level large enough to dominate the negative eigenvalues of the projected Hessian.

Direct optimization of the tangent loss A third strategy replaces the nonlinear network output by its tangent model but retains the task loss without a quadratic expansion:

$$\underset{\boldsymbol{\eta} \in \mathbb{R}^r}{\text{minimize}} \quad \widetilde{\mathcal{L}}(\boldsymbol{\eta}) + \frac{\lambda}{2} \|\boldsymbol{\eta}\|_2^2. \tag{8}$$

The directional Jacobians $\boldsymbol{D}_{t,A}(\boldsymbol{X})$ are computed only at the anchor. Once they are available, subsequent objective and gradient evaluations operate entirely in the output and coefficient spaces, without further differentiation through the nonlinear network. If ℓ_t is convex in its prediction argument, then $\widetilde{\mathcal{L}}$ is convex in $\boldsymbol{\eta}$, and the regularized objective is strictly convex when $\lambda > 0$. Nevertheless, common losses such as cross-entropy do not yield a closed-form minimizer, so an iterative solver is still required. Direct tangent loss optimization also retains higher order variation of the outer loss along the tangent model and may therefore move beyond the region in which a local correction around the anchor is most reliable.

Quadratic expansion of the tangent loss Our method instead expands the tangent loss:

$$q_{\mathrm{G}}(\boldsymbol{\eta}) = \mathcal{L}(\mathbf{0}) + \boldsymbol{g}^\top \boldsymbol{\eta} + \frac{1}{2} \boldsymbol{\eta}^\top \boldsymbol{G} \boldsymbol{\eta} + \frac{\lambda}{2} \|\boldsymbol{\eta}\|_2^2. \tag{9}$$

Because the tangent output is affine in $\boldsymbol{\eta}$, its model curvature term is identically zero. Thus, the quadratic term is exactly the GGN curvature rather than the full parameter Hessian. When each ℓ_t is convex in its prediction argument, $\boldsymbol{W}_{t,A} \succeq \mathbf{0}$ and hence $\boldsymbol{G} \succeq \mathbf{0}$. For $\lambda > 0$, the regularized coefficient matrix is positive definite and the unique minimizer is

$$\boldsymbol{\eta}^\star = -(\boldsymbol{G} + \lambda \boldsymbol{I}_r)^{-1} \boldsymbol{g}.$$

Computing $\boldsymbol{g}$ and $\boldsymbol{G}$ requires only the directional Jacobians $\boldsymbol{D}_{t,A}$ and the first two derivatives of the loss with respect to the model output. It avoids second-order differentiation through the network and reduces optimization to one regularized $r \times r$ linear system. The quadratic approximation also deliberately restricts the objective to the local first- and second-order behavior at the anchor. Together with explicit ridge regularization, this can act as a local regularizer against corrections driven by higher order behavior away from the anchor.

The four strategies therefore make distinct tradeoffs. Direct original loss optimization is the most faithful to the nonlinear network but is also the most computationally demanding and remains nonconvex. Expanding the original loss produces a quadratic surrogate but retains the expensive and potentially indefinite model curvature term. Direct tangent loss optimization removes network nonlinearity and can be convex, but still requires iterative optimization. In the PSD-GGN setting, LossOpt combines fixed directional Jacobians, positive semidefinite GGN curvature, and a closed-form regularized solution.

The comparison with direct tangent loss optimization in Figure 4 further shows that retaining the unexpanded tangent objective does not necessarily yield better test performance. In that experiment, direct tangent loss optimization achieves a higher validation score, whereas the local quadratic surrogate attains a higher test score. This result does not imply that the quadratic surrogate universally dominates direct optimization, but is consistent with its role as an additional local regularizer.

C Experimental Details and Additional Analyses

This section provides the experimental protocol, implementation details, and additional results.

Table 6: Dataset split sizes used in the vision experiments.

Task	Training pool	Validation	Test	Classes
Cars	7330	814	8041	196
DTD	3384	376	1880	47
EuroSAT	21600	2700	2700	10
GTSRB	23976	2664	12630	43
MNIST	55000	5000	10000	10
RESISC45	17010	1890	6300	45
SVHN	68257	5000	26032	10
SUN397	17865	1985	19850	397
STL10	4500	500	8000	10
OxfordIIITPet	3312	368	3669	37
Flowers102	918	102	6149	102
CIFAR100	45000	5000	10000	100
PCAM	257144	5000	32768	2
FER2013	25839	2870	7178	7
CIFAR10	45000	5000	10000	10
Food101	70747	5000	25249	101
FashionMNIST	55000	5000	10000	10
RenderedSST2	6228	692	1821	2
EMNIST	235000	5000	40000	10
KMNIST	55000	5000	10000	10

C.1 Datasets and data splits

Vision task suites Following the protocol commonly adopted for this benchmark (Wang et al., 2024; Gargiulo et al., 2025; Marczak et al., 2025; Li et al., 2026), we use the dataset loaders and train, validation, and test splits implemented in TSV-M. The suites containing 8, 14, and 20 tasks consist of the first 8, 14, and 20 datasets in Table 6, respectively. The training pool is used only to construct the calibration set, while the validation and test splits are kept separate. The Food101 counts reflect the removal of three unavailable training images and one unavailable test image. EMNIST uses the digits configuration containing ten classes.

For each task, $2{,}000$ examples are sampled from the training pool to form the calibration set. Sampling uses seed zero and is performed without replacement when the training pool contains at least $2{,}000$ examples. Flowers102, whose training pool is smaller, is sampled with replacement. The calibration sets are used only to estimate the gradient and curvature in coefficient space. The validation split is used for block and model selection, including the regularization parameter. The test split is evaluated once after all choices have been fixed. Example contributions are averaged first within each task and then across tasks, giving every task equal weight.

GLUE task suite The language model experiments use the eight GLUE tasks included in the WUDI protocol (Cheng et al., 2025): CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. We use the data splits distributed with the WUDI implementation. For a task with N original training examples, the data are divided without shuffling: the first $\lfloor 0.9N \rfloor$ examples form the training pool, and the remaining examples form the validation split.

The released WUDI test artifact is constructed from the official GLUE validation data. The matched validation split is used for MNLI, and some tasks are subsampled to $1{,}001$ test examples, as reported in Table 7. We use this artifact without further modification. The expert score is the corresponding single task checkpoint score under the WUDI evaluation protocol and is used to normalize each task metric.

We use the pre-trained RoBERTa-Base model and the eight task-specific checkpoints distributed with the WUDI experimental setup. For each GLUE task, $1{,}000$ examples are sampled without replacement from the training pool to form the calibration set. Sampling is deterministic and uses base seed 42 with a task-specific offset. These examples are disjoint from the validation and test splits and are used only to estimate the gradient and curvature in coefficient space. The validation split is used for model selection, after which the released WUDI test set is evaluated once.

Table 7: Dataset configuration for the RoBERTa-Base experiments. The test counts follow the released WUDI test artifact rather than the complete official GLUE validation splits. Expert scores are the single task references used for normalization.

Task	Training pool	Validation	Test	Output	Expert score
CoLA	7,695	856	1,001	Binary	56.52
SST-2	60,614	6,735	872	Binary	94.72
MRPC	3,301	367	408	Binary	87.99
STS-B	5,174	575	1,001	Regression	86.36
QQP	327,461	36,385	1,001	Binary	89.71
MNLI	353,431	39,271	1,001	3 classes	87.01
QNLI	94,268	10,475	1,001	Binary	91.71
RTE	2,241	249	277	Binary	66.43

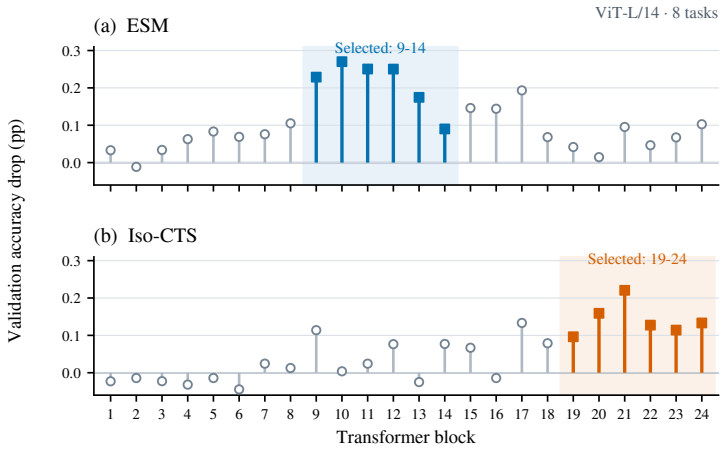


Figure 5: Individual block ablations on ViT-L/14 with eight tasks. Each point shows the decrease in validation macro accuracy after removing one block of the source update. Shading marks the ranges used by LOSSOPT (T+I+E): ESM blocks 9–14 and Iso-CTS blocks 19–24.

C.2 BLOCK AND MODEL SELECTION

Block selection Block ranges are chosen through preliminary ablations on the validation split. For each source model, we remove the update of one Transformer block while keeping the remaining parameters unchanged and measure the resulting decrease in validation macro accuracy. These ablations are used to identify regions that generally have a greater influence on performance, rather than to produce a strict ranking of individual blocks. We then select six blocks covering each influential region and fix them for the subsequent experiments.

Figure 5 provides an example using ViT-L/14 with eight tasks. The ESM update has a greater influence around blocks 9–14, whereas the effect of Iso-CTS is concentrated in the later blocks 19–24. This difference motivates selecting source-specific block regions when constructing the subspace.

Model selection Each source method first provides a merged update with its own source scale. These scales are determined before applying LOSSOPT and remain fixed when constructing the anchor, basis directions, gradient, and curvature.

Given the scaled source updates, we symmetrize the empirical curvature matrix and consider $\mathcal{K} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, with $\lambda_\kappa = \kappa \mathrm{tr}(\widehat{\boldsymbol{G}})/r$. For each κ, the coefficient vector is obtained by solving

$$\left(\widehat{\boldsymbol{G}} + \lambda_\kappa \boldsymbol{I}_r\right) \widehat{\boldsymbol{\eta}}^\star_\kappa = -\widehat{\boldsymbol{g}}.$$

Table 8: Source scales and model selection settings for LOSSOPT (T+I+E) in the vision experiments. TA denotes Task Arithmetic.

Backbone	Tasks	TA	Iso-CTS	ESM	κ
ViT-B/32	8	0.2	1.5	0.9	10^{-2}
ViT-B/32	14	0.1	1.2	0.7	10^{-5}
ViT-B/32	20	0.1	1.0	0.7	10^{-5}
ViT-B/16	8	0.3	1.6	0.8	10^{-2}
ViT-B/16	14	0.1	1.2	0.7	10^{-3}
ViT-B/16	20	0.1	1.1	0.7	10^{-2}
ViT-L/14	8	0.3	1.9	0.8	10^{-2}
ViT-L/14	14	0.2	1.5	0.7	10^{-2}
ViT-L/14	20	0.1	1.2	0.7	10^{-2}

The candidate with the highest mean normalized accuracy on the validation split is selected. Thus, the source scales determine the updates supplied to LOSSOPT, while κ controls regularization of the correction coefficients. The test split is evaluated once after these choices have been fixed.

Table 8 reports the settings used for LOSSOPT (T+I+E). For example, on ViT-B/16 with fourteen tasks, the source scales for Task Arithmetic, Iso-CTS, and ESM are 0.1, 1.2, and 0.7, respectively, and validation selects $\kappa = 10^{-3}$.

C.3 ADDITIONAL MERGING RESULTS

Taskwise performance Figures 6 and 7 complement the aggregate results by showing how performance changes across individual tasks. The per-task vision results in Figure 6 are obtained from the authors' own reproduction. In the vision experiments, LOSSOPT generally preserves the strong performance of the ESM anchor while improving a broad range of tasks across backbones and task suites. A similar pattern appears on GLUE, where most tasks benefit from the correction although a few show small decreases. Thus, the aggregate improvements are distributed across tasks rather than confined to a single dataset.

Interpretation of optimized direction coefficients To examine how LOSSOPT combines different source updates, we visualize the optimized coefficients of LOSSOPT (T+I+E) on ViT-B/32 with fourteen tasks. Let $\mathcal{G}$ contain the six selected Transformer blocks and the Rest group, which contains all entries of each source update outside its six selected blocks. For $s \in \{\mathrm{E}, \mathrm{I}, \mathrm{T}\}$ and $g \in \mathcal{G}$, let $\widehat{\eta}^{\star}_{s,g}$ denote the optimized correction coefficient for source s and parameter group g, where E, I, and T represent ESM, Iso-CTS, and Task Arithmetic, respectively.

Because ESM provides the anchor, the final merged update can be written as

$$\boldsymbol{\Delta}_M = \sum_{g \in \mathcal{G}} \left[\left(1 + \widehat{\eta}^{\star}_{\mathrm{E},g}\right) \boldsymbol{b}_{\mathrm{E},g} + \widehat{\eta}^{\star}_{\mathrm{I},g} \boldsymbol{b}_{\mathrm{I},g} + \widehat{\eta}^{\star}_{\mathrm{T},g} \boldsymbol{b}_{\mathrm{T},g} \right].$$

Accordingly, the absolute coefficients shown in the figure are $\alpha_{\mathrm{E},g} = 1 + \widehat{\eta}^{\star}_{\mathrm{E},g}$, $\alpha_{\mathrm{I},g} = \widehat{\eta}^{\star}_{\mathrm{I},g}$, and $\alpha_{\mathrm{T},g} = \widehat{\eta}^{\star}_{\mathrm{T},g}$. The ESM anchor therefore corresponds to absolute coefficients $(1, 0, 0)$ for every parameter group.

A coefficient measures the signed scaling of its corresponding basis direction. For an ESM direction, a value below or above one attenuates or amplifies that part of the anchor. For Iso-CTS and Task Arithmetic, a positive coefficient adds the source direction, whereas a negative coefficient moves in the opposite direction. The coefficients are estimated jointly and act on source updates whose scales have already been applied. Their magnitudes therefore should not be interpreted as standalone measures of source importance, since the basis directions may have different scales and norms.

Figure 8 shows that the ESM coefficients remain positive and generally close to one, indicating that the optimized model largely preserves the strong anchor while adjusting its contribution across parameter groups. Most Iso-CTS coefficients are positive, showing that its directions provide complementary corrections to the ESM update. In contrast, most Task Arithmetic coefficients are negative, with particularly large corrections in several late blocks. Conditional on the other directions, the

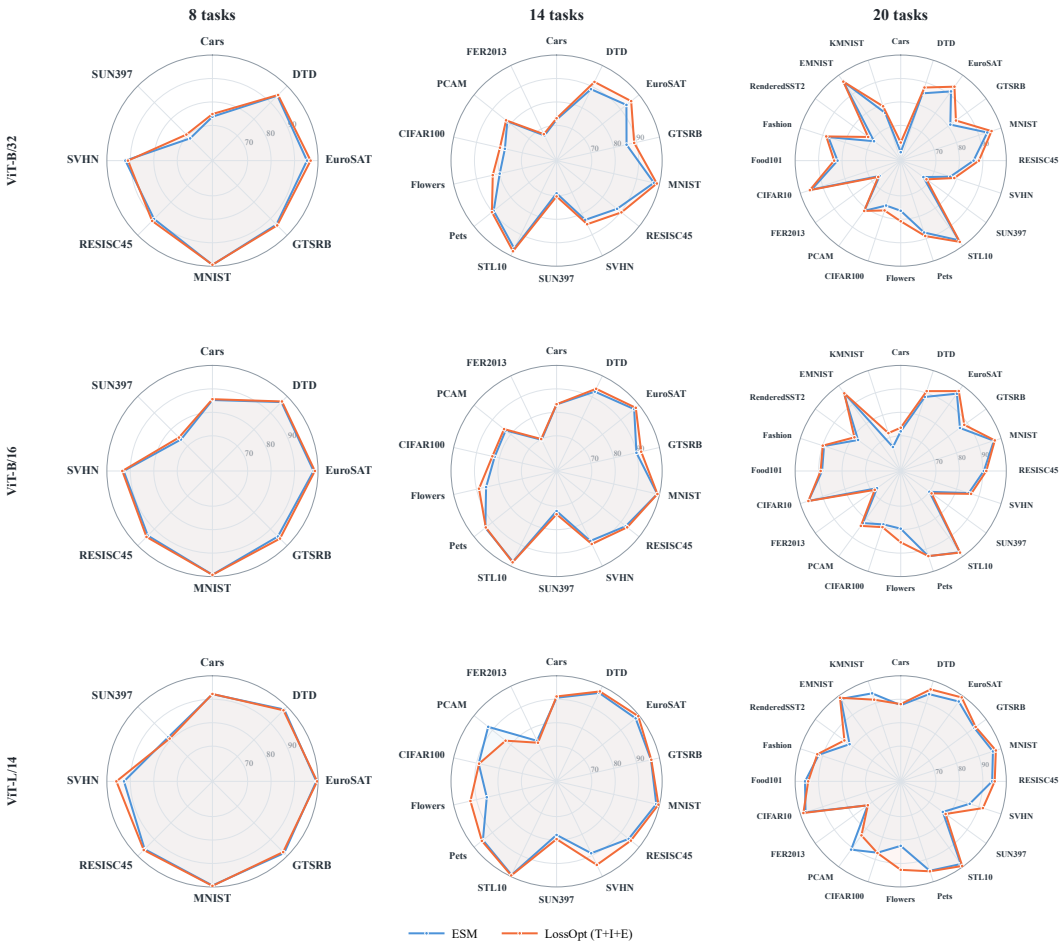


Figure 6: Per-task test accuracies of ESM and LOSSOPT (T+I+E) across three CLIP backbones and task suites. All panels use the same radial range of 55–100.

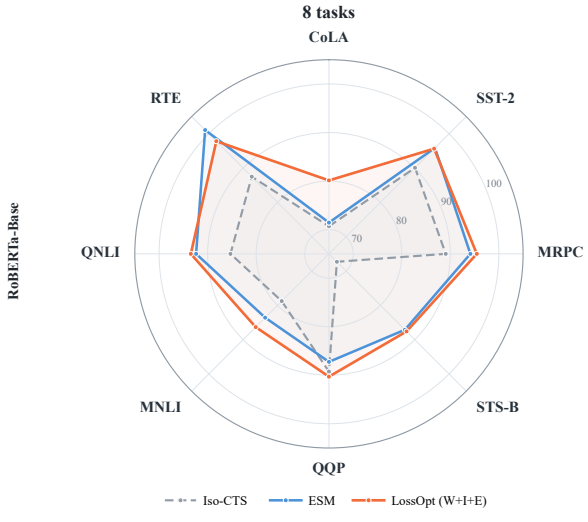


Figure 7: Task-level normalized test scores of Iso-CTS, ESM, and LOSSOPT (W+I+E) on the eight task GLUE benchmark using RoBERTa-Base. W, I, and E denote WUDI-Merging, Iso-CTS, and ESM, respectively. The radial range is 65–105.

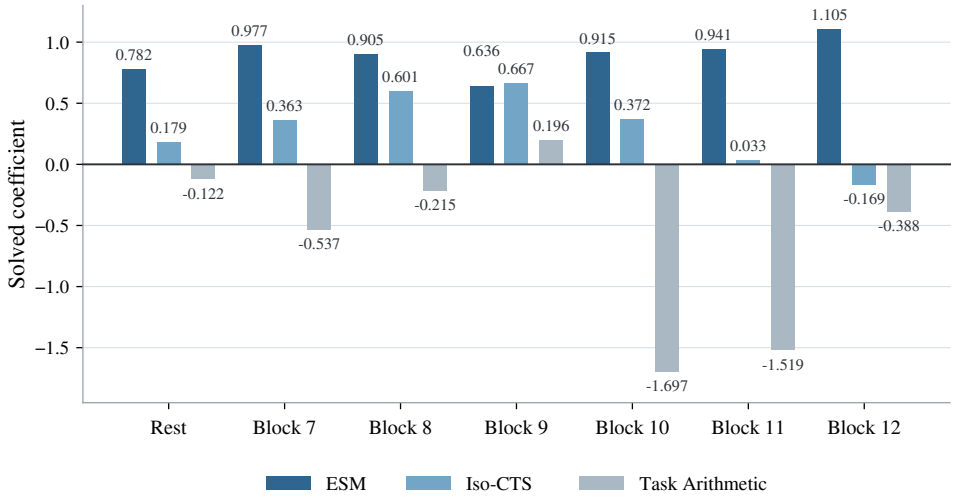


Figure 8: Optimized coefficients for the ESM, Iso-CTS, and Task Arithmetic directions in LOSSOPT (T+I+E) on ViT-B/32 with fourteen tasks. Rest contains all entries of each source update outside its six selected Transformer blocks.

Table 9: Runtime of LOSSOPT (T+I+E) in the vision experiments. Times are reported as `h:mm:ss`.

Backbone	Tasks	Deriv./solve	Selection/eval.	Core total
ViT-B/32	8	0:09:43	0:10:13	0:19:56
ViT-B/32	14	0:20:01	0:11:32	0:31:33
ViT-B/32	20	0:23:32	0:22:17	0:45:49
ViT-B/16	8	0:31:01	0:17:13	0:48:14
ViT-B/16	14	0:53:51	0:26:55	1:20:46
ViT-B/16	20	1:16:47	0:41:40	1:58:27
ViT-L/14	8	1:48:42	0:46:45	2:35:27
ViT-L/14	14	3:09:33	1:19:23	4:28:56
ViT-L/14	20	4:30:30	2:13:18	6:43:48

local loss objective therefore prefers reversing these components rather than adding them in their original orientation.

The substantial variation across parameter groups shows that a single coefficient for each source would be insufficient. Instead, LOSSOPT selectively retains, supplements, attenuates, or reverses different parts of the source updates. Together with the source ablation results, this pattern indicates that the improvement arises from direction dependent recombination of complementary updates rather than uniform averaging or global rescaling.

Runtime Table 9 reports core runtime results for LOSSOPT (T+I+E) across the nine vision configurations on a single NVIDIA GeForce RTX 4090 GPU. Deriv./solve includes derivative estimation and the linear solve. Selection/eval. combines validation-based selection with constructing the final merged model and evaluating it on the test split. Core total sums these recorded core stages. Runtime increases with the number of tasks and backbone size, primarily because derivative estimation requires executing the underlying network.

Local Approximation Diagnostics We examine the local approximation on two representative configurations: ViT-B/32 with eight tasks and ViT-B/16 with twenty tasks, both using ESM anchors. The original nonlinear loss, tangent loss, and quadratic surrogate are evaluated along

$$\boldsymbol{\theta}(s) = \boldsymbol{\theta}_A + s\boldsymbol{B}\widehat{\boldsymbol{\eta}}^{\star},$$

where $s = 1$ corresponds to the correction obtained by LOSSOPT. All three losses are recomputed on the same diagnostic data to isolate approximation error from differences in data sampling.

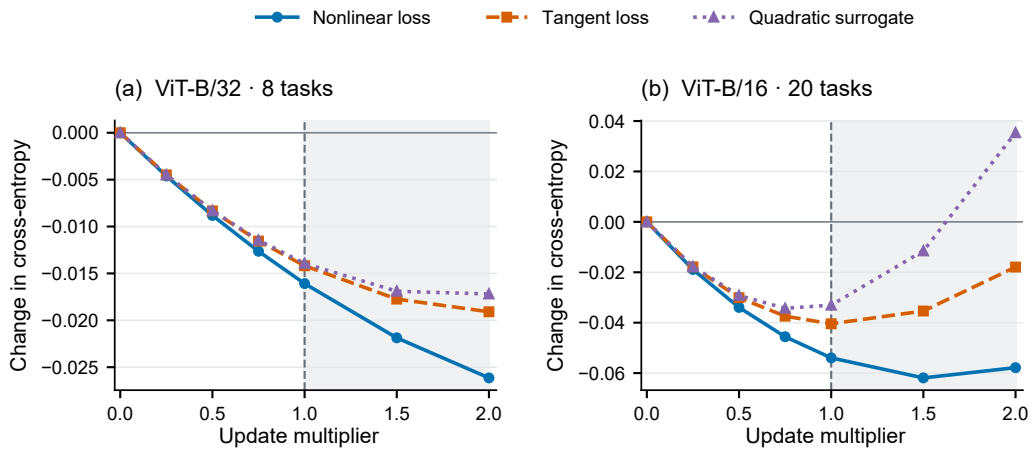


Figure 9: Loss changes along $\boldsymbol{\theta}(s) = \boldsymbol{\theta}_A + s\boldsymbol{B}\widehat{\boldsymbol{\eta}}^\star$. The curves show the loss of the nonlinear model, the tangent loss, and its quadratic approximation, all evaluated on the same diagnostic data. Changes from the anchor are averaged equally across tasks. The dashed line marks $s = 1$, and shading denotes extrapolation beyond this point.

At $s = 1$, the nonlinear cross-entropy changes by -0.016085 on ViT-B/32 and -0.053968 on ViT-B/16, while the quadratic surrogate predicts changes of -0.013937 and -0.033045, respectively. The surrogate therefore predicts the correct descent direction with comparable loss changes in both configurations, supporting its accuracy near the anchor. The average absolute task-level gaps between the nonlinear and tangent losses and between the tangent and quadratic losses are 0.001923 and 0.000261 on ViT-B/32, and 0.014224 and 0.007389 on ViT-B/16. Model linearization is therefore the larger source of approximation error, while the quadratic surrogate closely approximates the tangent loss in both configurations.

Direct tangent loss optimization uses the full task loss evaluated on the linearized model output, whereas LossOpt optimizes its local quadratic approximation. However, this greater fidelity does not necessarily yield better generalization. In the RoBERTa-Base experiment in Figure 4, direct tangent loss optimization attains the higher validation score, whereas LossOpt achieves the higher test score. This result indicates that the quadratic approximation provides additional local regularization.