GRADIENT-SIGN MASKING FOR TASK VECTOR TRANSPORT ACROSS PRE-TRAINED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

When a new release of a foundation model is published, practitioners typically need to repeat full fine-tuning, even if the same task has already been solved in the previous version. A promising alternative is to reuse the parameter changes (*i.e.*, task vectors) that capture how a model adapts to a specific task. However, they often fail to transfer across different pre-trained models due to their misaligned parameter space. In this work, we show that the key to successful transfer lies in the sign structure of the gradients of the new model. Based on this insight, we propose GradFix, a novel method that approximates the ideal gradient sign structure and leverages it to transfer knowledge using only a handful of labeled samples. Notably, this requires no additional fine-tuning: the adaptation is achieved by computing a few gradients at the target model and masking the source task vector accordingly. This yields an update that is locally aligned with the target loss landscape, effectively rebasing the task vector onto the new pre-training. We provide a theoretical guarantee that our method ensures first-order descent. Empirically, we demonstrate significant performance gains on vision and language benchmarks, consistently outperforming naive task vector addition and few-shot fine-tuning.

1 INTRODUCTION

Over the past few years, the paradigm in deep learning has shifted from training models from scratch to fine-tuning large pre-trained models. Adapting these large models to downstream tasks is advantageous, as it leads to stronger performance at a fraction of the cost. Such a shift has been evident in natural language processing and computer vision, where pre-trained models such as BERT (Devlin et al., 2019), CLIP (Radford et al., 2021), and their successors (OpenAI et al., 2023; Liu et al., 2023) have become the standard starting point for developing new applications.

Since companies and researchers often update the checkpoints by using more data or improved training pipelines, practitioners often face the need to repeat fine-tuning on the same downstream tasks. This creates redundancy: the work invested in adapting one release is not directly reusable on the next. To address this issue, several lines of research have investigated how to systematically relate or transfer knowledge across parameter spaces. The model rebasin literature (Ainsworth et al., 2023; Rinaldi et al., 2025) investigates how to align and merge independently trained models by exploiting permutation symmetries in their parameters. In parallel, task arithmetic (Ilharco et al., 2023; Ortiz-Jiménez et al., 2023; Yadav et al., 2023; Panariello et al., 2025; Marczak et al., 2025) has shown that task vectors (*i.e.*, the difference $\tau = \theta^{ft} - \theta^0$ between the base and fine-tuned parameters) can be added, subtracted, and merged across models to induce new capabilities. On a similar note, the literature on *mode connectivity* (Garipov et al., 2018; Frankle et al., 2020) demonstrates that different fine-tuned solutions can be linked by low-loss paths, highlighting that model parameters encode highly structured and transferable representations. Together, these advances suggest that parameters encode *rich* and *transferable* structure that can be systematically manipulated to obtain the desired behavior at reduced cost.

In particular, Rinaldi et al. (2025) formalizes this setting and proposes a technique to transport task vectors across transformer-based architectures. However, there remains a substantial gap between the transported fine-tune and an actual fine-tuned model. This gap highlights a key challenge: *while task vectors are informative about adaptation, their direct transfer across different pre-trains is not guaranteed to align with the loss geometry of the target model*. In fact, naive transfer may intro-

duce harmful directions in parameter space, *i.e.*, components of the task vector that are misaligned with the descent directions of the target loss, thus increasing the loss and limiting its effectiveness. Addressing this problem is crucial both for reducing the cost of adapting rapidly evolving foundation models and for enabling their use in low-data regimes, where re-running full fine-tuning is infeasible.

In this work, we introduce a framework for transporting task-specific knowledge across pre-trained models using **gradient-sign masking**. Our key insight is that, although a fine-tuning trajectory encodes valuable task information, its effectiveness on a new pre-trained model depends on the local loss geometry of the target. Inspired by findings from the optimization and distributed training literature (Bernstein et al., 2018; Alistarh et al., 2017), we exploit the observation that the sign of the gradient provides a robust surrogate for the descent direction. Leveraging this insight, we introduce a simple yet effective method to transport a task vector from a source model to a target pre-trained model: we mask the source task vector using the gradient signs of the target, keeping only the components aligned with the target's local loss landscape. We further provide a formal guarantee that, to first order, this transported update reduces the target loss, ensuring a principled safeguard against harmful or misaligned transfer.

Empirically, we show that this method enables highly effective transfer of fine-tuning knowledge from an outdated pre-trained model to a newer one, even in the low-data regime where gradients can only be estimated from a handful of samples, partially closing the gap between naive transfer and full fine-tuning on the target model. Our contributions are:

- We establish a theoretical connection between the *oracle task vector*, the ideal fine-tuning update on the target model, and quantities we can actually compute, namely the source task vector and the gradient at the zero-shot target model. We show that the sign of the zero-shot gradient provides a reliable proxy for the descent directions encoded in the target model.
- Building on this insight, we propose **GradFix**, a simple yet theoretically grounded mechanism that filters the source task vector according to the local loss geometry of the target model. We formally prove that this guarantees, to first order, that the transported update reduces the target loss.
- We demonstrate empirically that our method enables effective transport of fine-tuning knowledge across pre-trained models in both vision and text domains, even in the *low-data regime* where gradients must be estimated from only a handful of samples. This shows the practicality of our approach in scenarios where re-finetuning is infeasible.

2 Related Works

Model merging. A growing literature explores how to merge multiple fine-tuned checkpoints derived from the same pretrained model. Wortsman et al. (2022) introduced model soups, showing that simple weight averaging of fine-tuned checkpoints often improves generalization. Task arithmetic formalizes fine-tuning deltas as task vectors, which can be added or negated to edit model behavior (Ilharco et al., 2023). Building on this view, Yadav et al. (2023) proposed TIES-Merging, which resolves conflicts among task vectors by enforcing sign consistency before aggregation. For improved robustness and scalability, ATM interleaves tuning with merging (Zhou et al., 2024), and task-vector-based cluster vectors into compact representations (Zeng et al., 2024).

Model rebasin. A different family of methods focuses on explicit rebasin, aligning independently pretrained models into a shared parameterization so that task vectors can be transported across different pretrainings. Git Re-Basin introduced permutation matching to map two networks into a common basin (Ainsworth et al., 2023). For transformers, Imfeld et al. (2024) applies Optimal Transport to softly align components, while Rinaldi et al. (2025) proposes permutation- and spectral-based procedures that enable task vector transfer across distinct pretrained models. Extensions such as permutation least-squares (Nasery et al., 2025) and supernet formulations (Stoica et al., 2024) address more heterogeneous or large-scale settings.

Gradient information. A complementary line of work studies the utility of gradient signs and compressed gradient information. SignSGD (and its majority-vote variant) shows that one-bit sign information can suffice for convergence in distributed settings (Bernstein et al., 2018), while quan-

tization and error-feedback analyses formalize guarantees for compressed updates (Alistarh et al., 2017; Karimireddy et al., 2019). More recent methods leverage gradient magnitudes or sign statistics for efficient adaptation: Gradient-Mask Tuning masks low-importance parameters during LLM fine-tuning (Li et al., 2025), and sign-based federated variants weight client updates to address heterogeneity (Park et al., 2024).

3 PRELIMINARIES

Let θ_A and θ_B denote the parameters of the same architecture, pre-trained on different datasets (or with different hyperparameters). The fine-tune of θ_A on a downstream dataset $\mathcal{D}$ is denoted as θ_A^{ft}.

Model rebasin. The goal of rebasin (Ainsworth et al., 2023) is to align two independently trained models by mapping the parameters of one into the loss **basin** of the other, so that they become functionally compatible. This setting concerns only the pre-trained weights, without involving fine-tuning updates.

Task Arithmetic. A complementary perspective to model rebasin is offered by *task arithmetic* (Ilharco et al., 2023; Yadav et al., 2023), which studies linear operations on task-specific parameter updates. Given a pre-trained model θ^0 and a fine-tuned counterpart $\theta^\star$, the difference vector in parameter space $\tau = \theta^\star - \theta^0$ is called a *task vector*. Task vectors describe how a base model adapts to the task and can be added, subtracted, or merged to induce new behaviors. This setting usually assumes that all models share the same initialization θ^0, which ensures comparability across tasks.

Our Setting. In contrast, our goal is to apply task-vector style transfer when the same base initialization assumption does not hold: we want to transfer $\tau_A = \theta_A^{ft} - \theta_A$ from a source pre-train θ_A onto a different pre-train θ_B. This connects to rebasin in that the bases differ, but unlike traditional rebasin approaches, we do not seek to explicitly align parameterizations. Instead, we ask:

Which components of τ_A are truly transferable, and which would instead harm θ_B?

This question motivates our method, which leverages the local gradients of θ_B to selectively filter τ_A into a compatible and transferable update, effectively performing direct task vector transportation.

4 METHOD

GradFix is a framework for transferring task vectors across different pre-trains by filtering them with gradient information from the target model. As a conceptual starting point, we consider an *oracle* ideal setting where the target task vector, obtained from full fine-tuning, defines the ideal transferable directions (Sec. 4.1). Then the oracle is approximated with a *single gradient step* on the target model, using the signs of the gradients to capture an approximate direction of the full fine-tuning trajectory. This yields a *gradient-sign mask* that selectively filters the source task vector into a compatible update (Sec. 4.2). Finally, we extend the approach to the limited-data regime, where gradients are estimated from only a handful of labeled samples (Sec. 4.3).

4.1 GRADFIX (GRADIENT-SIGN MASKING)

We begin by considering an ideal scenario where the true fine-tuned task vector $\tau_B := \theta_B^{ft} - \theta_B$ of the model B and the whole target dataset $\mathcal{D}$ are available. Such a vector represents the optimal parameter change to adapt B on the target dataset $\mathcal{D}$. In this ideal setting, it is possible to construct a mask that perfectly retains only the components of a candidate update (*e.g.*, τ_A) that are aligned with τ_B, ensuring that every retained coordinate contributes to decreasing the loss. In other words, τ_B (or its sign structure) defines the "gold standard" for locally beneficial directions. Formally, we define as $\boldsymbol{m}^\star \in \{0,1\}^d$ the mask induced by τ_B, where d is the total number of model parameters and $i \in \{1, \dots, d\}$ indexes each coordinate:

$$m_i^\star = \mathbb{1}\{\text{sign}(\tau_{A,i}) = \text{sign}(\tau_{B,i})\}. \tag{1}$$

As shown in Fig. 1, applying this mask to τ_A produces the oracle-masked update $\delta^\star$, which preserves only the components consistent with τ_B:

$$\delta^\star := \boldsymbol{m}^\star \odot \tau_A, \tag{2}$$

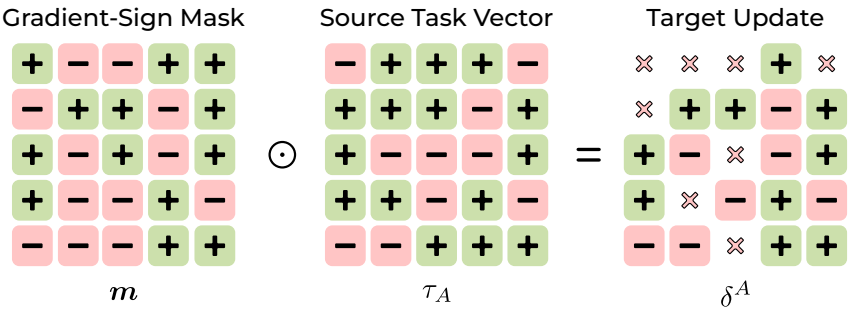

Figure 1: Illustration of our masking procedure. The gradient mask $\boldsymbol{m}$ suppresses harmful directions in the task vector τ_A while preserving those aligned with the target model.

where $\odot$ denotes the element-wise multiplication. This vector $\delta^\star$ represents a reliable transfer of τ_A onto θ_B, since it filters out all components of τ_A that are misaligned with the true adaptation directions of B. In practice, however, τ_B (and thus $\delta^\star$) is unavailable because it requires access to the fine-tuned target model θ_B^{ft}, which defeats the purpose of transporting the solution from A to B. To approximate this ideal mask, we consider the gradient of the zero-shot target model as a surrogate for τ_B, indicating locally beneficial directions:

$$\boldsymbol{g} := \nabla_\theta \mathcal{L}(\theta_B), \quad \mathcal{L}(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)], \tag{3}$$

where ℓ is the training objective (*e.g.*, cross-entropy) and (x, y) is a labeled example from $\mathcal{D}$. Based on this gradient, we define the gradient-sign mask $\boldsymbol{m}$, which retains only the components of τ_A whose sign matches that of the corresponding gradient coordinate:

$$m_i := \mathbb{1}\{\text{sign}(\tau_{A,i}) = \text{sign}(g_i)\}. \tag{4}$$

Intuitively, $\boldsymbol{g}$ acts as a signal for local alignment with the loss geometry of B. Notably, in the idealized setting where B is fine-tuned using full-batch gradient descent for a single epoch, the resulting task vector τ_B is exactly proportional to $-\boldsymbol{g}$, so the gradient-sign mask coincides with the oracle. This observation provides a clear justification for using the gradient-sign mask as an approximation of the ideal update, even when only a few labeled examples are available. The gradient-sign mask selectively retains only the components of τ_A whose sign matches the gradient of $\mathcal{L}(\theta_B)$, effectively pruning coordinates that would increase the loss for the target model B. In this way, the gradient-sign mask provides a practical surrogate for the trajectory-informed directions encoded in the unavailable τ_B, capturing the locally beneficial update directions without requiring access to the fully fine-tuned target model.

4.2 Transporting the Update

Given the gradient-sign mask $\boldsymbol{m}$ from Eq. (4), we define the updated target parameters by directly applying the masked task vector with a scaling factor $\alpha > 0$:

$$\theta_B^{\text{trans}} = \theta_B - \delta^A, \quad \delta^A := \alpha\,(\boldsymbol{m} \odot \tau_A), \tag{5}$$

It is important to note that τ_A points in a descent direction for model A, whereas the gradient $\boldsymbol{g}$ of the target model points in the ascent direction of its loss. By subtracting the masked vector δ^A, we ensure that each retained component moves opposite to the ascent direction of B, producing a descent-aligned update, whereas $\delta^\star$ already points in the descent direction since it is computed directly from τ_B.

Descent guarantee. To understand why such a gradient masking provides effective transfer, we analyze its effect on the loss of the target model B. Consider the transported update from Eq. (5), by expanding the target loss $\mathcal{L}$ around θ_B via a first-order Taylor approximation we obtain:

$$\mathcal{L}(\theta_B - \alpha\delta^A) \approx \mathcal{L}(\theta_B) - \alpha \boldsymbol{g}^\top \delta^A, \quad \text{where } \boldsymbol{g} = \nabla_\theta \mathcal{L}(\theta_B). \tag{6}$$

The sign of the inner product $\boldsymbol{g}^\top \delta^A$ determines whether the update increases or decreases the loss to first order. By construction, the gradient-sign mask $\boldsymbol{m}$ retains only components of τ_A that are

aligned with $\boldsymbol{g}$. Concretely, for each coordinate i, we have:

$$g_i \cdot (m_i \tau_{A,i}) = m_i |g_i| \, |\tau_{A,i}| \; \geq \; 0, \tag{7}$$

so that the overall inner product satisfies the following:

$$\alpha \boldsymbol{g}^\top \delta^A = \alpha \sum_i m_i |g_i| \, |\tau_{A,i}| \; \geq \; 0. \tag{8}$$

Thus, this implies that, for sufficiently small α, the update δ^A is guaranteed to be a descent direction for $\mathcal{L}$. Practically, the mask removes all sign-mismatched components of τ_A, so that every retained entry contributes positively to reducing the loss. Without masking, τ_A could contain harmful directions that increase the loss for B; with masking, the transported update is locally aligned with the descent geometry of the target model.

4.3 Limited Data Regime

In Sec. 4.1, we have assumed access to the full target dataset $\mathcal{D}$ to compute the gradient $\boldsymbol{g}$ at the zero-shot target model θ_B. In practice, one of the main motivations for task vector transport is the *few-shot* or limited data regime. If we had access to the entire dataset, we could directly fine-tune θ_B to obtain θ_B^{ft}, making task vector transfer efficient but sub-optimal.

When only a small number of samples is available, we estimate the gradient signs using a subset of labeled examples. Let $\mathcal{D}_s \subset \mathcal{D}$ denote a small subset of N samples. For each parameter coordinate i, we compute the sign of the gradient via **majority voting** across these samples:

$$\hat{s}_i = \operatorname{sign}\left(\sum\nolimits_{(x_n, y_n) \in \mathcal{D}_s} \operatorname{sign}\left(\nabla_\theta \ell(f_{\theta_B}(x_n), y_n) \right) \right). \tag{9}$$

Lemma (Concentration of Majority Vote Sign Estimator)**.** *Let $p_i = \Pr[\operatorname{sign}(\nabla_\theta \ell(f_{\theta_B}(x), y)) = \operatorname{sign}(\tau_{B,i})]$ denote the probability that a single-sample gradient aligns with the oracle task vector τ_B. Then, under mild independence assumptions, $p_i > 1/2$, and the majority vote estimator satisfies the following:*

$$\Pr\left[\operatorname{sign}(g_i) = \operatorname{sign}(\tau_{B,i})\right] \geq 1 - \exp\left(-2N(p_i - 1/2)^2\right), \tag{10}$$

which shows that the estimated sign concentrates around the true gradient direction as the number of samples N grows.

We provide a proof of this lemma in Appendix A, which makes use of Hoeffding's inequality (Hoeffding, 1963). In practice, even a few samples provide a robust estimate of the true gradient direction. Each gradient acts as a "vote" for the correct sign, and majority voting filters out noisy or conflicting directions. This guarantees that, with high probability, the masked task vector δ_B^{mask} points in a descent direction, preserving the first-order loss reduction property of the full-data update. As shown in Sec. 5.2, this approach is robust to small sample sizes, making it particularly attractive when direct fine-tuning of θ_B is expensive or prone to overfitting.

Algorithm 1 outlines the gradient-sign masked task vector transport procedure, showing how the source task vector is selectively applied to the target model using only a small subset of labeled data.

5 Experimental Results

Implementation details. For the Vision Settings, we consider CLIP ViT-B/16 and ViT-L/14 Vision Transformers (Radford et al., 2021), implemented in Open-CLIP (Cherti et al., 2023). The original pre-trained weights are denoted θ_A and the target model weights θ_B. For ViT-B/16, θ_A was pre-trained on Datacomp XL (`s13b`, `b90k`) and θ_B on LAION-2B (`s34b`, `b88k`). For ViT-L/14, θ_A was pre-trained on Datacomp XL (`s13b`, `b90k`) and θ_B on LAION-2B (`s32b`, `b82k`). For the Language Settings, we investigated different Text-To-Text Transfer Transformer (T5) (Raffel et al., 2020) models in the base configuration. As θ_A, we used `T5v1.1`, pre-trained on the C4 (Raffel et al., 2020) dataset without any supervised training. For θ_B, we used `FLAN-T5` (Chung et al., 2024), pre-trained and instruction-tuned on several datasets, including GSM8K (Cobbe et al., 2021), AQUA-RAT (Ling et al., 2017), and LAMBADA (Paperno et al., 2016). Task vectors were obtained

Algorithm 1 Gradient-Sign Masked Transport

Require: Source model θ_A, θ_A^{ft}, target model θ_B, target data subset $\mathcal{D}_s$, scaling α.
Ensure: Transported model θ_B^{trans}

1: Compute source task vector: $\tau_A \leftarrow \theta_A^{ft} - \theta_A$
2: **for** $(x_n, y_n) \in \mathcal{D}_s$ **do**
3: $\quad g^{(n)} \leftarrow \nabla_\theta \ell(f_{\theta_B}(x_n), y_n)$
4: Compute gradient signs $\hat{s}_i$ by majority voting ▷ Eq. (9)
5: Build gradient-sign mask: $m_i \leftarrow \mathbb{1}\{\text{sign}(\tau_{A,i}) = \hat{s}_i\}$
6: Compute transported update: $\delta^A \leftarrow \alpha\,(\boldsymbol{m} \odot \tau_A)$
7: **return** $\theta_B^{trans} \leftarrow \theta_B - \delta^A$ ▷ Updated target model

Table 1: Cross-dataset performance comparison between ViT-B/16 and ViT-L/14 models.

		EUROSAT		SVHN		GTSRB		RESISC45		DTD	
Model	$\lvert\mathcal{D}_s^c\rvert$	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**	**B/16**	**L/14**
θ_B *zero-shot*	-	49.41	62.80	50.58	37.28	48.29	56.12	67.98	73.12	55.96	63.35
θ_B *fine-tune*	-	98.70	98.95	97.45	97.80	98.65	99.16	95.66	97.06	83.19	83.56
$\theta_B + \tau_A$	-	49.58	62.77	50.84	39.09	49.31	56.03	67.87	73.49	56.27	63.56
$\theta_B + \delta^\star$	-	95.06	96.75	92.04	92.60	82.92	88.65	87.06	90.30	71.44	72.66
$TransFusion$	-	50.12	63.21	53.26	37.38	50.24	56.78	67.99	73.36	56.70	64.10
θ_B^{opt}	1	56.61	64.65	61.32	62.51	56.08	63.97	69.25	74.54	56.21	63.76
$\theta_B - \delta^A$	1	61.94	69.67	71.07	70.15	60.88	66.82	70.05	76.45	58.32	65.50
θ_B^{opt}	2	59.49	70.76	62.01	45.23	61.70	69.91	71.20	76.62	57.00	64.97
$\theta_B - \delta^A$	2	65.07	74.10	70.19	54.31	64.33	71.55	71.42	76.97	58.51	66.10
θ_B^{opt}	5	61.99	69.75	67.03	67.11	63.08	73.25	73.01	75.41	59.65	66.72
$\theta_B - \delta^A$	5	66.05	75.59	73.59	74.41	66.61	73.14	71.57	76.82	60.02	66.95

following the fine-tuning protocol of Ilharco et al. (2023): 2000 iterations, batch size 128, learning rate 1×10^{-5}, cosine annealing with 200 warm-up steps, AdamW optimizer (Loshchilov & Hutter, 2019), weight decay 0.1. The text encoder backbone was kept frozen following Cherti et al. (2023).

Baselines. We evaluate our method against several baselines. As a lower bound, we consider the zero-shot target model (θ_B *zero-shot*), *i.e.*, the base model without any fine-tuning. As an upper bound, we report $\theta_B + \delta^\star$, obtained by adding the source task vector τ_A masked with the signs of the true task vector τ_B to the target model. We also include the performance of the fully fine-tuned target model (θ_B *fine-tune*) and the naive task arithmetic transport ($\theta_B + \tau_A$). In addition, we compare against *TransFusion* (Rinaldi et al., 2025), which transports task vectors across transformer-based models via permutation alignment. Finally, we report the accuracy of a target model fine-tuned with the same number of randomly sampled examples per class $|D_s|$ used by our approach.

Supervision Budget $\mathcal{D}_s$. In all experiments, the subset $\mathcal{D}_s$ is drawn from the full downstream fine-tuning dataset $\mathcal{D}$ and constitutes only a fraction of its size. Throughout the tables, $|\mathcal{D}_s^c|$ indicates the number of examples *per class* used to estimate gradient signs for the target model θ_B. The corresponding proportions of $\mathcal{D}$ used to form $\mathcal{D}_s$ are provided in Appendix C.

5.1 Transport Experiments

Transport in the Vision Setting. Tab. 1 summarizes the results of task vector transport across CLIP ViT-B/16 and CLIP ViT-L/14 architectures, averaged over multiple random seeds that determine the composition of the sampled $\mathcal{D}_s$ (standard deviations are reported in Appendix B.3). Our GradFix , denoted by $\theta_B - \delta^A$, yields a consistent improvement over naive task vector addition ($\theta_B + \tau_A$) even when using a single sample per class to approximate true gradient signs. Notably, the naive addition performs nearly at the level of zero-shot initialization and fails to transfer meaningful task knowl-

Table 2: Cross-dataset performance of T5 models on different NLP tasks

Model	$\lvert\mathcal{D}_s^c\rvert$	SNLI	MNLI	RTE	QNLI	SCITAIL	AVG
θ_B *zero-shot*	-	34.24	35.21	47.20	50.54	50.38	43.51
θ_B *fine-tune*	-	88.20	86.30	84.40	92.79	95.32	89.40
$\theta_B + \tau_A$	-	31.61	30.75	47.36	50.52	50.46	42.12
$\theta_B + \delta^\star$	-	58.69	69.97	72.93	65.32	62.38	65.86
θ_B^{opt}	50	35.09	26.05	47.29	51.45	51.78	42.33
$\theta_B - \delta^A$	50	68.06	49.68	54.25	60.50	59.89	58.48

Table 3: Performance of θ_B with oracle or estimated gradient signs under different mask strategies: **agreement** retains matching signs, **force agreement** aligns all signs, and **random** assigns signs uniformly. Results averaged over seeds with $\lvert\mathcal{D}_s^c\rvert = 1$ on CLIP ViT-B/16.

Model	Mask Strategy	EUROSAT	RESISC45	GTSRB	SVHN	DTD	AVG
θ_B *zero-shot*	-	49.41	67.98	48.29	50.58	55.96	54.45
θ_B *fine-tune*	-	98.70	95.66	98.65	97.45	83.19	94.73
$\theta_B + \delta^\star$	force agreement	97.95	93.51	95.94	96.60	80.59	92.92
	agreement	95.06	87.06	82.92	92.04	71.44	85.71
$\theta_B - \delta^A$	force agreement	61.32	70.10	60.91	70.52	58.05	64.18
	agreement	61.94	70.05	60.89	71.07	58.32	64.45
$\theta_B - \delta^A$	random	49.49	67.97	48.41	50.54	56.06	54.50

edge. This confirms that GradFix effectively suppresses misaligned components of τ_A, preventing negative transfer due to pre-training mismatch.

To further evaluate our approach, we compare it against few-shot fine-tuning of θ_B, denoted as θ_B^{opt}, using the same limited target samples. GradFix achieves better performance, on both ViT-B/16 and ViT-L/14, while exhibiting smaller variance across seeds with respect to few-shot fine-tuning. Moreover, as the $\mathcal{D}_s$ size increases, our method continues to provide stable gains, whereas θ_B^{opt} suffers from fluctuations and instability due to the composition of the supervision dataset. These results demonstrate that our approach ensures consistent and reliable task vector transport, remaining stable across different subsets $\mathcal{D}_s$. Importantly, this robustness is achieved with a single forward–backward pass to obtain the mask $\boldsymbol{m}$, highlighting the efficiency and simplicity of the proposed method.

Transport in the Language Setting. Tab. 2 reports results on task vector transport across T5 models evaluated on closed-vocabulary text classification benchmarks. While direct addition of τ_A to θ_B fails to transfer knowledge effectively, our method substantially closes the gap toward full fine-tuning, confirming its ability to identify and retain task-relevant directions. Notably, the relative improvement over naive transfer is even larger than in the vision setting, underscoring the robustness of our approach in domains where task transfer is especially challenging. This demonstrates that the benefits of GradFix are not confined to vision, and that a single forward–backward pass suffices to enable reliable and efficient task vector transport also in the language domain.

5.2 Masking Strategies

To analyze the effect of different mask sign strategies on the transport of the task vector τ_A, we compare our mask construction method, denoted **agreement**, with two alternatives: **force agreement** and **random** masks. Tab. 3 reports results using 1 sample per class on ViT-B/16, averaged across multiple random seeds. As reference points, we include the zero-shot base model (θ_B) and the fully fine-tuned model (θ_B *fine-tune*), providing lower and upper bounds on performance.

For both δ^A and $\delta^\star$, the mask $\boldsymbol{m}$ determines which directions of τ_A are retained Eq. (5). In **agreement**, $\boldsymbol{m}$ retains only the coordinates whose signs match those of the reference as in Eq. (4). In

force agreement, all signs of τ_A are aligned with the signs of the gradient $\boldsymbol{s}$, obtaining the mask:

$$m_i^{fa} = \text{sign}(\tau_{A,i}) \cdot \hat{s}_i, \tag{11}$$

flipping any entries that disagree with the reference. When the oracle τ_B is used as the reference, force agreement generally outperforms agreement, as fully leveraging the true task direction maximizes transfer. In contrast, using few-shot gradient-based estimates from θ_B, agreement performs slightly better than force agreement. This is consistent with the fact that the gradient-based estimate is noisy. Forcing all directions can propagate errors, while keeping only agreeing entries provides a more reliable mask. Finally, we evaluate **random masks**, where the signs are randomly sampled from a uniform distribution $\mathcal{U}\{-1, +1\}$. The random masks yield performance close to the zero-shot baseline, confirming that unstructured perturbations provide no meaningful guidance.

5.3 Subset Data Selection

In the previous experiments, we randomly selected the subset $\mathcal{D}_s$ from $\mathcal{D}$. Here, we evaluate different heuristics, including random, herding, k-medoids, and coreset, for constructing $\mathcal{D}_s$ to estimate gradient signs, aiming to understand how subset selection impacts the accuracy and efficiency of gradient estimation. Each strategy is evaluated at $b \in \{1, 2, 5, 10, 20\}$ examples per class. For herding (Rebuffi et al., 2016; Harvey, 2014), k-medoids (Kaufman & Rousseeuw, 1987), and coreset (Sener & Koltun, 2018), images are embedded using the frozen CLIP image encoder of the source model θ_A. Let f denote this frozen image encoder, normalized features are computed as $\boldsymbol{z}(x) = f(x)/\|f(x)\|$, and let $\mathcal{D}^c := \{(x, y) \in \mathcal{D} \mid y = c\}$.

Random. Sample uniformly from $\mathcal{D}$ without replacement.

Herding. Greedily select representatives $\mathcal{D}_s^c \subseteq \mathcal{D}^c$ of size b to match the class mean feature by minimizing the discrepancy of the running average:

$$\mathcal{D}_s^c \;=\; \underset{|\mathcal{D}_s^c|=b}{\arg\min} \Big\| \boldsymbol{\mu}_c - \frac{1}{|\mathcal{D}_s^c|} \sum_{x \in \mathcal{D}_s^c} \boldsymbol{z}(x) \Big\|_2, \qquad \boldsymbol{\mu}_c := \frac{1}{|\mathcal{D}^c|} \sum_{x \in \mathcal{D}^c} \boldsymbol{z}(x). \tag{12}$$

k-Medoids. Select $\mathcal{D}_s^c \subseteq \mathcal{D}^c$ of size b that minimizes the in-class assignment cost under distance d:

$$\mathcal{D}_s^c \;=\; \underset{|\mathcal{D}_s^c|=b}{\arg\min} \sum_{x \in \mathcal{D}^c} \min_{s \in \mathcal{D}_s^c} d\big(\boldsymbol{z}(x), \boldsymbol{z}(s)\big). \tag{13}$$

Coreset (medoid-proximity greedy). Adopt a medoid-proximity greedy selection within $\mathcal{D}^c$:

$$\begin{aligned} &\text{(i) Seed:} \quad s_1 = \underset{j \in \mathcal{D}^c}{\arg\min} \sum_{k \in \mathcal{D}^c} d\left(\boldsymbol{z}(j), \boldsymbol{z}(k)\right), \\ &\text{(ii) Greedy:} \quad s_t = \underset{j \in \mathcal{D}^c \setminus \mathcal{D}_{s,t-1}^c}{\arg\min} \min_{s \in \mathcal{D}_{s,t-1}^c} d\left(\boldsymbol{z}(j), \boldsymbol{z}(s)\right), \quad t = 2, \ldots, b, \end{aligned} \tag{14}$$

where $\mathcal{D}_{s,t-1}^c = \{s_1, \ldots, s_{t-1}\}$ denotes the set of already selected samples. This strategy emphasizes prototypical samples to reduce variance in small budgets and is closely related to coreset selection for active learning (Sener & Koltun, 2018).

For each dataset and budget b, $\mathcal{D}_s = \bigcup_c \mathcal{D}_s^c$, gradient signs are estimated as discussed in Sec. 4.3 using majority-vote aggregation, and masked transport is applied to θ_B. We report in Fig. 2 the accuracy for each strategy as a function of images per class, with standard deviation across random seeds for the random baseline. Across datasets and budgets, structured selectors (coreset, herding, k-medoids) often provide small but consistent gains over random selection in the few-shot regime. Yet, random selection remains a strong baseline, with performance approaching that of structured methods as b increases, while incurring no memory or computation overheads from embedding or distances pre-computation. Importantly, this shows that our approach remains effective even when the subset $\mathcal{D}_s$ is chosen at random, validating its applicability in strict few-shot settings where sophisticated selection strategies are infeasible. Trends are stable across seeds, with variance shrinking as b grows. Finally, structured methods require access to the full target dataset $\mathcal{D}$, which is unrealistic in scenarios such as privacy-constrained or large-scale settings.

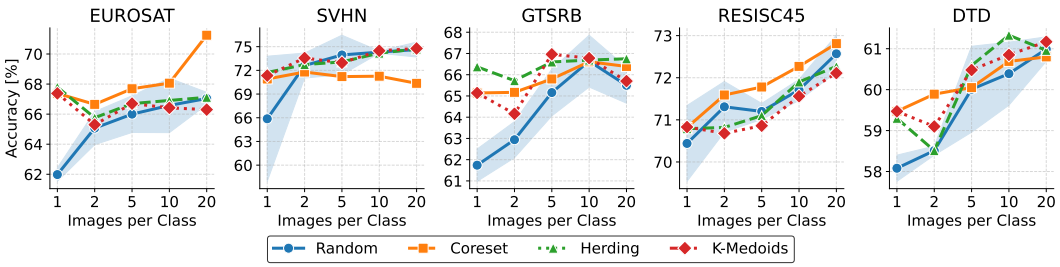

Figure 2: Accuracy at different numbers of images per class for various $\mathcal{D}_s$ construction heuristics.

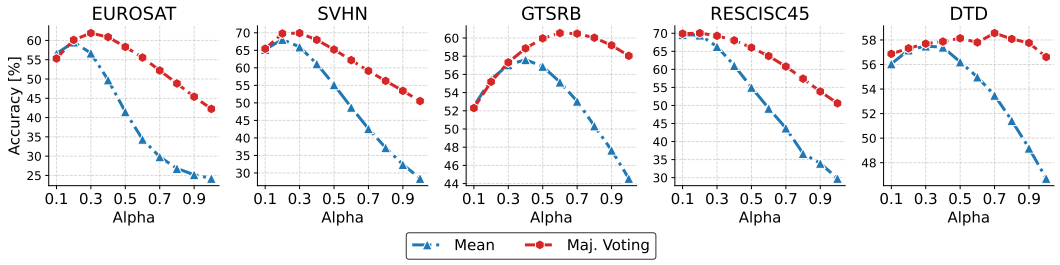

Figure 3: Accuracy at different α values for mean and majority voting sign selections.

5.4 Sensitivity to the Scaling Coefficient

We investigate the sensitivity of masked transport to the scaling factor $\alpha \in (0, 1]$, providing a proxy for how compatible and robust the transported task vector is with the target backbone. In addition to our proposed *majority voting* strategy, we consider a baseline where the estimated sign is taken as the sign of the averaged gradients (*mean*). Results are reported in Fig. 3.

Across datasets, majority voting consistently outperforms the mean strategy for all values of α, providing a more reliable approximation of the true gradient sign. Notably, majority voting yields smooth performance curves without sudden drops, and maintains higher accuracy over a broader range of α. This difference arises from the aggregation mechanism; averaging gradients before thresholding is highly sensitive to variance and outliers, so even a small subset of misaligned samples can flip the estimated sign and destabilize updates as α grows. Majority voting, instead, depends only on the relative frequency of signs, which concentrates rapidly around the true direction with increasing samples (as shown in Appendix A). As a result, it is inherently more stable and preserves transfer accuracy even in few-shot or noisy regimes.

From a practical perspective, this robustness means that masked transport with majority voting does not require fine-grained tuning of α to achieve good performance. The method remains effective across a wide range of scaling choices, which is particularly valuable when adapting to new datasets where validation data or tuning budgets are limited.

6 Conclusions & Future Works

In this work, we show that the sign structure of gradients provides a powerful and reliable proxy for the ideal directions of the loss landscape of a fine-tuned model. The exceptional performance of our oracle-based approach, which uses the signs of the true task vector, validates this core insight, confirming that effective transfer is possible when the transported task vector is aligned with the target model's local loss geometry. **GradFix** successfully approximates this oracle, achieving significant gains and outperforming naive transfer by using only a handful of labeled samples to estimate gradient signs. While our approach is highly effective and robust in low-data regimes, the remaining gap between our method and the oracle highlights clear avenues for future research. Specifically, future work could focus on developing more advanced strategies for estimating gradient signs to better approximate the ideal oracle, exploring different data selection heuristics, or applying this framework to other architectures and transfer learning settings.

Reproducibilty Statement

We provide the codebase in supplementary material to replicate of our results. All hyperparameters used in our experiments are detailed in the appendix. Additionally, we provide complete proofs for all claims made in the paper, ensuring that our results and statements can be independently verified.

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 2017.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 2017.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 2024.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 2018.

Stanford NLP Group et al. The stanford natural language inference (snli) corpus, 2022.

Nick Harvey. Near-optimal herding. In *International Conference on Machine Learning*, 2014.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer Fusion with Optimal Transport. In *International Conference on Learning Representations*, 2024.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *International Conference on Machine Learning*, 2019.

Leonard Kaufman and Peter J. Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Haoling Li, Xin Zhang, Xiao Liu, Yeyun Gong, Yifan Wang, Qi Chen, and Peng Cheng. Enhancing large language model performance with gradient-based parameter selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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 *International Conference on Machine Learning*, 2025.

Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas-merging models with permutations and least squares. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2025.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems Workshops*, 2011.

OpenAI, Josh Achiam, and Steven Adler et al. Gpt-4 technical report. *arXiv preprint arXiv: 2303.08774*, 2023.

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*, 2023.

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D. Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. In *Advances in Neural Information Processing Systems*, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context, 2016.

Chanho Park, H Vincent Poor, and Namyoon Lee. Signsgd with federated voting. *arXiv preprint arXiv:2403.16372*, 2024.

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 *International Conference on Machine Learning*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.

Filippo Rinaldi, Giacomo Capitani, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli, Elisa Ficarra, Emanuele Rodolà, Simone Calderara, and Angelo Porrello. Update your transformer to the latest release: Re-basin of task vectors. *International Conference on Machine Learning*, 2025.

Ozan Sener and Vladlen Koltun. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2018.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *International Conference on Learning Representations*, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Siqi Zeng et al. Efficient model editing with task vector bases: A theoretical and empirical study. In *International Conference on Machine Learning*, 2024.

Luca Zhou, Daniele Solombrino, Donato Crisostomi, Maria Sofia Bucarelli, Fabrizio Silvestri, and Emanuele Rodolà. Atm: Improving model merging by alternating tuning and merging. *arXiv preprint arXiv:2411.03055*, 2024.

APPENDIX

A GRADIENT SIGN ESTIMATOR GUARANTEE

We formalize why majority-vote estimation of gradient signs provides a reliable proxy for the true gradient direction in the limited-data regime.

Lemma (Majority-Vote Gradient Sign). *Let $g_i := \nabla_{\theta_i} \mathcal{L}_B(\theta_B)$ denote the true gradient of the target loss with respect to parameter θ_B, and let $g_i^{(n)}$ be i.i.d. per-sample gradients:*

$$g_i^{(n)} = g_i + \varepsilon_i^{(n)}, \quad \mathbb{E}[\varepsilon_i^{(n)}] = 0, \tag{15}$$

where $\varepsilon_i^{(n)}$ is symmetric noise around zero. Define

$$p_i := \Pr\left[\operatorname{sign}(g_i^{(n)}) = \operatorname{sign}(g_i)\right], \tag{16}$$

where p_i represents the probability that the sign of a single per-sample gradient $g_i^{(n)}$ matches the true gradient g_i.

Then, for all i with $g_i \neq 0$, $p_i > 1/2$.

In particular, the majority-vote estimator:

$$\hat{s}_i := \operatorname{sign}\left(\sum_{n=1}^{N} \operatorname{sign}(g_i^{(n)})\right) \tag{17}$$

recovers the correct sign with probability, via Hoeffding's inequality, of:

$$\Pr[\hat{s}_i = \operatorname{sign}(g_i)] \geq 1 - \exp\left(-2N(p_i - 1/2)^2\right). \tag{18}$$

Proof. We divide the proof into two parts.

Step 1: Bias of single sample signs.

Define the indicator random variable $X_n := \mathbb{1}\{\operatorname{sign}(g_i^{(n)}) = \operatorname{sign}(g_i)\} \in \{0, 1\}$. The success probability of a single sample is $p_i = \Pr[X_n = 1] = \Pr[\operatorname{sign}(g_i^{(n)}) = \operatorname{sign}(g_i)]$.

Without loss of generality, assume $g_i > 0$. The event of a successful sign match is $g_i^{(n)} > 0$, which can be rewritten as $g_i + \varepsilon_i^{(n)} > 0$, or $\varepsilon_i^{(n)} > -g_i$. Since the noise $\varepsilon_i^{(n)}$ is symmetric around zero, we know that $\Pr[\varepsilon_i^{(n)} > 0] = \Pr[\varepsilon_i^{(n)} < 0] = 1/2$. Because $g_i > 0$, the interval $(-g_i, 0)$ is non-empty. The probability of the noise falling into this interval, $\Pr[-g_i < \varepsilon_i^{(n)} < 0]$, is positive; therefore, the total probability of success is:

$$\begin{aligned} p_i &= \Pr[\varepsilon_i^{(n)} > -g_i] \\ &= \Pr[\varepsilon_i^{(n)} > 0] + \Pr[-g_i < \varepsilon_i^{(n)} < 0] \\ &= 1/2 + \Pr[-g_i < \varepsilon_i^{(n)} < 0] \end{aligned}$$

Since $\Pr[-g_i < \varepsilon_i^{(n)} < 0] > 0$, it follows that $p_i > 1/2$.

Step 2: Concentration of majority vote.

The majority-vote estimator $\hat{s}_i$ succeeds if

$$\sum_{n=1}^{N} X_n > N/2. \tag{19}$$

Now, we bound the probability of failure, which is the event that the sum of correct sign estimates is less than or equal to $N/2$. We can express this event as a deviation from the expected value of the

sum. The expected value of the sum is $\mathbb{E}\left[\sum_{n=1}^{N} X_n\right] = \sum_{n=1}^{N} \mathbb{E}[X_n] = Np_i$. Thus, the deviation is:

$$\sum_{n=1}^{N} X_n - \mathbb{E}\left[\sum_{n=1}^{N} X_n\right] = \sum_{n=1}^{N} X_n - Np_i \tag{20}$$

If we rewrite the event of failure, $\sum_{n=1}^{N} X_n \leq N/2$, in terms of this deviation we obtain:

$$\sum_{n=1}^{N} X_n - Np_i \leq N/2 - Np_i = -N(p_i - 1/2) \tag{21}$$

According to Hoeffding's inequality (Hoeffding, 1963) for a sum of i.i.d. random variables $X_n \in [0, 1]$, we have:

$$\Pr\left(\sum_{n=1}^{N} X_n - Np_i \leq -N(p_i - 1/2)\right) \leq \exp\left(-\frac{2\left(N(p_i - 1/2)\right)^2}{\sum_{n=1}^{N}(1-0)^2}\right) \tag{22}$$

The denominator simplifies to $\sum_{n=1}^{N} 1^2 = N$. Substituting this back into the inequality gives:

$$\Pr\left[\sum_{n=1}^{N} X_n \leq N/2\right] \leq \exp\left(-\frac{2N^2(p_i - 1/2)^2}{N}\right) = \exp\left(-2N(p_i - 1/2)^2\right) \tag{23}$$

The probability of correct recovery is the complement of this failure probability:

$$\Pr[\hat{s}_i = \text{sign}(g_i)] = \Pr\left[\sum_{n=1}^{N} X_n > N/2\right] = 1 - \Pr\left[\sum_{n=1}^{N} X_n \leq N/2\right] \tag{24}$$

Therefore, we obtain the final bound:

$$\Pr[\hat{s}_i = \text{sign}(g_i)] \geq 1 - \exp\left(-2N(p_i - 1/2)^2\right) \tag{25}$$

□

This result formalizes the intuition that, under mild assumptions on per-sample gradient noise, the majority-vote sign over a small batch of samples provides a reliable approximation to the true gradient direction. The probability of correct recovery grows exponentially with both the number of samples N and the signal-to-noise ratio $p_i - 1/2$. In practice, this guarantees that even very few labeled samples suffice to construct a gradient-sign mask that preserves most of the descent-aligned components of the source task vector.

B Additional Results

B.1 Sign Agreement Analysis

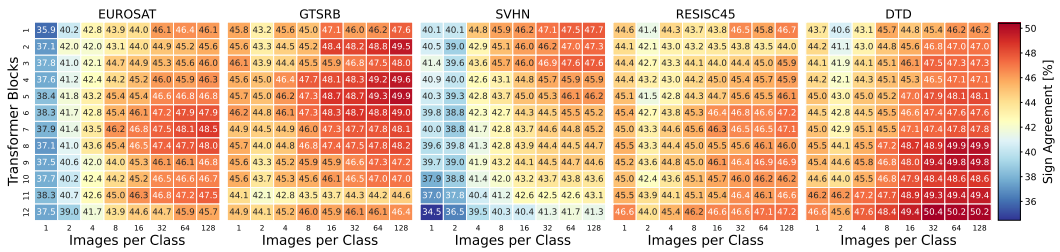

Figure 4: Sign agreement per block for ViT-B/16. The heatmaps show the percentage of sign agreements between the $\boldsymbol{m}^*$ and $\boldsymbol{m}$ constructed by computing the signs of the gradients at different $|D_s|$ budgets.

To understand the nature of the misalignment between source and target models, we show in Fig. 4 a heatmap of gradient sign agreement. Specifically, we computed the element-wise agreement between the signs of the target task vector (τ_B) and the estimated gradients at different data budgets.

This heatmap reveals that agreement is not uniform across different layers, and there is no simple, global sign correlation. This observation provides a direct explanation for the failure of naive task vector transfer. Without a mechanism to correct for this misalignment, simply adding the task vector introduces harmful directions that degrade performance. Our method, in contrast, actively addresses this by using the target model's gradients to construct a mask, ensuring that only the useful components of the task vector are transferred and aligned with the target loss landscape. Tabs. 5 and 6 demonstrate that δ^A consistently matches or outperforms θ_B^{opt} across both ViT-B/16 and ViT-L/14 models, even at larger $|D_s|$ budgets. Moreover, δ^A exhibits substantially lower standard deviation across seeds, confirming its robustness to the random choice of supervision data and its efficiency compared to direct fine-tuning under identical data constraints.

B.2 Hypeparameters selection

We evaluated the optimal task vector application coefficient α using the validation set of each dataset. For δ^*, the optimal α is equal to 1 across all datasets. In Tab. 4, we summarize the coefficients corresponding to the optimal performance of δ^A for each dataset. θ_B^{opt} are obtained for each dataset by training with the AdamW optimizer (learning rate $1\text{e}{-5}$) on the corresponding dataset D_s used to compute the gradient mask $\boldsymbol{m}$, with a single step of gradient descent.

Architecture	**EUROSAT**	**SVHN**	**GTSRB**	**RESISC45**	**DTD**
ViT-B/16	0.3	0.3	0.6	0.2	0.7
ViT-L/14	0.4	0.5	0.5	0.2	0.5

Architecture	**SNLI**	**MNLI**	**RTE**	**QNLI**	**SCITAIL**
T5v1.1	0.3	0.1	0.2	0.1	0.7

Table 4: Optimal hyperparameters for δ^A on different CLIP and T5 architectures.

B.3 Additional vision results

Table 5: Cross-dataset performance of ViT-B/16 (A: `datacomp xl s13b b90k`, B: `laion2b s32b b82k`) models averaged across different seeds initializations

Model	$\lvert\mathcal{D}_s^c\rvert$	**EUROSAT**	**RESISC45**	**GTSRB**	**SVHN**	**DTD**	**AVG**
θ_B *zero-shot*		49.41	67.98	48.29	50.58	55.96	54.45±7.30
θ_B *fine-tune*		98.70	95.66	98.65	97.45	83.19	94.73±5.94
$\theta_B + \tau_A$		49.58	67.87	49.31	50.84	56.27	54.78±7.05
$\theta_B + \delta^*$ *(oracle)*		95.06	87.06	82.92	92.04	71.44	85.71±8.30
θ_B^{opt}	1	56.61±6.06	69.25±0.96	56.08±2.87	61.32±4.09	56.21±0.88	59.89±6.05
$\theta_B - \delta^A$	1	61.94±0.43	70.05±0.56	60.88±2.85	71.07±1.82	58.32±0.30	64.45±5.47
θ_B^{opt}	2	59.49±1.43	71.20±1.13	61.70±0.76	62.01±4.40	57.00±0.54	62.29±5.31
$\theta_B - \delta^A$	2	65.07±1.10	71.42±0.90	64.33±1.05	70.19±4.55	58.51±0.10	65.96±5.06
θ_B^{opt}	5	61.99±7.29	73.01±0.48	63.08±1.41	67.03±3.93	59.65±0.80	64.95±5.81
$\theta_B - \delta^A$	5	66.05±1.21	71.57±0.88	66.61±0.42	73.59±0.82	60.02±0.20	67.57±4.96
θ_B^{opt}	10	59.98±3.77	72.27±1.65	64.54±1.12	67.85±1.02	60.96±0.33	65.12±4.97
$\theta_B - \delta^A$	10	66.59±1.83	72.05±0.59	66.02±1.59	74.82±1.22	60.18±0.40	67.93±5.38
θ_B^{opt}	20	60.59±3.94	74.22±0.66	65.51±0.80	67.19±0.65	62.59±0.08	66.02±5.10
$\theta_B - \delta^A$	20	67.05±0.41	72.29±0.14	66.42±0.47	74.11±0.59	60.92±0.08	68.15±4.84
θ_B^{opt}	50	58.80±2.45	75.88±0.83	64.91±0.53	67.58±2.91	64.13±0.11	66.26±5.97
$\theta_B - \delta^A$	50	66.94±0.46	72.26±0.22	66.13±0.14	74.07±1.52	61.35±0.03	68.15±4.75

Table 6: Cross-dataset performance of ViT-L/14 (A: `datacomp xl s13b b90k`, B: `laion2b s32b b82k`) models averaged across different seeds initializations

Model	$\lvert\mathcal{D}_s^c\rvert$	EUROSAT	RESISC45	GTSRB	SVHN	DTD	AVG
θ_B *zero-shot*		62.80	73.12	56.12	37.28	63.35	58.53±12.35
θ_B *fine-tune*		98.95	97.06	99.16	97.80	83.56	95.31±6.13
$\theta_B + \tau_A$		62.77	73.49	56.03	39.09	63.56	58.99±11.80
$\theta_B + \delta^*$ *(oracle)*		96.75	90.30	88.65	92.60	72.66	88.19±8.52
θ_B^{opt}	1	64.65±5.90	74.54±0.57	63.97±4.50	62.51±4.58	63.76±0.27	65.89±5.61
$\theta_B - \delta^A$	1	69.67±1.44	76.45±1.33	66.82±0.84	70.15±5.18	65.50±0.74	69.72±4.46
θ_B^{opt}	2	70.76±1.77	76.62±0.26	69.91±1.89	45.23±1.87	64.97±0.21	65.50±11.23
$\theta_B - \delta^A$	2	74.10±2.00	76.97±0.68	71.55±2.73	54.31±2.54	66.10±0.59	68.61±8.44
θ_B^{opt}	5	69.75±1.64	75.41±2.94	73.25±0.62	67.11±2.39	66.72±0.13	70.45±3.86
$\theta_B - \delta^A$	5	75.59±2.24	76.82±0.48	73.14±1.23	74.41±2.22	66.95±0.69	73.31±3.88
θ_B^{opt}	10	70.36±3.82	78.07±1.76	74.11±0.31	60.43±3.68	69.36±0.27	70.46±6.45
$\theta_B - \delta^A$	10	73.74±1.58	77.59±0.57	74.94±0.73	75.88±2.80	67.41±0.48	73.77±3.86
θ_B^{opt}	20	78.74±2.96	80.77±1.17	74.65±0.28	65.99±2.12	71.40±0.35	74.31±5.65
$\theta_B - \delta^A$	20	74.87±0.71	78.16±0.33	74.90±0.55	75.79±0.90	67.55±0.24	74.15±3.83
θ_B^{opt}	50	77.07±1.66	82.16±0.31	75.38±0.45	67.83±1.64	73.81±0.08	75.25±4.90
$\theta_B - \delta^A$	50	74.75±0.89	78.27±0.18	74.61±0.91	76.79±1.26	67.77±0.01	74.27±3.86

C Datasets and Supervision Proportions

In this section, we provide detailed information about the datasets used in our experiments and compute, for each one, the supervision proportions corresponding to our subset budgets $|\mathcal{D}_s|$. Recall that $|\mathcal{D}_s|$ denotes the number of labeled examples *per class* used to estimate gradient signs. The resulting percentages indicate what fraction of the full training set those few-shot budgets represent.

C.1 Visual Datasets

- **EuroSAT**: A dataset based on Sentinel-2 satellite images covering 13 spectral bands, consisting of 27 000 labeled and geo-referenced samples across 10 classes (Helber et al., 2019).
- **SVHN**: A real-world image dataset from Google Street View house numbers, containing 73 257 labeled digits across 10 classes (0–9) (Netzer et al., 2011).
- **GTSRB**: The German Traffic Sign Recognition Benchmark, comprising 39 209 training images and 12 630 test images across 43 classes (Stallkamp et al., 2011).
- **RESISC45**: A scene classification dataset with 31 500 RGB images 256×256 from Google Earth, covering 45 scene classes with 700 images per class (Cheng et al., 2017).
- **DTD**: The Describable Textures Dataset, consisting of 5640 images organized into 47 categories inspired by human perception (Cimpoi et al., 2014).

Table 7: Supervision proportions for visual datasets. $|\mathcal{D}_s|$ denotes examples per class. Each cell shows the total dataset percentage.

			$\lvert\mathcal{D}_s\rvert$					
Dataset	**# Samples**	**Classes**	1	2	5	10	20	50
EUROSAT	27,000	10	0.04%	0.07%	0.19%	0.37%	0.74%	1.85%
SVHN	73,257	10	0.01%	0.03%	0.07%	0.14%	0.27%	0.68%
GTSRB	39,209	43	0.11%	0.22%	0.55%	1.10%	2.19%	5.48%
RESISC45	31,500	45	0.14%	0.29%	0.71%	1.43%	2.86%	7.14%
DTD	5,640	47	0.83%	1.66%	4.15%	8.30%	16.60%	41.49%

C.2 Textual Datasets

- **SNLI**: Stanford Natural Language Inference dataset, containing 570 000 sentence pairs labeled for entailment, contradiction, or neutral (Group et al., 2022).
- **MNLI**: Multi-Genre Natural Language Inference dataset, comprising 433 000 sentence pairs annotated with textual entailment information across various genres (Williams et al., 2018).

- **RTE**: Recognizing Textual Entailment dataset, with 2490 examples for training, 277 for validation, and 3000 for testing, divided into two classes (Wang et al., 2018).
- **QNLI**: Question Natural Language Inference dataset, containing 104 743 training examples divided into two classes (Wang et al., 2018).
- **SCITAIL**: A science entailment dataset built from science question answering, with 23 596 training examples divided into two classes (Khot et al., 2018).

Table 8: Supervision proportions for textual datasets. $|\mathcal{D}_s^c|$ denotes examples per class. Each cell shows the total dataset percentage.

Dataset	**# Samples**	**Classes**	$\frac{\lvert\mathcal{D}_s^c\rvert}{50}$
SNLI	570,000	3	0.03%
MNLI	433,000	3	0.03%
RTE	2,490	2	4.02%
QNLI	104,743	2	0.10%
SCITAIL	23,596	2	0.42%