🎯 DARTS: Decoder-Aware Representation Tuning via Surgery for Model Merging

Anonymous ACL submission

Abstract

Model merging combines multiple task-specific fine-tuned LLMs into a single multi-task model without additional training. However, merged models are known to suffer from *representation bias*: systematic drift between the merged model's hidden states and those of each individual source model. Prior work (Yang et al., 2024a) study and mitigate this bias for encoder-based vision models using a lightweight correction module trained with L1 loss. However, such bias is ill-defined for decoder models due to their autoregressive nature. We first formally define the notion of causal representation bias for decoder models, and show two challenges absent in encoders: (1) the causal attention mask causes bias to **accumulate** across token positions, requiring position-dependent correction; and (2) not all token positions are equally important, i.e., high-entropy (decision-critical) positions matter far more than low-entropy ones. To address these challenges, we propose **D**ecoder-**A**ware **R**epresentation **T**uning via **S**urgery (DARTS). DARTS employs a novel entropy-weighted L1 loss to upweight correction at high-entropy positions where errors most affect generation quality, and a per-position additive bias that captures position-dependent error without overparameterization. Extensive evaluation DARTS on three domains: code generation (HumanEval), mathematical reasoning (GSM8K), and instruction following (AlpacaEval) on Llama-2-7B models, and shows DARTS achieves significant improvement over the standard surgery approach while adding negligible parameters (0.1% of total parameters).

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, from code generation (Chen et al., 2021; Rozière et al., 2023) and mathematical reasoning (Cobbe et al., 2021; Luo et al., 2023; Wei et al., 2022) to open-ended instruction following (Ouyang et al., 2022). However, these capabilities are typically obtained through task-specific fine-tuning (Hu et al., 2022; Touvron et al., 2023), producing a proliferation of specialized models, each of which excels on its target domain but requires independent deployment. As the number of specialized models grows, so does the cost of serving them, making multi-task consolidation increasingly attractive.

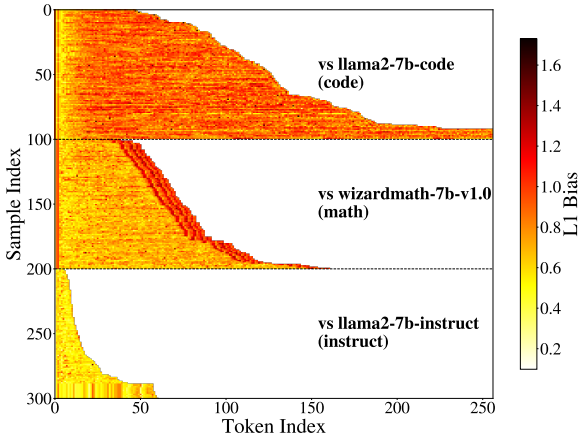

Figure 1: Per-token L1 representation bias between a merged Llama-2-7B model and each individual fine-tune, measured at the final transformer layer. Each row is a calibration sample; each column is a token position. Across all three domains (code, math, instruct), we see an increasing trend of bias with token position due to error accumulation under the causal attention mask. This position-dependent structure motivates the decoder-aware corrections proposed in DARTS.

Model merging (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Yang et al., 2024b; Yadav et al., 2024) offers an elegant solution to this problem: combine the weights of multiple fine-tuned models into a single model that inherits capabilities from all source models, *without any additional training data or computation*. Methods such as Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and DARE (Yu

et al., 2024) have shown that simple arithmetic operations in weight space (such as adding scaled task vectors, resolving sign conflicts, or dropping redundant parameters) can produce surprisingly effective multi-task models. This has made model merging a standard tool in the open-source LLM ecosystem (Goddard et al., 2024), where practitioners routinely merge instruction-tuned, code-specialized, and math-specialized models into unified checkpoints.

Despite this progress, a substantial performance gap persists between merged models and their individual fine-tuned counterparts. Merged models frequently degrade on one or more constituent tasks (Ilharco et al., 2023; Yadav et al., 2023; Yang et al., 2024b), and the severity of this degradation varies unpredictably across tasks, merging methods, and model scales. Understanding *why* merging hurts performance, and finding principled ways to close the gap, remains an open challenge.

Yang et al. (2024a) recently offered a compelling explanation for this degradation. By visualizing the hidden representations of merged vision models (ViT-B/32, ViT-B/16, ViT-L/14 on eight classification tasks), they demonstrated that merged models suffer from **representation bias**: a systematic discrepancy between the feature representations extracted by the merged model and those of each individual fine-tuned model. Crucially, they showed that representation bias *correlates* with task performance: merging methods that produce smaller representation bias (e.g., AdaMerging (Yang et al., 2024b)) consistently outperform those with larger bias (e.g., Weight Averaging). Building on this insight, they proposed *representation surgery*: a lightweight low-rank module trained with L1 loss to predict and subtract the representation bias, achieving significant performance gains across all merging methods.

However, the original surgery framework was designed exclusively for **encoder-based vision models**. In ViT architectures (Dosovitskiy et al., 2021), bidirectional self-attention ensures that each token (image patch) is contextualized by all other tokens equally, producing representation bias that is roughly *uniform* across token positions. A position-agnostic surgery module, which applies the same correction at every position, is well-suited to this setting. Likewise, the uniform L1 loss used for training treats all token positions as equally important, which is reasonable when every patch contributes similarly to the classification decision.

Decoder-based LLMs are fundamentally different. The autoregressive transformer architecture (Vaswani et al., 2017; Brown et al., 2020) processes tokens left-to-right under a causal attention mask: token t can only attend to tokens $0, 1, \ldots, t-1$. This architectural constraint has two consequences for representation bias that render the original surgery framework inadequate:

First, representation bias in decoder models is position-dependent. Because each token's representation is computed from a strictly causal context, any bias introduced by weight-space merging at earlier positions propagates forward through the attention mechanism. Later tokens aggregate not only the original merging error at their own position but also the accumulated errors from all preceding positions. Figure 1 empirically shows this effect: plotting per-token L1 distance between merged and individual model representations reveals a clear *increasing* trend, with later positions exhibiting 2–3$\times$ higher bias than early positions. A position-agnostic surgery module, which applies identical corrections at every position, is structurally incapable of capturing this pattern, forcing it to compromise between under-correcting later tokens and over-correcting earlier ones.

Second, not all token positions are equally important for generation quality. In autoregressive LLMs, the model's prediction at each position determines the next token in the generated sequence. At some positions, the model is highly confident and assigns most probability mass to a single token (e.g., closing brackets, deterministic syntax, common phrases). At other positions, the model distributes probability across many plausible continuations; these are the *decision-critical* positions where the choice of next token meaningfully affects the quality and correctness of the generated output. Small hidden-state errors at high-entropy positions can flip the argmax prediction and cascade through subsequent tokens, while comparable errors at low-entropy positions are absorbed without consequence. The standard L_1 loss (Yang et al., 2024a) treats all positions uniformly, wasting the surgery module's limited correction capacity on positions that do not affect downstream generation.

To address these decoder-specific challenges, we propose **D**ecoder-**A**ware **R**epresentation **T**uning via **S**urgery (DARTS), which adapts representation surgery to the autoregressive setting through two complementary innovations. First, we introduce **entropy-weighted L_1 loss (EWL1)**, which

weights the per-position correction penalty by the target model's prediction entropy, focusing the surgery module's capacity on decision-critical positions where errors most affect generation quality. Second, we augment the low-rank content correction with a **position correction module**, a direct per-position additive bias table that learns independent correction vectors for each token position, capturing the structured position-dependent bias without network overparameterization. We evaluate DARTS on Llama-2-7B (Touvron et al., 2023) models merged from three specialized fine-tunes (code generation, mathematical reasoning, and instruction following), showing that it achieves significant improvement over the original surgery approach.

The main contributions of this paper are as follows:

1. We formally define causal representation bias for decoder models and empirically demonstrate two challenges absent in encoder models: **position-dependent bias accumulation** under the causal attention mask, and **non-uniform position importance** arising from the autoregressive generation process.

2. To address these challenges, we propose DARTS, which uses a novel **entropy-weighted L_1 loss** and a **position correction module**, to upweights correction at decision-critical positions, and complement content-based correction with per-position bias terms.

3. We perform extensive experiments to show that DARTS reduces causal representation bias and significantly improves MTL performance on a variety of different tasks while adding merely 0.1% of parameters to the model.

2 Related Work

Model Merging for Multi-Task Learning. Model merging has emerged as a practical alternative to traditional multi-task learning (Crawshaw, 2020; Zhang and Yang, 2021), which requires centralized data collection and joint training. Early work showed that simply averaging model weights can improve generalization for models trained on the *same* task (Wortsman et al., 2022), and subsequent methods extended this idea to merging models trained on *different* tasks. Task Arithmetic (Ilharco et al., 2023) introduced the concept of *task vectors*, defined as the difference $\tau_i = \theta_i - \theta_{\text{pre}}$ between fine-tuned and pretrained weights, and showed that adding scaled task vectors to the pretrained model produces effective multi-task models. TIES-Merging (Yadav et al., 2023) addressed parameter sign conflicts and redundancy by trimming, electing signs, and merging only aligned components. DARE (Yu et al., 2024) randomly drops delta parameters and rescales survivors to reduce interference. Fisher-Merging (Matena and Raffel, 2022) uses Fisher information to weight parameter importance, while RegMean (Jin et al., 2023) reweights based on training statistics. More recently, AdaMerging (Yang et al., 2024b) learns task-wise or layer-wise merging coefficients, Concrete (Tang et al., 2023) finds shared subspaces, and ZipIt (Stoica et al., 2024) merges models by aligning intermediate features. A comprehensive survey of these methods can be found in Yadav et al. (2024). All of the above methods operate exclusively in weight space and do not explicitly address the resulting representation-level discrepancies.

Representation Surgery. Yang et al. (2024a) identified representation bias as a key obstacle in model merging: the hidden representations of merged ViT models (Dosovitskiy et al., 2021; Radford et al., 2021) diverge systematically from those of individual fine-tuned models, and this divergence correlates with task performance degradation. They proposed a lightweight adapter-like (Houlsby et al., 2019) correction module $\Phi(Z) = W_{\text{up}} \cdot \text{ReLU}(W_{\text{down}} \cdot Z)$ trained with L_1 loss to subtract the bias from merged representations. This approach was validated on eight image classification tasks across three ViT architectures, achieving consistent gains when applied on top of Weight Averaging, Task Arithmetic, TIES-Merging, and AdaMerging. However, the original framework assumes position-agnostic correction and uniform loss weighting. These assumptions are well-motivated for bidirectional encoder models but, as we demonstrate, inadequate for autoregressive decoder models.

Parameter-Efficient Fine-Tuning. LoRA (Hu et al., 2022) adds low-rank weight update matrices for parameter-efficient fine-tuning, and subsequent work has explored prefix tuning (Li and Liang, 2021), adapters (Houlsby et al., 2019), and other approaches. Our surgery module shares the low-rank bottleneck structure with LoRA and adapters, but operates in a fundamentally different setting: it corrects *hidden-state* representations (in representation space) rather than adapting model *weights*

(in weight space), and is trained to remove merging artifacts rather than to learn new task capabilities. The surgery module is applied only at the final transformer layer and adds negligible inference overhead.

3 Causal Representation Bias in Model Merging

We begin by establishing notation and defining the problem (Section 3.1), then empirically characterize the position-dependent structure of representation bias in decoder models (Section 3.3). We then present the two core components of DARTS: the position correction module (Section 3.4) and entropy-weighted L_1 loss (Section 3.5).

3.1 Preliminaries

3.1.1 Notation

We consider K task-specific models obtained by fine-tuning a shared pretrained model on tasks $\mathcal{T}_1, \ldots, \mathcal{T}_K$. Let M denote the merged model and M_k^* denote the k-th individual fine-tune (the per-task "target"). For an input sequence $x_{1:T}$ tokenized to length T, we denote:

- $h_t^M \in \mathbb{R}^H$: the merged model's final-layer hidden state at position t
- $h_t^{*k} \in \mathbb{R}^H$: target model k's corresponding hidden state
- $W_U^{*k} \in \mathbb{R}^{V \times H}$: target model k's LM head (unembedding matrix)

with $H = 4096$ (hidden size, Llama-2-7B), $V = 32{,}000$ (vocabulary size), and $T_{\max} = 512$ (calibration max sequence length). For each task k, we learn a surgery module $\mathcal{S}_k$ that produces a corrected hidden state $\hat{h}_t = \mathcal{S}_k(h^M)_t$. We drop the k subscript where clear from context.

3.1.2 Task Vectors and Merging

Given a pretrained model θ_{pre} and a model fine-tuned on task k, θ_k, the task vector is $\tau_k = \theta_k - \theta_{\text{pre}}$. Task Arithmetic (Ilharco et al., 2023) merges K tasks via:

$$\theta_{\text{merged}} = \theta_{\text{pre}} + \lambda \sum_{k=1}^{K} \tau_k \tag{1}$$

where λ is a scaling coefficient. Other merging methods (TIES-Merging, DARE, AdaMerging) modify how task vectors are combined but share the same general framework (Yadav et al., 2024).

3.1.3 Representation Bias

Following Yang et al. (2024a), we define the per-position representation bias as the normalized L_1 distance between the merged and target model's hidden states:

$$\delta_t = \frac{1}{H} \| h_t^M - h_t^{*k} \|_1 \tag{2}$$

In encoder models, Yang et al. showed that this bias is approximately uniform across token positions and correlates with downstream task performance. We extend this analysis to decoder models in the next section.

3.1.4 Surgery Module and Inference

At inference, corrected hidden states are projected through the **target model's LM head**:

$$z_t = W_U^{*k} \hat{h}_t, \qquad p(x_{t+1} \mid x_{\leq t}) = \text{softmax}(z_t) \tag{3}$$

The merged model's own LM head is dropped; each domain uses its target model's LM head from the original fine-tune. This ensures that the output distribution is aligned with the vocabulary geometry the target model was optimized for.

3.2 Standard Surgery (Baseline)

The original representation surgery (Yang et al., 2024a) employs a low-rank bottleneck with rank r and content correction function $c(h) = W_{\text{up}}\, \text{ReLU}(W_{\text{down}}\, h)$:

$$\hat{h}_t = h_t^M - c(h_t^M) \tag{4}$$

where $W_{\text{down}} \in \mathbb{R}^{r \times H}$ and $W_{\text{up}} \in \mathbb{R}^{H \times r}$. Following Yang et al. (2024a), W_{up} is zero-initialized so that the surgery module starts as the identity map, meaning an untrained module produces no correction. Parameter count: $2rH$. Crucially, this module is **position-agnostic**: the same parameters are applied identically at every token position t.

3.3 Position-Dependent Bias in Decoders

To examine whether the uniform-bias assumption of encoder-based surgery transfers to decoder models, we compute the per-token L_1 bias δ_t (Eq. 2) across a calibration set of prompts from each domain using Llama-2-7B merged models.

The results (Figure 1) reveal a striking pattern absent in encoder models: representation bias exhibits a clear *increasing trend* with token position. Tokens at early positions (0–50) show relatively low bias, while tokens at later positions (100–256)

exhibit significantly higher bias, often 2–3× the magnitude of early tokens. This pattern is consistent across all three domains (code, math, instruction following).

The explanation follows directly from the causal attention mechanism (Vaswani et al., 2017). In encoder models (e.g., ViT), bidirectional attention distributes merging errors uniformly across all tokens. In decoder models, token t attends only to the preceding context $\{0, 1, \ldots, t-1\}$. If earlier tokens carry representation bias from merging, token t aggregates these biased representations through attention, causing the bias to *compound*. Each successive position accumulates not only its own merging error but also the errors of all preceding positions, producing the observed monotonically increasing pattern.

This has a direct implication for surgery module design: a position-agnostic correction, which applies the same magnitude of correction at every position, must compromise between under-correcting later tokens (where bias is largest) and over-correcting earlier tokens (where bias is smallest). This motivates the position-aware correction described below.

3.4 Position Correction Module

To address the position-dependent structure of representation bias, we augment the content correction with a direct per-position additive bias $b \in \mathbb{R}^{T_{\max} \times H}$, zero-initialized:

$$\hat{h}_t = h_t^M - \underbrace{c(h_t^M)}_{\text{content (input-dep.)}} - \underbrace{b_t}_{\text{position (input-indep.)}} \tag{5}$$

The design decouples two sources of representation bias. The content correction $c(h_t^M)$ handles *input-dependent* errors, specifically the bias that varies with the specific input at position t. The position bias b_t captures the *systematic per-position offset* that is consistent across inputs, reflecting the cumulative effect of causal error accumulation documented in Section 3.3.

A key design choice is that b_t is a *free parameter* per position, rather than being parameterized by a network. Each position's correction is independent of other positions and cannot compose. This provides *structural regularization* that prevents the position correction from overfitting to the small calibration set. We explored network-parameterized alternatives (concatenation, parallel bottlenecks, scalar gating; see Appendix C), but found that the direct additive bias consistently matched or outperformed these more complex architectures while requiring no additional hyperparameters.

Parameter count. $2rH + T_{\max}H$. For Llama-2-7B (H=4096, r=16, $T_{\max}$=512): ≈2.2M per domain, ≈6.6M total for 3 domains (≈0.1% of 7B).

3.5 Entropy-Weighted L1 Loss

The standard surgery objective (Yang et al., 2024a) minimizes the L_1 distance between corrected and target representations, treating all token positions equally:

$$\mathcal{L}_{\text{L1}} = \frac{1}{H\,|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|\hat{h}_t - h_t^*\|_1 \tag{6}$$

where $\mathcal{T}$ is the set of non-padded positions. This uniform weighting is well-suited to classification tasks where all patch positions contribute similarly to the final prediction. In autoregressive generation, however, not all positions matter equally.

We propose weighting each position by the target model's prediction entropy. Let $p^*_{t,v} = \mathrm{softmax}(W_U^* h_t^*)_v$ be the target model's predicted distribution at position t, and let $\mathcal{H}_t = -\sum_v p^*_{t,v} \log p^*_{t,v}$ denote its entropy. The entropy-weighted L_1 loss is:

$$\mathcal{L}_{\text{EWL1}} = \frac{1}{H\,|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{\mathcal{H}_t}{\bar{\mathcal{H}}} \|\hat{h}_t - h_t^*\|_1 \tag{7}$$

where $\bar{\mathcal{H}} = \frac{1}{|\mathcal{T}|} \sum_{t'} \mathcal{H}_{t'}$ is the mean entropy. The unit-mean normalization ensures that the overall loss scale remains comparable to standard L_1, allowing the same learning rate without adjustment.

The intuition is as follows. At high-entropy positions, where the target model distributes probability across many plausible next tokens, small hidden-state errors can flip the argmax prediction, changing the token that would be generated and cascading through subsequent autoregressive steps. At low-entropy positions (e.g., deterministic syntax, common phrases, punctuation), the model's prediction is robust to perturbation: even a moderate hidden-state error leaves the argmax unchanged. By upweighting high-entropy positions, EWL1 focuses the surgery module's limited correction capacity on the positions that actually affect generation quality, analogous to how importance-weighted distillation (Hinton et al., 2015) focuses on informative training examples.

4 Experimental Setup

4.1 Models and Merging

We use Llama-2-7B (Touvron et al., 2023) (H=4096, V=32,000) as the pretrained base and merge three publicly available fine-tunes: Llama-2-7B-Instruct (Meta), WizardMath-7B-v1.0 (Luo et al., 2023), and Llama-2-7B-Code (Rozière et al., 2023).

We evaluate four families of merging methods across an extensive hyperparameter sweep totaling over 124 configurations. Task Arithmetic (Ilharco et al., 2023) merges task vectors with a scaling coefficient $\lambda \in \{0.3, 0.5, 0.7, 0.9, 1.0\}$. Weight Averaging spans 22 weight distributions, including equal weighting, single-model dominant (40–80% weight on one model), and pairwise-heavy configurations. TIES-Merging (Yadav et al., 2023) is swept over 42 configurations combining weight $\in \{0.3, 0.5, 0.7, 1.0, 1.2, 1.5\}$ and density $\in \{0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$, including both symmetric and asymmetric per-model weight assignments. DARE (Yu et al., 2024) covers 60 configurations across both `dare_ties` and `dare_linear` variants with weight $\in \{0.3, 0.5, 0.7, 1.0, 1.2\}$ and density $\in \{0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$. All merges are produced using mergekit (Goddard et al., 2024). For each merging method, we select the best-performing configuration by aggregate benchmark score as the representative baseline for that method in subsequent surgery experiments.

4.2 Evaluation Metrics

We evaluate on three benchmarks spanning complementary capabilities. For code generation, we use **HumanEval** (Chen et al., 2021) (164 problems) and report **Pass@1**, which measures the fraction of problems solved correctly on the first attempt. For mathematical reasoning, we use the test set of **GSM8K** (Cobbe et al., 2021) (1,319 problems) with chain-of-thought zero-shot prompting (Wei et al., 2022) and report **Accuracy**. For instruction following, we use **AlpacaEval** (Dubois et al., 2024) (805 instructions) and report the **Win Rate** against GPT-3.5 as judged by GPT-4.

In addition to these task-level metrics, we report L_1 **Reduction** (ρ_{L_1}) as an indicator of how effectively surgery corrects representation bias in hidden-state space. Let $\delta_t^{\text{before}} = \frac{1}{H}\|h_t^M - h_t^*\|_1$ and $\delta_t^{\text{after}} = \frac{1}{H}\|\hat{h}_t - h_t^*\|_1$ denote the per-position L_1 residuals before and after surgery. The L_1 Reduction is:

$$\rho_{L_1} = 1 - \frac{\sum_{t \in \mathcal{T}} \delta_t^{\text{after}}}{\sum_{t \in \mathcal{T}} \delta_t^{\text{before}}} \tag{8}$$

Higher ρ_{L1} indicates that a larger fraction of the representation bias has been removed. As shown by Yang et al. (2024a), this metric correlates with downstream task performance in encoder models; we examine whether the same relationship holds for decoder models in Section 6.

4.3 Implementation Details

All merged models are produced using mergekit (Goddard et al., 2024) from three Llama-2-7B fine-tunes: Llama-2-7B-Instruct, WizardMath-7B-v1.0, and Llama-2-7B-Code. Surgery modules are trained for 500 iterations with the Adam optimizer (β_1=0.9, β_2=0.999, lr $= 10^{-3}$) on 50 domain-specific calibration samples (HumanEval prompts for code, GSM8K questions for math, AlpacaEval instructions for instruct), tokenized to a maximum length of 512. Hidden states are precomputed from both frozen models in bfloat16 and cached on GPU; only the surgery parameters are updated. For evaluation, HumanEval and GSM8K are run through lm-eval-harness (Gao et al., 2024) with greedy decoding (temperature 0.0) and the HuggingFace backend; the maximum number of generated tokens is 512 for HumanEval and 512 for GSM8K (chain-of-thought, zero-shot). AlpacaEval responses are generated with sampling (temperature 0.7, top-p 0.9, max 1024 tokens) and judged pairwise against GPT-3.5-Turbo using GPT-4-Turbo as judge. All experiments are conducted on NVIDIA A100-SXM4-80GB GPUs.

4.4 Baselines

We compare four settings. **Individual models**: each fine-tuned model evaluated on all three benchmarks, establishing the upper bound for domain-specific performance. **Merged (no surgery)**: the merged model evaluated directly. **Standard + L_1** (Yang et al., 2024a): position-agnostic surgery with standard L_1 loss, representing the original approach. **DARTS** (ours): position correction module with entropy-weighted L_1 loss.

5 Results

5.1 Main Results

Table 1 presents the main results, comparing the original surgery approach (Standard + L_1) against

Merging Method	Surgery Method	HumanEval	GSM8K	AlpacaEval	Avg.
Weight Averaging	None (baseline)	27.4	31.2	37.3	32.0
	Yang et al. (2024a)	26.2	28.7	39.5	31.5
	DARTS (ours)	**26.8**	**28.9**	**41.3**	**32.3**
Task Arithmetic	None (baseline)	23.8	41.2	49.5	38.2
	Yang et al. (2024a)	23.2	42.5	47.0	37.6
	DARTS (ours)	**25.0**	**43.4**	**58.6**	**42.3**
TIES-Merging	None (baseline)	17.1	12.6	8.9	12.9
	Yang et al. (2024a)	**23.2**	7.3	12.8	**14.4**
	DARTS (ours)	21.9	**8.3**	**13.0**	**14.4**
DARE	None (baseline)	19.5	34.9	47.7	34.0
	Yang et al. (2024a)	18.3	**35.4**	47.6	33.8
	DARTS (ours)	**21.3**	34.9	**48.7**	**35.0**

Table 1: Main results on three benchmarks. Comparing the original surgery approach (Standard + L_1) against our decoder-specific adaptations. Best results per merged model in **bold**.

Model	HE	GSM	AE
Llama-2-7B-Base	12.8	4.5	1.5
Llama-2-7B-Instruct	15.2	24.5	58.2
WizardMath-7B-v1.0	12.8	39.7	7.0
Llama-2-7B-Code	28.1	5.2	0.2

Table 2: Individual model baselines. HE = HumanEval (pass@1), GSM = GSM8K (exact match), AE = AlpacaEval (win rate %).

DARTS across both merged model configurations and all three benchmarks.

6 Analysis

6.1 L1 Bias Reduction vs Task Performance

A central question is whether reducing representation bias in hidden-state space translates to downstream task performance. Table 3 reports the mean L_1 bias before and after DARTS surgery alongside the corresponding task performance changes.

6.2 Statistically Significant Improvements

To verify that the improvements of DARTS over the standard surgery baseline are not due to random variation in calibration data selection, we train both methods across 6 different random seeds (controlling the calibration data shuffle order) and compare their average scores. Table 4 reports the per-seed average performance (mean of HumanEval, GSM8K, and AlpacaEval) for each method.

We first test each group for normality using the Shapiro-Wilk test ($\alpha = 0.05$). The DARTS scores pass the normality test ($W = 0.80$, $p = 0.059$), while the Standard surgery scores do not ($W = 0.77$, $p = 0.034$). Since one group is non-normal, we apply the Mann-Whitney U test rather than a parametric t-test. The result ($U = 36.0$, $p = 0.002$) indicates that DARTS significantly outperforms Standard surgery at the $\alpha = 0.05$ level. Despite a modest absolute improvement of 2.0 percentage points in average score, the consistency of the gain across all 6 seeds (with DARTS outperforming Standard in every run) yields a highly significant p-value, confirming that the improvement is robust to calibration data variation.

7 Ablation Studies

7.1 Loss Function Ablation

Table 5 compares four loss functions using Standard (position-agnostic) surgery on TA-λ0.5, holding architecture constant to isolate the effect of the loss. From the results, we find that *loss function choice matters more than surgery module architecture* for decoder-based surgery.

7.2 Surgery Module Ablation

Table 6 compares Standard surgery against DARTS (with position correction), both using EWL1 loss, isolating the complementary contribution of the position correction module. From the results, we find adding position correction module leads to modest but significant improvements.

7.3 Learning Rate Sensitivity

Figure 2 shows DARTS performance across learning rates spanning two orders of magnitude (10^{-4} to 5×10^{-3}). Performance is remarkably stable: the average score varies by less than 2 percentage points across the entire range, and no single learning rate causes catastrophic degradation on any benchmark. The default learning rate of 10^{-3}

Merged Model	Domain	ρ_{L_1} (%)	Task Score		pp Change
			No Surg.	DARTS	
TA ($\lambda = 0.5$)	Code	8.2	23.8	25.0	+1.2
	Math	11.3	41.2	43.4	+2.2
	Instruct	42.9	49.5	58.6	+9.1
TA ($\lambda = 0.7$)	Code	14.2	21.9	26.8	+4.9
	Math	14.3	46.2	42.9	−3.3
	Instruct	51.0	49.3	59.2	+9.9

Table 3: Correlation between L_1 bias reduction and task performance. pp = percentage point change in the task metric.

Seed	DARTS (Avg.)	Standard (Avg.)
default	42.3	37.6
7	42.2	41.7
29	42.3	40.4
71	43.3	41.8
164	42.2	41.2
190	43.0	41.1
Mean ± Std	42.6 ± 0.5	40.6 ± 1.6
Mann-Whitney U test: $U = 36.0$, $p = 0.002$		

Table 4: Statistical significance of DARTS vs. Standard surgery (L_1 loss). Average score = mean of HumanEval, GSM8K, and AlpacaEval across 6 seeds.

Loss Function	HE	GSM	AE	Avg.
L_1 (original)	23.2	42.5	47.0	37.6
KL on logits	20.1	**43.4**	54.5	39.3
Cross-entropy distill.	18.3	33.1	40.0	30.5
Entropy-Weighted L_1 (ours)	**25.6**	41.7	**57.0**	**41.4**

Table 5: Loss function ablation with Standard surgery. EWL1 dominates, confirming that loss choice is the primary driver of improvement.

Module	HE	GSM	AE
Standard	25.6	41.7	57.0
+ Position corr. (ours)	**25.0**	**43.4**	**58.6**
Δ	−0.6	+1.7	+1.6

Table 6: Surgery module ablation with EWL1 loss. Position correction provides complementary gains on top of the improved loss.

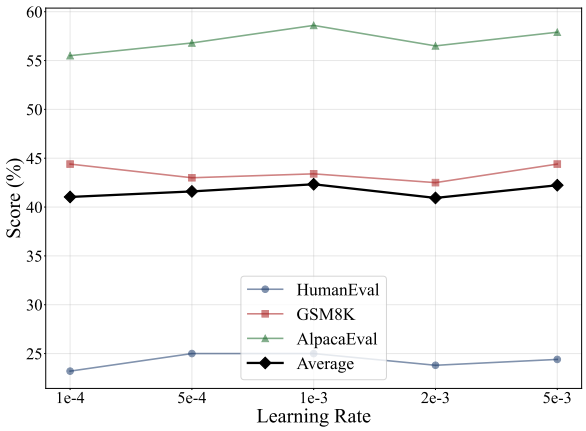

Figure 2: Learning rate sensitivity of DARTS. Performance is stable across two orders of magnitude, with the default 10^{-3} achieving the best average.

achieves the best AlpacaEval score and the highest average, but nearby values (5×10^{-4}, 5×10^{-3}) perform competitively. This robustness suggests that DARTS does not require careful learning rate tuning in practice.

8 Conclusion

We have extended representation surgery from encoder-based vision models to decoder-based language models and identified two fundamental challenges specific to the autoregressive setting: position-dependent bias accumulation under the causal attention mask, and non-uniform position importance arising from the structure of language generation. To address these challenges, we proposed DARTS, which combines entropy-weighted L_1 loss (focusing correction on decision-critical positions) with a structurally regularized position correction module that captures the monotonically increasing bias pattern. Our experiments on Llama-2-7B merged models across three diverse domains demonstrate that loss function design is the dominant factor: entropy-weighted L_1 alone accounts for the majority of performance gains, with position correction providing complementary improvements. On Llama-2-7B merged models across code, math, and instruction-following domains, DARTS achieves significant improvement over the original surgery approach.

Limitations

1. **Sequence length generalization.** The position correction bias table $b \in \mathbb{R}^{T_{\max} \times H}$ cannot

extrapolate beyond $T_{\max} = 512$. Positions exceeding $T_{\max}$ receive zero position correction (content correction still applies). Future work could explore continuous position functions or relative position encodings.

2. **Scale.** We evaluate exclusively on Llama-2-7B. Whether DARTS transfers to larger models (13B, 70B) or different architectures (Mistral (Jiang et al., 2023)) remains untested.

3. **Domain asymmetry.** The entropy weighting may deprioritize positions critical for code generation (e.g., syntactically deterministic but semantically critical tokens), potentially explaining domain-dependent gains.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meade, Hailey Zeng, and Jacob Solawetz. 2024. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations (ICLR)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1

others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*.

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. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, and Dacheng Tao. 2023. Concrete subspace learning based interference elimination for multi-task model fusion. In *arXiv preprint arXiv:2312.06173*.

Hugo Touvron, Louis Martin, Kevin Stone, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Prateek Yadav, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2024. A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*.

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

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024a. Representation surgery for multi-task model merging. In *International Conference on Machine Learning (ICML)*.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

A Effect of Calibration Data Size

The surgery module is trained on a small set of N domain-specific calibration samples. Table 7 examines how performance varies as N ranges from 10 to the full test set. Performance is relatively stable across calibration sizes, with as few as 20 samples achieving results competitive with larger budgets. This suggests that the surgery module learns a generalizable bias correction rather than memorizing calibration-specific patterns. Using the full test set for calibration yields comparable results, confirming that the module does not overfit even with abundant data.

Calibration N	HE	GSM	AE
10	24.4	42.5	54.0
20	24.4	44.8	58.4
30	26.2	43.8	56.8
50	25.0	43.4	58.6
100	26.2	42.8	55.4
Full Test	23.17	43.52	59.45

Table 7: Effect of calibration data size.

B Full Loss Function Comparison

In the main paper (Table 5), we compare loss functions using Standard surgery only. Here, Table 8 presents the full factorial comparison of loss functions crossed with surgery module type (Standard vs. position correction). This allows us to disentangle the contribution of the loss function from the contribution of the position correction module. Two patterns emerge. First, switching from L_1 to entropy-weighted L_1 consistently improves average performance regardless of the surgery module,

confirming that the loss function is the primary driver. Second, adding position correction on top of each loss function provides a further boost in average score, with the largest gain observed for EWL1 (+0.9 average), indicating that the two innovations are complementary.

C Position Correction Module Variants

Beyond the position correction module used in DARTS, we explored several alternative architectures for incorporating position information into the surgery module. Table 9 compares these variants, all trained with EWL1 loss at rank 16. **Standard** applies only the content bottleneck with no position awareness. **Add** adds a learned position embedding to the hidden state before the down-projection, allowing position to modulate the ReLU activation pattern. **Parallel** uses two independent bottleneck paths (one for content, one for position) whose outputs are summed; however, we observe that the position path causes severe degradation at inference, likely due to overfitting. **Pos. Corr. (Ours)** adds a free per-position bias vector that is independent across positions and zero-initialized, providing structural regularization that prevents overfitting while still capturing systematic position-dependent offsets. Among all variants, our position correction module achieves the highest average score while maintaining stable HumanEval performance, unlike the Parallel variant which collapses on code generation.

D Effect of Training Steps

Table 10 varies the number of training iterations from 50 to 1000. Performance is stable across this range, with even 50 iterations producing competitive results. Longer training (1000 steps) yields a modest GSM8K improvement without degrading other benchmarks.

E Effect of Surgery Rank

Table 11 varies the bottleneck rank r of the surgery module. Rank 16 (the default) achieves the best overall balance. Increasing rank to 32 or 64 does not improve performance and slightly hurts AlpacaEval, while rank 128 degrades HumanEval, suggesting that higher-rank modules overfit on the small calibration set.

F Seed Sensitivity

Table 12 reports DARTS performance across 6 random seeds that control the calibration data shuffle order. The low standard deviation (HE: ± 0.2, GSM: ± 0.8, AE: ± 1.0) confirms that DARTS is robust to the choice of calibration samples.

Loss Function	Surgery Module	HumanEval	GSM8K	AlpacaEval	Avg.
L_1	Standard	23.2	42.5	47.0	37.6
L_1	+ Pos. corr.	21.3	43.1	48.8	37.7
Entropy-Weighted L_1	Standard	25.6	41.7	57.0	41.4
Entropy-Weighted L_1	+ Pos. corr.	25.0	43.4	58.6	42.3

Table 8: Full loss function $\times$ surgery module grid.

Variant	HE	GSM	AE	Extra Params
Standard	25.6	41.7	57.0	0
Add	21.9	44.2	57.9	2.1M
Parallel	12.2	45.0	55.5	2.2M
Pos. Corr. (Ours)	25.0	43.4	58.6	2.1M

Table 9: Position correction module variant comparison.

Steps	HE	GSM	AE
50	25.6	42.3	57.9
100	25.0	42.9	56.5
200	25.0	43.4	57.9
400	24.4	43.0	57.0
500 (default)	25.0	43.4	58.6
1000	25.6	44.3	58.4

Table 10: Effect of training iterations.

Rank r	HE	GSM	AE	Params/domain
16	25.0	43.4	58.6	131K + 2.1M
32	25.0	44.3	56.4	262K + 2.1M
64	25.0	42.8	56.0	524K + 2.1M
128	23.8	42.6	58.1	1.04M + 2.1M

Table 11: Effect of surgery rank on DARTS.

Seed	HE	GSM	AE
default	25.0	43.4	58.6
7	24.4	45.0	57.3
29	25.0	43.6	58.3
71	25.0	45.0	59.9
164	25.0	44.4	57.1
190	25.0	45.0	59.0
Mean $\pm$ Std	24.9 $\pm$ 0.2	44.4 $\pm$ 0.8	58.4 $\pm$ 1.0

Table 12: Reproducibility across random seeds. Seed controls the calibration data shuffle order.