META-WEIGHTED SELF-TRAINING FOR LANGUAGE MODELS UNDER EXTREME DATA SCARCITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting language models from only a small set of prompt–completion pairs is challenging: direct fine-tuning overfits, while self-training on model-generated data can reinforce errors. We propose **MWeiST**, a meta-weighted self-training framework for extreme data scarcity that assigns each synthetic example a weight reflecting its *long-term training utility*: examples are upweighted if imitating them leads, after several adaptation steps, to the largest likelihood improvement on the real demonstrations. The weights therefore assign credit over future training dynamics rather than ranking examples by current sample quality. Starting from a frozen pretrained backbone with LoRA adapters, MWeiST generates synthetic completions and optimises these weights via a meta-objective on the held-out real demonstrations. To make this tractable at scale, we introduce **GradAlign**, a first-order approximation that replaces backpropagation through the inner optimisation with gradient alignment between per-example updates and the meta-loss; a double-gradient implementation recovers all alignments in two backward passes per step, independent of batch size. Working from as few as 80 real demonstrations, we adapt five backbones spanning 270M to 4B parameters on three summarisation benchmarks. MWeiST attains the lowest held-out perplexity in every model–dataset cell and improves completion cross-entropy over all five baselines in the large majority of paired comparisons. GradAlign tracks the exact BPTT meta-gradient closely while admitting substantially larger inner batches at the same memory budget. Learning which synthetic examples to train on is therefore practical at scales where exact meta-gradients are not.

1 INTRODUCTION

Adapting language models to new tasks or domains typically relies on fine-tuning with large collections of labeled or curated data (Howard & Ruder, 2018; Gururangan et al., 2020). In many practical settings, however, only a handful of real prompt-completion pairs is available, and standard fine-tuning becomes unstable and prone to overfitting (Dodge et al., 2020; Mosbach et al., 2020). Self-training on model-generated completions is a natural alternative, but synthetic data is highly heterogeneous: some generations provide useful learning signal, others reinforce errors, spurious patterns, or low-diversity behavior (Wang et al., 2022).

We study self-training in the regime of *extreme data scarcity*. Rather than treating all synthetic examples equally, we cast synthetic-data training as a bilevel optimization in which each generated completion carries a learned importance weight (Ren et al., 2018; Shu et al., 2019; Franceschi et al., 2018). At each iteration the current model generates synthetic completions for the real prompts, and a teacher-forced meta-objective on the real pool identifies which synthetic examples induce beneficial parameter updates, allowing the model to amplify useful generations and suppress harmful ones.

Our approach, **MWeiST**, assigns learnable weights to synthetic examples and optimises them through short look-ahead training trajectories. The inner objective is standard causal language modeling on the weighted synthetic data; the outer objective measures whether those updates improve prediction on real completions. We use cross-entropy as the default meta-loss and instantiate the method in a parameter-efficient setting by freezing the pretrained backbone and training only LoRA adapters. The result is a self-training framework that requires no separate teacher model and remains practical at billion-parameter scale.

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

The central technical challenge is that exact meta-gradient computation through the inner training dynamics is prohibitively expensive at this scale (Maclaurin et al., 2015). We address it with **GradAlign**, a first-order approximation that replaces backpropagation through the inner optimisation with the alignment between per-example training gradients and the meta-loss gradient. A minibatch-wise double-gradient implementation recovers all per-example alignments without one backward pass per example. Empirically, this approximation closely matches the direction of the exact meta-gradient while substantially reducing memory and runtime.

We make the following contributions: (i) a meta-weighted self-training framework for language model adaptation under extreme data scarcity that learns example-level weights over synthetic completions; (ii) a parameter-efficient instantiation that freezes the base model and trains only LoRA adapters, making the method practical at billion-parameter scale; (iii) GradAlign, a first-order approximation to exact truncated BPTT for synthetic-example reweighting; and (iv) a minibatch-wise double-gradient implementation that yields substantial savings in compute and memory while preserving the direction of the exact meta-gradient.

2 Problem Statement, Background and Motivations

2.1 Objective and low-data adaptation setting

We consider the problem of adapting a pretrained language model to an unknown target distribution. Let $(X, Y) \sim p^\star(X, Y)$ denote context–completion pairs. The objective is to maximise

$$\mathbb{E}_{(X,Y)\sim p^\star}\big[\log p_\theta(Y \mid X)\big], \tag{1}$$

which is intractable as $p^\star$ is unknown. Instead, we observe a small dataset $\mathcal{D}_{\text{real}} = \{(X_i, Y_i)\}_{i=1}^{n_{\text{tr}}}$ with $n_{\text{tr}} \ll 1{,}000$, and a held-out evaluation set $\mathcal{D}_{\text{eval}}$. We assume a pretrained backbone with frozen parameters θ_0, adapted via a small set of trainable parameters ϕ (e.g. LoRA adapters). This restriction reflects practical regimes where full fine-tuning is either unstable or computationally prohibitive.

The standard approach is supervised fine-tuning (SFT), which optimises the empirical log-likelihood under teacher forcing:

$$\mathcal{L}_{\text{sft}}(\phi; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \ell(\phi; X, Y), \quad \ell(\phi; X, Y) = -\frac{1}{|Y|} \sum_t \log p_{\theta_0,\phi}(Y^{(t)} \mid X, Y^{(<t)}). \tag{2}$$

However, even with abundant data, SFT has inherent limitations: it relies on teacher forcing, leading to exposure bias, and optimises token-level likelihood rather than sequence-level generation quality. In low-data regimes, these limitations are compounded by overfitting and instability, as fine-tuning on small datasets tends to memorise examples rather than generalise (Howard & Ruder, 2018; Gururangan et al., 2020; Dodge et al., 2020; Mosbach et al., 2020).

In-context learning provides an alternative adaptation mechanism, but remains fundamentally limited by the pretrained model's inductive biases and struggles under distribution shift. As a result, parameter updates are typically required to achieve robust adaptation.

2.2 Generation-based training as weighted policy gradients

To overcome data scarcity, a common strategy is to augment $\mathcal{D}_{\text{real}}$ with model-generated samples. Let $X \sim \mathcal{D}_{\text{real}}$ and $\hat{Y} \sim p_{\theta_0,\phi}(\cdot \mid X)$, and define a synthetic dataset $\mathcal{D}_{\text{syn}} = \{(X_j, \hat{Y}_j)\}_{j=1}^{N}$.

A natural approach is to apply standard SFT on $\mathcal{D}_{\text{syn}}$, i.e. to minimise $\mathcal{L}_{\text{sft}}(\phi; \mathcal{D}_{\text{syn}})$. However, this corresponds to treating all synthetic examples equally, which can reinforce model errors: low-quality generations are learned as if they were correct, leading to degeneration or mode collapse (Wang et al., 2022). To mitigate this issue, generation-based self training methods usually consider a weighted variant of the objective, where each synthetic example i is assigned a weight w_i reflecting its usefulness for training:

$$\mathcal{L}_{\text{train}}(\phi, w, \mathcal{D}_{\text{syn}}) = \frac{1}{N} \sum_{i\in\mathcal{D}_{\text{syn}}} w_i\, \ell_i(\phi), \qquad \ell_i(\phi) = \ell(\phi, X_i, \hat{Y}_i). \tag{3}$$

Most generation-based methods are iterative, alternating a phase of synthetic data generation and a phase of improvement on the produced dataset. In expectation, this induces an optimization problem $\min_\phi \mathbb{E}_{\mathcal{D}_{\text{syn}} \sim p_{\theta_0,\phi}} \mathcal{L}_{\text{train}}(\phi, w, \mathcal{D}_{\text{syn}})$ equivalent to the RL-like objective $\max_\phi \mathbb{E}_{X \sim \mathcal{D}_{\text{real}},\, \hat{Y} \sim p_{\theta_0,\phi}} \big[w(X, \hat{Y})\big]$, where $w(X, \hat{Y})$ acts as a reward, with corresponding policy gradient:

$$\nabla_\phi \mathcal{L} = \mathbb{E}_{X \sim \mathcal{D}_{\text{real}},\, \hat{Y} \sim p_{\theta_0,\phi}} \big[w(X, \hat{Y})\, \nabla_\phi \log p_{\theta_0,\phi}(\hat{Y} \mid X)\big]. \tag{4}$$

This policy-gradient perspective unifies generation-based methods through different choices of the weighting function $w(X, \hat{Y})$, which acts either as a continuous score or a binary filter, defined at the level of individual samples (e.g., (STaR; Zelikman et al., 2022) for reasoning-trace fine-tuning where w selects generations consistent with the correct answer or a valid rationale) or relatively across multiple generations for the same context, as in GRPO-style schemes (including Best-of-N methods (Touvron et al., 2023; Dong et al., 2023), where w retain generations assigned to highest reward scores). While approaches such as (ReST and ReST^{EM}; Gulcehre et al., 2023; Singh et al., 2023) rely on explicit reward functions, we consider a setting where the task is specified only through the demonstration dataset $\mathcal{D}_{\text{real}}$, without access to external evaluators such as RLHF-style preference models or LLM-as-a-judge systems (Yuan et al., 2024).

In this regime, w must be inferred from data. Inverse Reinforcement Learning seeks a reward consistent with expert demonstrations (Wulfmeier et al., 2024), while contrastive or GAN-like methods (Yu et al., 2017; Lamprier et al., 2022) instantiate $w \propto \exp(D_\psi(X, \hat{Y}))$ via a discriminator separating real and generated samples. However, with limited data, these signals are fundamentally underconstrained and prone to capturing spurious correlations rather than task-relevant structure. When used in iterative self-training, this leads to a self-reinforcing loop where the model progressively shapes its own training distribution, inducing distributional drift, loss of diversity, and instability. In adversarial settings, this is exacerbated by the non-stationarity of the learned reward, which co-evolves with the generator and yields high-variance optimization dynamics. As a result, w becomes brittle, poorly grounded beyond $\mathcal{D}_{\text{real}}$, and susceptible to reward hacking as optimization exploits imperfections of the learned signal.

In all these methods, $w(X, \hat{Y})$ is estimated from the current model and reflects only the instantaneous quality of generated samples. This introduces two key limitations: (i) myopic credit assignment, since samples are evaluated independently of their long-term impact on future learning dynamics, and (ii) strong on-policy dependence, which restricts data reuse and increases variance. Existing approaches mitigate these issues either via validation-driven synthetic data selection (e.g., ReST and ReST^{EM}) or trust-region constraints such as PPO-style clipping and KL-regularized objectives (including DPO-like formulations), which restrict deviation from the data distribution. However, these methods remain fundamentally tied to sample-level evaluation and do not explicitly account for the effect of data on future learning dynamics, motivating a shift toward a meta-level view of synthetic data as shaping the trajectory of optimization itself.

2.3 Learning weights as future utility: a meta-gradient perspective

We revisit the problem from a meta-learning and bilevel optimization perspective (Maclaurin et al., 2015; Ren et al., 2018; Shu et al., 2019; Franceschi et al., 2018), combining meta data reweighting with self-training on generated samples. Instead of scoring synthetic examples by their immediate quality, we evaluate their effect on downstream generalization.

Given a synthetic dataset $\mathcal{D}_{\text{syn}}$, we consider a weighted training objective $\mathcal{L}_{\text{train}}(\phi, w, \mathcal{D}_{\text{syn}})$ (see Eq. (3)). Starting from ϕ_0, let $\phi_T(w)$ denote the parameters obtained after T steps of SGD, defining an implicit mapping from weights to final model parameters.

We define the bilevel problem:

$$w^\star = \arg\min_{w \in \Delta^N} \mathcal{L}_{\text{meta}}(\phi_T(w)), \tag{5}$$

where $\Delta^N = \{w \in [w_{\min}, w_{\max}]^N : \sum_{i=1}^N w_i = 1\}$ enforces a normalization constraint on the weights and the meta-objective is defined as the supervised loss on real data (i.e., $\mathcal{L}_{\text{meta}}(\phi) = \mathcal{L}_{\text{SFT}}(\phi; \mathcal{D}_{\text{real}})$). This formulation links synthetic weight selection to downstream performance on real data, with w influencing the objective only through the induced training trajectory.

Our work connects bilevel optimization for data reweighting (Ren et al., 2018; Shu et al., 2019) with self-training on model-generated data, and more broadly relates to meta-learning and hypergradient methods that optimize generalization through training dynamics (Finn et al., 2017; Nichol et al., 2018; Lorraine et al., 2020; Rajeswaran et al., 2019), as well as curriculum learning and dataset distillation approaches (Bengio et al., 2009; Wang et al., 2018).

Several works share our framing of synthetic training data as a continuous metaparameter optimized by gradient through the inner training loop. Such et al. (2019) introduce Generative Teaching Networks, which differentiate through a learner's training to update a synthetic-data generator and accelerate neural architecture search. Concurrently, Kaya & Rui (2026) propose MASS, a test-time meta-adaptation framework that backpropagates through inner updates to generate per-instance curricula for mathematical reasoning, and Thrush et al. (2026) introduce Dataset Policy Gradient, which optimizes synthetic-data generators via exact higher-order gradients to embed targeted properties into model weights.

All of the methods above, however, either optimize model parameters across tasks or operate on fixed datasets within each optimization episode. We instead consider a closed-loop setting where the model generates its own training data, which is then reweighted to update the same model that produced it. This introduces a coupling between data generation and optimization dynamics that violates standard assumptions of meta-gradient methods: unrolled differentiation and checkpoint-based REPLAY (Engstrom et al., 2025) become intractable under long horizons, implicit differentiation requires a stationary optimization problem, and episodic approximations such as MAML-style methods assume fixed datasets.

We therefore introduce GRADALIGN, which approximates the effect of w given current data on final performance, without full backpropagation through the trajectory, by aligning local update effects with their accumulated impact on generalization.

3 METHOD

3.1 ALGORITHM OVERVIEW

MWeiST runs three nested loops, summarised in Algorithm 1. The **outer loop** (M generation cycles) is responsible for on-policy updates, akin to all generation-based self training presented in section 2.2, but using weights obtained with meta-learning (c.f., middle and inner loops below), on a new synthetic dataset $\mathcal{D}_{\text{syn}}^{m}$ sampled from the current model $p_{\theta_0,\phi^{m-1}}$ at each iteration m. The **middle loop** (K meta-iterations) updates the importance weights w by Adam on the meta-gradient $\partial\mathcal{L}_{\text{meta}}/\partial w$; each update requires solving a short, fully differentiable inner trajectory. The **inner loop** (T SGD steps) is that look-ahead: it trains $\phi_0 \to \phi_T$ under the weighted loss $\mathcal{L}_{\text{train}}(\cdot, w)$ (3) and is what couples w to $\mathcal{L}_{\text{meta}}$ through the bilevel problem (5).

At the end of the outer loop, the update of the model is performed using the learned meta-weights via a retraining step (line 20 of Algorithm 1). Retrain denotes one epoch of LoRA training initialized at $\phi^{(m-1)}$ on $\mathcal{D}_{\text{syn}}^{(m)}$, using the weighted loss $\mathcal{L}_{\text{train}}(\cdot, w, \mathcal{D}_{\text{syn}}^{(m)})$ induced by tuned fixed weights w. Optimization is performed using a standard optimizer at learning rate η_r. Decoupling retraining from meta-gradient estimation serves two purposes: the optimised weights are applied consistently from the first step rather than as the last block of a time-varying schedule whose first block sees the uniform initialisation $w^{(0)} = \mathbf{1}$; and it permits the use of a stronger optimizer (SGD with Nesterov momentum by default, optionally Adam) than the plain SGD that meta-differentiability imposes on the inner-loop look-ahead.

Progressive look-ahead. The middle loop forms an extended training trajectory across its K iterations: iteration k starts from $\phi_T^{(k-1)}$, the endpoint of iteration $k-1$, and produces $\phi_T^{(k)}$. By the end of the middle loop the model has accumulated KT SGD steps from $\phi^{(m-1)}$. Each meta-gradient differentiates only through the most recent T-step window, treating $\phi_T^{(k-1)}$ as a fixed initial condition. This is exactly truncated BPTT (Williams & Peng, 1990); the truncation is justified by (i) the small window size T; (ii) the slow change in w under Adam at the chosen meta learning rate; and (iii) the empirical observation that later meta-gradients are computed from a more informative point along the training trajectory.

3.2 DIFFERENTIABLE INNER LOOP

The inner loop runs T steps of vanilla SGD on the weighted loss:

$$\phi_{t+1} \;=\; F_t(\phi_t, w) \;\coloneqq\; \phi_t \;-\; \frac{\alpha}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} w_i \, \nabla_\phi \ell_i(\phi_t), \qquad t = 0, \ldots, T-1, \tag{6}$$

where α is the inner learning rate and $\mathcal{B}_t$ is the mini-batch sampled at step t. We deliberately use plain SGD (no momentum, no adaptive scaling) in all reported experiments, for two reasons. First, (6) is linear in w, which is exactly what enables the double-grad trick in Sec. B. Second, adaptive optimizers maintain per-parameter running statistics whose dependence on w is non-trivial to differentiate through, both analytically and in memory (Finn et al., 2017; Grefenstette et al., 2019). Adaptive optimisation is recovered, when desirable, in the retrain phase, where no meta-gradient is needed.

3.3 META-GRADIENT COMPUTATION

3.3.1 EXACT BACKPROPAGATION THROUGH TRAINING

By the chain rule, the meta-gradient decomposes as

$$\frac{\partial \mathcal{L}_{\text{meta}}(\phi_T)}{\partial w} \;=\; \underbrace{\nabla_\phi \mathcal{L}_{\text{meta}}(\phi_T)}_{g_{\text{meta}} \in \mathbb{R}^P} \cdot \underbrace{\frac{\partial \phi_T}{\partial w}}_{\in \mathbb{R}^{P \times N}}, \qquad P = |\phi|. \tag{7}$$

The Jacobian $\partial \phi_T / \partial w$ is never materialised; reverse-mode automatic differentiation instead propagates the vector g_{meta} backward through the trajectory by a sequence of backward passes. With $s_T \coloneqq g_{\text{meta}}$ and $s_t \coloneqq s_{t+1}\, \partial F_t / \partial \phi|_{\phi_t, w}$,

$$\frac{\partial \mathcal{L}_{\text{meta}}}{\partial w} \;=\; \sum_{t=0}^{T-1} s_{t+1} \, \frac{\partial F_t}{\partial w}\bigg|_{\phi_t, w}. \tag{8}$$

Because F_t is itself defined in terms of a gradient, each backward step through F_t requires a second-order backward pass through $\nabla_\phi \ell_i$. The per-iteration cost is therefore T second-order backward passes, with a memory cost of $O(T \cdot P)$ for the saved trajectory plus the live adjoint state. At our scale the second-order memory dominates and exact BPTT becomes infeasible on a single accelerator at billion-parameter sizes; concrete memory and runtime numbers are reported in App. A.

3.3.2 FIRST-ORDER APPROXIMATION: **GRADALIGN**

The exact gradient (8) expands, after one step of the recursion, into a sum of inner products between the propagated adjoint s_{t+1} and the per-example training gradients $\nabla_\phi \ell_i(\phi_t)$. Following the spirit of first-order MAML (Finn et al., 2017; Nichol et al., 2018), we drop the backward chain and replace s_{t+1} with the meta-loss gradient g_{meta}:

$$\frac{\partial \mathcal{L}_{\text{meta}}}{\partial w_i} \;\approx\; -\frac{\alpha}{|\mathcal{B}_t|} \sum_{t \,:\, i \in \mathcal{B}_t} \big\langle g_{\text{meta}},\; \nabla_\phi \ell_i(\phi_t) \big\rangle. \tag{9}$$

The interpretation is direct: example i is upweighted in proportion to the alignment between its training gradient and the direction that most decreases the meta-loss. Examples whose gradients point "towards" the meta-objective are amplified; those that pull away are suppressed. The approximation is biased, since it discards the curvature terms BPTT captures through $\partial F_t / \partial \phi$. Appendix B reports near-perfect empirical agreement with truncated exact BPTT at 270M and 1B and at unroll lengths of 75 and 150 steps; we do not claim a theoretical error bound or extrapolate beyond that regime.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate MWeiST in low-resource settings across a range of tasks, model scales, and meta-optimisation variants. To simulate extreme data scarcity, adaptation on each dataset uses 80 real

Algorithm 1 MWeiST: Meta-Weighted Self-Training with progressive look-ahead and **GradAlign**.

Require: Frozen base parameters θ_0; initial LoRA parameters $\phi^{(0)}$; real train pool $\mathcal{D}_{\text{real}}$; hyperparameters M, K, T (step budgets); α, η, η_r (inner-SGD / meta-Adam / retrain LRs).

for $m = 1, \dots, M$ **do** ▷ outer: data refresh
  Generate $\mathcal{D}_{\text{syn}}^{(m)} = \{x_j\}_{j=1}^{N}$ from $p_{\theta_0, \phi^{(m-1)}}(\cdot \mid X_i)$
  $w \leftarrow \mathbf{1}_N$; init Adam state for w; $\tilde{\phi} \leftarrow \phi^{(m-1)}$
  for $k = 1, \dots, K$ **do** ▷ middle: meta-learning
    $w \leftarrow \text{clip}(w, w_{\min}, w_{\max})$; $w \leftarrow N\, w / \sum_j w_j$
    for $t = 0, \dots, T-1$ **do** ▷ inner: differentiable SGD
      Sample mini-batch $\mathcal{B}_t \subset \{1, \dots, N\}$
      Save $\phi_t \leftarrow \tilde{\phi}$
      $\tilde{\phi} \leftarrow \tilde{\phi} - (\alpha / |\mathcal{B}_t|) \sum_{i \in \mathcal{B}_t} w_i \nabla_\phi \ell_i(\tilde{\phi})$
    end for
    $g_{\text{meta}} \leftarrow \nabla_\phi \mathcal{L}_{\text{meta}}(\tilde{\phi})$
    $g_w \leftarrow \mathbf{0}_N$
    for $t = 0, \dots, T-1$ **do** ▷ double-grad alignment
      $h_t(w) \leftarrow \langle g_{\text{meta}},\ \nabla_\phi \mathcal{L}_{\text{train}}(\phi_t, w, \mathcal{D}_{\text{syn}}^{(m)}) \rangle$
      $g_w[\mathcal{B}_t] \mathrel{-}= \alpha \cdot \nabla_w h_t(w)$
    end for
    $w \leftarrow w - \eta\, \text{Adam}(g_w)$
  end for
  $w \leftarrow \text{clip}(w, w_{\min}, w_{\max})$; $w \leftarrow N\, w / \sum_j w_j$
  $\phi^{(m)} \leftarrow$ **Retrain**$\big(\phi^{(m-1)},\ \mathcal{D}_{\text{syn}}^{(m)},\ w;\ \eta_r\big)$ ▷ one epoch, full optimizer (e.g. SGD+momentum)
end for
return $\phi^{(M)}$

demonstrations. The three primary benchmarks cover both dialogue and news summarisation: SAMSum (chat-style messenger dialogue), DialogSum (transcribed spoken dialogue), and XSum (extreme news summarisation). These 80 demonstrations are the only real data the optimiser ever sees: they supply the prompts from which synthetic completions are generated and the targets of the teacher-forced meta-loss. Headline results are reported on 500 held-out examples per dataset, drawn from each dataset's official test split and unseen throughout the training pipeline (Table 1). A further 20 held-out examples, likewise never optimised against, serve as an online monitoring proxy: they track learning dynamics from cycle to cycle at negligible cost, and carry the cycle-by-cycle trajectories and the ablations.

We test five base models spanning three architecture families: Gemma-3 at 270M, 1B, and 4B parameters (dense Gemma); Llama-3.2-3B (dense Llama); and GraniteMoE-3B (sparse mixture-of-experts).[1] All five models are used in their pre-trained base form, prior to any instruction tuning or preference optimisation. Unless stated otherwise, the pretrained backbone is frozen and only LoRA adapters are trained. Synthetic data is regenerated at each cycle from the current model, and each cycle then applies the procedure of Section 3. We report perplexity (PPL), teacher-forced cross-entropy on the target completion only (CCE), and character n-gram F-score (ChrF) (Popović, 2015). PPL scores the complete prompt–completion sequence, whereas CCE is averaged over the target-completion tokens alone, with the prompt tokens excluded from the loss, and is therefore our primary likelihood measure.

Synthetic data generation. Each cycle samples $n_{\text{per}} = 15$ completions per real prompt ($N = 80 \cdot 15 = 1200$ synthetic examples per cycle) using nucleus sampling with a repetition penalty of 1.3. The maximum number of generated tokens per sample is set per dataset to match typical target lengths: 80 tokens for XSum, and 120 tokens for SAMSum and DialogSum. The same generation

[1] For models above the 1B-parameter threshold the frozen base weights are cast to `bfloat16` at load time, since second-order derivatives through the full float32 model exceed our single-GPU memory budget. LoRA adapters and the loss computation remain in float32. See App. A for details.

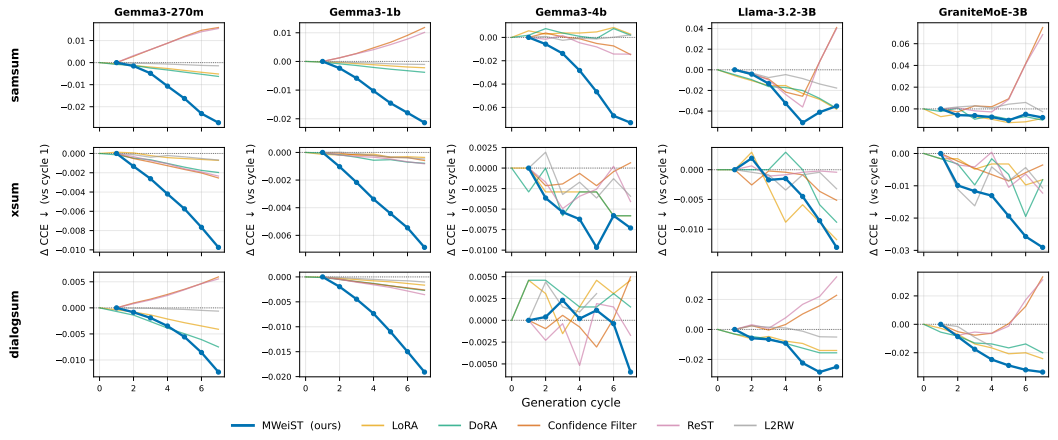


Figure 1: **Held-out CCE meta-loss across generation cycles**. Each panel tracks CCE on the 20-example monitoring set over M self-improvement cycles; Table 1 reports the final checkpoints. Cycle 0 is the base model.

budget applies to all baselines that consume synthetic data, ensuring fair comparison. The full list of hyperparameters is given in App. A (Table 2).

4.2 Baselines

We compare MWeiST against five baselines in two regimes. *LoRA* and *DoRA* are non-adaptive parameter-efficient fine-tuning recipes that ignore synthetic data entirely and provide a reference point for what scarce real data alone buys. *Confidence Filtering*, *ReST*, and *L2RW* are self-training procedures that share MWeiST's pipeline (same base model, generation budget, retrain optimizer and schedule) and differ only in the rule used to reweight or filter the synthetic pool before the retrain step.

LoRA and DoRA. We fine-tune LoRA adapters (Hu et al., 2021) directly on the 80 ground-truth pairs (X_i, Y_i^{real}), with no synthetic data, in a single round of supervised cross-entropy fine-tuning to convergence. DoRA (Liu et al., 2024) factorizes each adapted weight matrix into a unit-norm direction produced by the LoRA delta and an explicit magnitude vector m learned alongside it, $W' = m \odot (W_0 + \frac{\alpha}{r}BA)/\|W_0 + \frac{\alpha}{r}BA\|_{\text{col}}$, which empirically narrows the gap to full fine-tuning at low ranks; we train DoRA on the same 80 real pairs. Together, LoRA and DoRA distinguish gains from synthetic data from gains due to a stronger PEFT parameterization.

Confidence Filtering. At each cycle, the current LoRA scores every synthetic completion by its length-normalized log-probability conditioned on the prompt and only the top-K candidates per template are retained for retraining with uniform weights. Per-template selection (rather than a global threshold) preserves the diversity of all 80 prompts in the scarce regime. Confidence filtering is closely related to Noisy Student self-training (Xie et al., 2019) and FixMatch-style confidence thresholding (Sohn et al., 2020); it is binary and uses no information from B^{real}, isolating whether MWeiST's continuous, validation-aware weights help over a model-only filter.

ReST. Reinforced Self-Training (Gulcehre et al., 2023) alternates a *Grow* step that samples candidates and an *Improve* step that fine-tunes on the highest-reward survivors. Because the original work's learned reward model is not viable with 80 examples, we adopt the rejection-sampling variant popularized by ReST^{EM} (Singh et al., 2023): each candidate is scored against the gold completion with ROUGE-L F1, the top-M per template are retained, and SFT is run on the pool with uniform weights. ReST has per-sample access to Y^{real} via the reward, whereas MWeiST uses Y^{real} only through the aggregated CCE meta-loss; matching ReST is therefore already a meaningful result, and beating it under weaker supervision is a strong one.

Table 1: **Final checkpoints evaluated on 500 held-out examples per dataset.** Training is unchanged from the 80-example protocol. The evaluation examples come from each dataset's official test split and are unseen throughout the training pipeline. Entries are pooled point estimates; **bold** marks the best value per model, dataset, and metric. MWeiST obtains the lowest PPL in all 15 cells and the lowest CCE in 13.

		SAMSum			XSum			DialogSum		
Model	**Method**	PPL↓	CCE↓	ChrF↑	PPL↓	CCE↓	ChrF↑	PPL↓	CCE↓	ChrF↑
Gemma3-270m	MWeiST (ours)	**19.694**	**2.212**	**24.146**	**16.605**	**2.318**	14.049	**10.900**	**2.288**	24.301
	LoRA	20.953	2.240	23.630	16.746	2.324	14.053	11.365	2.300	24.221
	DoRA	20.926	2.238	23.714	16.743	2.324	**14.060**	11.348	2.295	24.173
	Confidence Filter	20.514	2.264	23.908	16.698	2.324	14.049	11.215	2.311	24.312
	ReST	20.510	2.263	23.933	16.697	2.324	14.044	11.213	2.311	**24.312**
	L2RW	20.988	2.244	23.710	16.747	2.325	14.042	11.378	2.304	24.204
Gemma3-1b	MWeiST (ours)	**14.727**	**1.791**	**24.672**	**11.234**	**1.806**	13.897	**8.162**	**1.806**	**24.034**
	LoRA	15.509	1.825	23.744	11.300	1.808	**13.935**	8.374	1.828	20.604
	DoRA	15.497	1.824	23.862	11.296	1.808	13.925	8.369	1.826	21.899
	Confidence Filter	15.343	1.837	24.106	11.276	1.808	13.912	8.319	1.827	23.547
	ReST	15.318	1.835	24.340	11.276	1.808	13.912	8.315	1.826	23.885
	L2RW	15.507	1.826	23.914	11.299	1.808	13.898	8.372	1.829	21.260
Gemma3-4b	MWeiST (ours)	**11.856**	**1.520**	24.837	**8.685**	**1.529**	14.075	**6.624**	1.537	23.633
	LoRA	12.601	1.578	24.920	8.782	1.536	14.071	6.783	1.532	23.968
	DoRA	12.597	1.578	24.908	8.777	1.536	14.071	6.782	**1.531**	**23.989**
	Confidence Filter	12.450	1.564	25.027	8.761	1.534	14.081	6.759	1.537	23.744
	ReST	12.453	1.566	**25.073**	8.760	1.534	**14.084**	6.736	1.531	23.826
	L2RW	12.596	1.577	25.042	8.780	1.537	14.082	6.783	1.532	23.864
Llama-3.2-3B	MWeiST (ours)	**12.125**	**1.536**	18.615	**9.490**	**1.604**	**15.421**	**6.848**	1.549	17.686
	LoRA	12.331	1.555	19.672	9.689	1.612	15.202	7.213	1.547	19.218
	DoRA	12.326	1.552	**19.676**	9.690	1.613	15.192	7.204	**1.546**	**19.285**
	Confidence Filter	12.521	1.598	17.830	9.647	1.619	15.375	7.140	1.592	17.289
	ReST	12.506	1.598	18.036	9.651	1.621	15.379	7.175	1.605	16.844
	L2RW	12.314	1.570	19.040	9.686	1.624	15.347	7.212	1.564	18.380
GraniteMoE-3B	MWeiST (ours)	**8.829**	**1.333**	22.316	**6.734**	**1.302**	14.884	**5.366**	**1.304**	22.345
	LoRA	8.912	1.337	**22.479**	6.758	1.310	14.887	5.516	1.323	**22.432**
	DoRA	8.892	1.339	22.435	6.757	1.312	14.885	5.547	1.324	22.409
	Confidence Filter	8.986	1.385	22.140	6.748	1.310	14.880	5.487	1.366	22.180
	ReST	9.010	1.390	22.154	6.746	1.312	**14.892**	5.472	1.361	22.189
	L2RW	8.891	1.340	22.373	6.761	1.316	14.876	5.549	1.333	22.393

L2RW. Learning to Reweight (Ren et al., 2018) computes per-example weights via a *single* SGD lookahead. For each minibatch, a hypothetical step parameterized by per-example coefficients ε_i at ε=0 yields a closed-form expression for the gradient of the meta-loss with respect to ε_i, $g_i = -\alpha \langle \nabla_\phi \mathcal{L}_{\text{meta}}(\phi),\, \nabla_\phi \ell_i(\phi) \rangle$, i.e. the alignment between the per-example training gradient and the meta-loss gradient. Weights are taken as $\tilde{w}_i = \max(0, -g_i)$, normalized to sum to one, and applied to the immediate next SGD update; minibatches with no positively-aligned examples are skipped. The validation signal and inner learning rate are matched to MWeiST. L2RW is the cleanest algorithmic competitor: it shares MWeiST's validation-aware, continuous weighting philosophy but uses a one-step approximation with no outer refinement, so beating it is direct evidence that MWeiST's multi-step, multi-iteration meta-gradient earns its additional compute.

4.3 Results in Low-Resource Regimes

Table 1 reports the final-checkpoint evaluation; the trajectories in Figures 1 and 5 are read on the monitoring set. MWeiST attains the lowest PPL in all 15 model–dataset cells, and it wins all 75 paired PPL comparisons against the five baselines with confidence intervals strictly above zero. CCE is nearly as decisive: MWeiST holds the lowest point estimate in 13 of 15 cells, with 63 statistically significant improvements, 10 inconclusive comparisons, and two cells in which a baseline is ahead by at most 0.006 nats, both on Gemma-3 4B/DialogSum. In the one remaining cell, Llama-3.2

3B/DialogSum, MWeiST still improves significantly on three of the five baselines and is inconclusive against the other two.

The larger evaluation set also reverses an apparent XSum deficit that was visible on the monitoring set. Against LoRA and DoRA, all ten paired XSum CCE differences favour MWeiST; eight are significant, and only the two Gemma-3 1B intervals overlap zero. We therefore attribute MWeiST's weaker CCE ranking on the 20-example monitoring set to evaluation noise rather than to a systematic failure on news summarisation. The multi-split control in Appendix C locates the source of the XSum improvement in the self-training loop itself, with meta-weighting and uniform self-training close to each other on that dataset.

The evidence is strongest for the likelihood metrics. Reference-overlap metrics discriminate far less: for ChrF, and for the ROUGE-L and BLEU checks reported in Appendix C, most paired intervals overlap zero and some model-dataset cells favour a baseline. We therefore claim a reliable reduction in held-out loss, not a uniform improvement in reference-overlap quality. The monitored trajectories remain mechanistically informative: MWeiST's CCE usually descends as the synthetic pool is regenerated, whereas Confidence Filtering and ReST frequently drift upward and L2RW stays comparatively flat. Paired uncertainty estimates, split sensitivity, and additional task formats are reported in Appendix C.

5 Limitations

Three limitations bound what we can claim, and each suggests a direction for further work. First, GradAlign is a biased estimator: Eq. (9) discards the curvature terms that BPTT captures through $\partial F_t/\partial\phi$. We find near-perfect empirical agreement at 270M and 1B and at unroll lengths of 75 and 150 steps (App. B), but we have no theoretical bound on the bias and do not extrapolate beyond the scales and horizons we tested. Second, the scarcity regime we study is narrow. Every comparison in the main grid uses $n_{\mathrm{tr}} = 80$, and we have not measured the moderate-scarcity regime of 10^3 to 10^4 examples, in which uniform self-training and direct fine-tuning may already be competitive. Third, our evidence outside summarisation is thin: the grammar-correction and data-to-text results in Table 9 cover only the three Gemma scales. The 4B E2E comparison is also mixed: MWeiST has the lowest CCE, while ReST has lower perplexity and higher ChrF. The instruction-following and arithmetic evaluations use 100 held-out examples (Table 8). These results do not yet establish gains on open-ended generation, code, or multi-step reasoning.

A further limitation stems from the method's design. MWeiST depends on base-model competence: it reweights existing synthetic completions and cannot up-weight a useful completion that the base model never produces, so on tasks that the base model cannot attempt, the method degrades to standard self-training (or worse). And our adaptation is LoRA-only: we freeze the backbone and adapt only attention projections at rank 8. Whether the weighting structure learned by MWeiST transfers to full-model fine-tuning, higher LoRA ranks, or different target modules is unexplored.

6 Conclusion

MWeiST makes meta-weighted self-training tractable at billion-parameter scale. Across five backbones and three summarisation datasets, adapting from 80 real demonstrations each, it attains the lowest held-out perplexity in all 15 cells and significantly lower completion cross-entropy in 63 of 75 paired comparisons. The controls locate the source of that advantage. On both dialogue benchmarks the meta-learned weights add a clear margin over uniform self-training at a matched budget, and they beat reward-driven reweighting of the same synthetic pool; GradAlign supplies the weights at first-order cost while tracking the direction of the exact meta-gradient.

REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. *ICML*, 2009.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. URL https://arxiv.org/abs/2002.06305.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment, 2023. URL https://arxiv.org/abs/2304.06767.

Logan Engstrom, Andrew Ilyas, Benjamin Chen, Axel Feldmann, William Moses, and Aleksander Mądry. Optimizing ML training with metagradient descent. *arXiv preprint arXiv:2503.13751*, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017. URL https://arxiv.org/abs/1703.03400.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning, 2018. URL https://arxiv.org/abs/1806.04910.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning, 2019. URL https://arxiv.org/abs/1910.01727.

Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, March 2000. ISSN 1557-7295. doi: 10.1145/347837.347846. URL http://dx.doi.org/10.1145/347837.347846.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Zeyneb N. Kaya and Nick Rui. Test-time meta-adaptation with self-synthesis, 2026. URL https://arxiv.org/abs/2603.03524.

Sylvain Lamprier, Thomas Scialom, Antoine Chaffin, Vincent Claveau, Ewa Kijak, Jacopo Staiano, and Benjamin Piwowarski. Generative cooperative networks for natural language generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 11891–11905. PMLR, 2022. URL https://proceedings.mlr.press/v162/lamprier22a.html.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024. URL https://arxiv.org/abs/2402.09353.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*, 2020.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. Gradient-based hyperparameter optimization through reversible learning, 2015. URL https://arxiv.org/abs/1502.03492.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines, 2020. URL https://arxiv.org/abs/2006.04884.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. URL https://arxiv.org/abs/1803.02999.

Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pp. 392–395, 2015.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning, 2018. URL https://arxiv.org/abs/1803.09050.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting, 2019. URL https://arxiv.org/abs/1902.07379.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, 2023. URL https://arxiv.org/abs/2305.17493.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models, 2023. URL https://arxiv.org/abs/2312.06585.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020. URL https://arxiv.org/abs/2001.07685.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data, 2019. URL https://arxiv.org/abs/1912.07768.

Tristan Thrush, Sung Min Park, Herman Brunborg, Luke Bailey, Marcel Roed, Neil Band, Christopher Potts, and Tatsunori Hashimoto. Synthetic data for any differentiable target, 2026. URL https://arxiv.org/abs/2604.08423.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. In *arXiv preprint arXiv:1811.10959*, 2018.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2022. URL https://arxiv.org/abs/2212.10560.

Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, December 1990. ISSN 1530-888X. doi: 10.1162/neco.1990.2.4.490. URL http://dx.doi.org/10.1162/neco.1990.2.4.490.

Markus Wulfmeier, Michael Bloesch, Nino Vieillard, Arun Ahuja, Jörg Bornschein, Sandy Huang, Artem Sokolov, Matt Barnes, Guillaume Desjardins, Alex Bewley, et al. Imitating language via scalable inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 90714–90735, 2024.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification, 2019. URL https://arxiv.org/abs/1911.04252.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2024. URL https://arxiv.org/abs/2401.10020.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

A Engineering and JAX Implementation Details

Most directly related, Engstrom et al. (2025) introduce REPLAY, an algorithm for computing *exact* metagradients through iterative training by combining reverse-mode autodifferentiation with a checkpoint-replay data structure, and apply it to pretraining data selection on DataComp/CLIP, instruction-tuning data selection on Gemma-2B, data-poisoning, and learning-rate-schedule discovery. Whereas their setting optimises a single fixed importance-weight vector over a static candidate pool with a one-shot training run, our metagradients are recomputed each generation cycle over a freshly sampled synthetic batch, and we use a first-order (FOMAML-style) approximation rather than exact BPTT to keep per-cycle cost bounded.

Rematerialisation in the double-grad step. Step (iii) of the double-grad computation in Sec. B uses rematerialisation (`jax.checkpoint`, Griewank & Walther, 2000) so that activations are recomputed during the second-order backward rather than stored, keeping peak memory at $\approx 1\times$ a single forward–backward pass instead of $\approx 2\times$. The trick is compatible with custom VJP rules in the model (which block forward-mode AD) because it only ever calls reverse-mode differentiation.

Concrete cost of BPTT vs. GradAlign. The asymptotic comparison in Table 4 translates into substantial savings in practice. On a 4B model at our chosen inner batch size, exact BPTT peaks at $\sim$77 GB - approaching the limit of a single 80 GB accelerator and impractical to run at the larger inner batch sizes GradAlign permits - whereas **GradAlign** peaks at $\sim$11 GB and runs roughly $4\times$ faster. Concretely, on SAMSum with the 4B model a single generation cycle takes $\sim$57 minutes under **GradAlign**, with the BPTT cost projected at $\sim 4\times$ this from smaller-scale measurements where both can be run. Per-cycle wall-clock numbers across all model/dataset pairs are reported in Table 3; they range from $\sim$25 min on the smallest cell (Gemma-3 270M) to $\sim$67 min on the largest (GraniteMoE-3B).

Mixed-precision base model. The frozen base model is cast to `bfloat16` at load time. Since the base parameters appear in every forward pass and every VJP, their dtype controls the size of all stored activations during reverse-mode AD. Halving precision halves the peak memory of the second-order backward in the double-grad step (Sec. B); on the 4B model this is the difference between OOM and a $\sim$48 GB peak. LoRA adapters and the loss computation stay in float32: logits are sliced in `bfloat16` and only upcast to float32 after slicing, so the embedding-matmul VJP stores activations at 2 bytes per element rather than 4.

LoRA-only trajectory storage. Only the LoRA adapters need to be saved at each inner step; the base parameters are constant by construction. Trajectory storage is therefore $O(T \cdot |\phi|)$ rather than $O(T \cdot |\theta_0|)$, several orders of magnitude smaller for the model sizes considered.

Double-grad derivation. The compact form of the double-grad trick (Sec. B) is as follows. Define the scalar

$$h_t(w) \;\coloneqq\; \big\langle g_{\text{meta}},\; \nabla_\phi \mathcal{L}_{\text{train}}(\phi_t, w) \big\rangle \;=\; \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} w_i \, \big\langle g_{\text{meta}},\; \nabla_\phi \ell_i(\phi_t) \big\rangle. \tag{10}$$

Since h_t is linear in w, its gradient with respect to w returns all per-example alignments simultaneously:

$$|\mathcal{B}_t| \cdot \frac{\partial h_t}{\partial w_j} \;=\; \big\langle g_{\text{meta}},\; \nabla_\phi \ell_j(\phi_t) \big\rangle \qquad \text{for } j \in \mathcal{B}_t. \tag{11}$$

Computing $\nabla_w h_t(w)$ takes (i) one backward pass to obtain $\nabla_\phi \mathcal{L}_{\text{train}}(\phi_t, w)$, (ii) the inner product against g_{meta}, and (iii) a second-order backward differentiating h_t with respect to w. The total cost is two backward passes per inner step, independent of $|\mathcal{B}_t|$.

Retrain phase. The retrain step in line 20 of Algorithm 1 is configurable: it can use SGD, SGD with momentum, Adam, or AdamW, and it runs for one or more epochs over $\mathcal{D}_{\text{syn}}^{(m)}$. The retrain learning rate follows an optimizer-conditional batch-size scaling rule. Under plain SGD, where GradAlign's larger inner batch reduces the number of updates per epoch, we apply the linear scaling

rule LR $\propto$ batch size to compensate. Under SGD with momentum, Adam, or AdamW, no scaling is applied: the running statistics in these optimizers already amplify the effective step size, and stacking linear scaling on top of them caused divergence in 4B-scale ablations. We find that resuming from $\phi^{(m-1)}$ rather than retraining from scratch is important for stability across cycles.

The full $|\mathcal{D}_{\text{real}}|$-pair backward pass does not fit on a single accelerator at billion-parameter scale. We therefore partition $\mathcal{D}_{\text{real}}$ into micro-batches of size 1-10 (model-dependent) and accumulate $\nabla_\phi \mathcal{L}_{\text{meta}}$ across them. The per-iteration memory required for the meta-loss backward is then bounded by one micro-batch rather than the full pool, while every pair in $\mathcal{D}_{\text{real}}$ still contributes equally to the meta-gradient.

Hyperparameters. Table 2 lists the hyperparameters used in all experiments. Model-specific batch sizes are reported separately because exact BPTT does not fit into memory at the larger sizes used by GradAlign on the 4B model. For reproducibility, every run uses master PRNG seed 42 unless stated otherwise, covering LoRA initialisation, synthetic generation, and shuffling; the 20-example monitoring set is held fixed by the same seed.

Table 2: Hyperparameters used across all experiments. Model-specific batch sizes reflect the largest setting that fits in 80 GB of accelerator memory for each meta-gradient estimator.

Symbol	Meaning	Value
M	Generation cycles (outer loop)	7
K	Meta-iterations (middle loop)	5
T	Inner SGD steps per meta-iteration	75 (batch 16), 150 (batch 8)
N	Synthetic examples per cycle	1,200
n_{per}	Synthetic samples per real prompt	15
α	Inner-loop SGD learning rate	4×10^{-4}
η	Meta-loop Adam learning rate	0.25
$w_{\min}, w_{\max}$	Weight clipping bounds	0.01, 10
r, α_{LoRA}	LoRA rank, scaling	8, 16
LoRA targets	Attention projections adapted	q_{proj}, v_{proj}
Inner-loop batch size B (per model)		
Gemma-3 270M / 1B	BPTT / GradAlign	16 / 16
Gemma-3 4B	BPTT / GradAlign	2 / 16
Llama-3.2 3B	BPTT / GradAlign	4 / 8
Granite-MoE 3B	BPTT / GradAlign	4 / 8

Per-cycle wall-clock time. Table 3 reports the total wall-clock time of a single generation cycle across model/dataset pairs.

Table 3: **Wall-clock time per generation cycle** on a single NVIDIA Ada GPU (RTX 6000 Ada / L40S, 48 GB). Includes synthetic generation, K meta-iterations, retrain, and held-out evaluation. Times reported in minutes; lower is better.

Model	SAMSum	XSum	DialogSum
Gemma3-270m	26.8	24.4	46.4
Gemma3-1b	57.9	59.5	67.5
Gemma3-4b	57.0	50.6	72.0
Llama-3.2-3B	47.9	46.4	49.0
GraniteMoE-3B	66.4	62.0	64.0

B Cost Analysis and Empirical Validation of GradAlign

The double-grad trick: vectorising the per-example alignment A naive evaluation of (9) requires $|\mathcal{B}_t|$ backward passes per inner step, one per mini-batch element, which is prohibitive at our scales. Exploiting the linearity of $\mathcal{L}_{\text{train}}$ in w, all per-example alignments in a mini-batch can instead be recovered from the gradient with respect to w of a single scalar, $h_t(w) \coloneqq \langle g_{\text{meta}},\ \nabla_\phi \mathcal{L}_{\text{train}}(\phi_t, w)\rangle$. The total cost is *two backward passes per inner step, independent of* $|\mathcal{B}_t|$: across the trajectory, $2T$ backward passes plus one for g_{meta}.

Asymptotic per-meta-iteration cost. Table 4 summarises the asymptotic cost of the two estimators.

Table 4: Per-meta-iteration cost. T is the inner-loop length, $|\phi|$ the number of LoRA parameters, B the inner batch size.

	BPTT (exact)	**GradAlign (ours)**
Inner forward / save trajectory	T fwd + T bwd	T fwd + T bwd
Meta-loss gradient g_{meta}	1 bwd	1 bwd
Per-step alignment	1 second-order bwd	2 bwd (any B)
Memory	$O(T\|\phi\|)$ + adjoint chain	$O(T\|\phi\|)$
Pearson r to exact	1.0 (definition)	≥ 0.9977 (270M, see below)

Empirical agreement at production scale. The dominant cost of exact meta-gradient computation via truncated BPTT is its memory footprint. The **GradAlign** approximation of Sec. 3.3.2, paired with the double-grad implementation, estimates per-example utility at the cost of two backward passes per inner step regardless of batch size.

The two estimators remain closely aligned in practice. Across every meta-iteration of every generation cycle on the 270M model and three datasets (105 snapshots in total, 35 per dataset), we measure GradAlign and exact BPTT per-example meta-gradients and find Pearson correlation $r \geq 0.9977$ (pooled r between 0.9997 and 0.9999 per dataset), per-coordinate sign agreement above 99.4%, and a median magnitude ratio close to one. Figure 2 traces the step-by-step behaviour within a single inner trajectory.

The takeaway is that preserving gradient direction is sufficient for effective example weighting; the higher-order terms BPTT computes contribute little. Together, **GradAlign** and the double-grad trick make bilevel synthetic-data training tractable at billion-parameter scale. Concrete wall-clock and peak-memory numbers backing this claim are given above in App. A (Table 3).

Agreement beyond the 270M, 75-step setting. Table 5 varies model scale and inner-unroll length in turn from the 270M, $T = 75$ reference. Scale only sharpens the approximation: at 1B the agreement is tighter than the reference on every measure, with cosine similarity 0.99999 and sign agreement 99.88%. Longer unrolls preserve the quantity the weights actually depend on. At $T = 150$ the direction is essentially unchanged (cosine 0.99923, sign agreement 98.83%) while the magnitude drifts, raising the mean relative L_2 error from 0.0330 to 0.0585. That drift costs nothing here: the meta-optimiser is Adam, which is insensitive to per-coordinate scale, so a systematic error in norm is absorbed and the induced ranking of examples barely moves. We validate alignment up to 1B parameters and unroll lengths of $T = 150$.

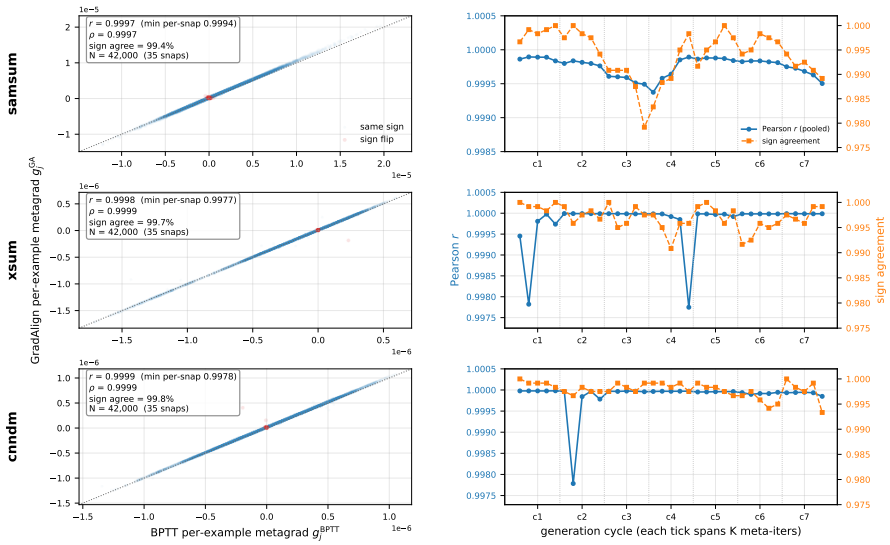


Figure 2: **Truncated exact BPTT vs. GradAlign meta-gradients**. At each inner-loop step t we compare the per-example meta-gradient produced by exact truncated BPTT against the GradAlign approximation. Cosine similarity near 1 confirms directional agreement; magnitude ratio near 1 confirms that no global rescaling is needed before passing the GradAlign metagradient to Adam.

Table 5: **GradAlign compared with exact BPTT at increased scale and unroll length.** Each row recomputes both meta-gradients from identical inner-parameter and weight snapshots; entries are mean $\pm$ s.d. over the indicated trajectory snapshots, each covering all $N = 1{,}200$ synthetic examples and evaluated in float64. The 1B run agrees more closely with exact BPTT than the 270M reference on every reported measure. At $T = 150$ the relative L_2 error is larger than at $T = 75$, while the gradient direction and the top-decile weight decision are preserved. Validation covers model sizes up to 1B parameters and unroll lengths up to $T = 150$.

Model	Data	T	Snapshots	Cosine	Sign agr.	Mag. ratio	Rel. L_2	Top-10% w	Bot-10% w
Gemma3-270m	SAMSum	75	35	0.99972±0.00026	0.9941±0.0049	1.0237±0.0039	0.0330±0.0086	0.9850±0.0185	0.9548±0.0331
Gemma3-1b	SAMSum	75	15	0.99999±0.00001	0.9988±0.0012	1.0042±0.0003	0.0063±0.0010	0.9972±0.0040	0.9967±0.0042
Gemma3-270m	SAMSum	150	15	0.99923±0.00084	0.9883±0.0099	1.0447±0.0096	0.0585±0.0170	0.9950±0.0069	0.9550±0.0281
Gemma3-270m	XSum	75	35	0.99984±0.00044	0.9971±0.0024	1.0009±0.0023	0.0121±0.0135	0.9976±0.0043	0.9979±0.0037
Gemma3-270m	CNN/DM	75	35	0.99988±0.00035	0.9979±0.0015	1.0063±0.0013	0.0138±0.0096	0.9981±0.0041	0.9936±0.0064

C Expanded Evaluation and Robustness

C.1 Paired uncertainty on the primary grid

Table 6 aggregates the paired bootstrap tests that underlie the main result. On perplexity the separation is complete: all 75 comparisons favour MWeiST and none of the intervals includes zero. On completion cross-entropy the record is 63 significant improvements, 10 inconclusive comparisons, and two cells in which a baseline is ahead. The reference-overlap metrics are far less decisive, and we read them as a check that the likelihood gains do not come at the cost of surface quality rather than as independent evidence of improvement.

Table 6: **Paired 95% bootstrap results on the 500-example held-out sets, aggregated over 15 model–dataset cells.** Counts are *significant wins / overlaps / significant losses*. The two column totals follow from the design: the 15 model–dataset cells are compared against two direct fine-tuning baselines, giving $15 \times 2 = 30$ paired tests, and against three self-training baselines, giving $15 \times 3 = 45$. ROUGE-L and BLEU complement the ChrF metric in Table 1. The likelihood metrics separate the methods strongly; the reference-overlap metrics mostly do not. The single CCE loss in each baseline group occurs on Gemma-3 4B/DialogSum.

Metric	vs. LoRA, DoRA (30)	vs. Conf. Filter, ReST, L2RW (45)
PPL↓	30/0/0	45/0/0
CCE↓	24/5/1	39/5/1
ChrF↑	7/15/8	13/30/2
ROUGE-L↑	4/20/6	15/26/4
BLEU↑	4/23/3	13/29/3

C.2 Sensitivity to data splits and stochastic seeds

The re-split study in Table 7 retrains the full pipeline on each new split rather than rescoring a fixed checkpoint, so it measures the variability of the method itself and not only of the evaluation set. It comprises 180 runs over all 15 model–dataset blocks, each with all four arms. In every block, the mean perplexity gap to LoRA exceeds MWeiST's cross-split standard deviation; the ratio ranges from $1.3\times$ to $81\times$.

Split variability and seed variability are separable, and we measure the second directly. Holding the split fixed on Gemma-3 270M/SAMSum and varying only the end-to-end seed from 42 to each of 1, 2, and 3, the four MWeiST runs span 0.18 in perplexity and 0.006 nats in completion cross-entropy, against gaps of at least 1.2 and 0.019 to LoRA and DoRA on the same split.

Table 7: **Data-split sensitivity over three re-seeded 80/20 splits.** The full pipeline is retrained from scratch and evaluated on 500 held-out examples; entries are mean ± sample standard deviation over splits. MWeiST has lower PPL in 90 of 90 same-split comparisons with LoRA and DoRA and in 43 of 45 with uniform self-training; the CCE records are 82 of 90 and 42 of 45.

		PPL↓				CCE↓			
Model	**Dataset**	MWeiST	Uniform ST	LoRA	DoRA	MWeiST	Uniform ST	LoRA	DoRA
Gemma3-270m	SAMSum	19.833±0.015	20.051±0.017	21.066±0.006	21.053±0.003	2.2184±0.0050	2.2933±0.0049	2.2463±0.0010	2.2428±0.0005
Gemma3-270m	XSum	16.582±0.076	16.607±0.073	16.727±0.071	16.722±0.072	2.3202±0.0045	2.3235±0.0043	2.3255±0.0046	2.3246±0.0048
Gemma3-270m	DialogSum	10.925±0.014	10.965±0.020	11.384±0.015	11.371±0.017	2.2896±0.0045	2.3106±0.0041	2.3035±0.0037	2.3004±0.0041
Gemma3-1b	SAMSum	14.848±0.012	15.046±0.006	15.606±0.011	15.594±0.011	1.7923±0.0033	1.8607±0.0052	1.8292±0.0013	1.8273±0.0007
Gemma3-1b	XSum	11.238±0.040	11.240±0.039	11.302±0.039	11.298±0.039	1.8068±0.0058	1.8064±0.0054	1.8082±0.0058	1.8079±0.0058
Gemma3-1b	DialogSum	8.189±0.022	8.226±0.018	8.395±0.014	8.392±0.015	1.8157±0.0032	1.8299±0.0027	1.8352±0.0014	1.8338±0.0017
Gemma3-4b	SAMSum	11.993±0.040	12.248±0.128	12.657±0.010	12.668±0.016	1.5339±0.0072	1.5839±0.0151	1.5847±0.0026	1.5854±0.0023
Gemma3-4b	XSum	8.660±0.028	8.678±0.028	8.775±0.026	8.775±0.025	1.5270±0.0043	1.5291±0.0049	1.5351±0.0044	1.5347±0.0063
Gemma3-4b	DialogSum	6.534±0.008	6.832±0.047	6.797±0.008	6.796±0.008	1.5388±0.0038	1.6086±0.0085	1.5375±0.0022	1.5370±0.0024
Llama-3.2-3B	SAMSum	12.246±0.023	12.510±0.028	12.420±0.030	12.424±0.022	1.5476±0.0029	1.5865±0.0080	1.5665±0.0011	1.5657±0.0020
Llama-3.2-3B	XSum	9.509±0.039	9.549±0.033	9.706±0.029	9.705±0.026	1.6024±0.0059	1.6117±0.0079	1.6124±0.0075	1.6123±0.0063
Llama-3.2-3B	DialogSum	6.879±0.014	6.983±0.008	7.222±0.015	7.221±0.012	1.5628±0.0115	1.6816±0.0139	1.5533±0.0032	1.5539±0.0026
GraniteMoE-3B	SAMSum	8.880±0.024	8.924±0.013	8.963±0.027	8.946±0.028	1.3345±0.0019	1.3586±0.0043	1.3419±0.0017	1.3416±0.0002
GraniteMoE-3B	XSum	6.748±0.018	6.746±0.019	6.771±0.016	6.768±0.018	1.2991±0.0029	1.3012±0.0037	1.3081±0.0012	1.3075±0.0039
GraniteMoE-3B	DialogSum	5.373±0.017	5.480±0.017	5.517±0.032	5.534±0.011	1.3078±0.0042	1.3608±0.0057	1.3241±0.0078	1.3275±0.0006

C.3 ADDITIONAL TASK FORMATS

We apply the same scarce-data protocol beyond the three primary summarisation datasets to see how far the weighting signal carries. Against the three self-training baselines the likelihood metrics continue to favour MWeiST on all three datasets, most clearly on CNN/DailyMail, where it improves CCE in all 15 comparisons. GSM8K marks the boundary of the regime: exact match stays at the noise floor for every method, so none of them acquires arithmetic reasoning from 80 examples.

Table 8: **Additional datasets evaluated on 100 held-out examples.** Counts are *significant wins/overlaps/significant losses* from paired 95% bootstrap intervals over five models and five comparison methods. CNN/DailyMail and GSM8K expand the 20-example monitoring evaluations to 100 held-out examples, while Alpaca adds instruction following. MWeiST holds a perplexity advantage on all three datasets.

Dataset	Metric	vs. LoRA, DoRA (10)	vs. CF, ReST, L2RW (15)
CNN/DM	PPL↓	6/2/2	12/0/3
	CCE↓	4/3/3	15/0/0
	ChrF↑	0/10/0	0/14/1
	ROUGE-L↑	0/10/0	0/14/1
	BLEU↑	0/10/0	1/13/1
GSM8K	PPL↓	9/1/0	13/0/2
	CCE↓	4/1/5	13/2/0
	ChrF↑	2/8/0	6/9/0
	ROUGE-L↑	0/8/2	5/10/0
	BLEU↑	2/8/0	4/11/0
Alpaca	PPL↓	6/3/1	7/4/4
	CCE↓	0/4/6	9/3/3
	ChrF↑	0/10/0	4/10/1
	ROUGE-L↑	2/5/3	4/8/3
	BLEU↑	0/8/2	1/12/2

Table 9: **JFLEG grammar correction and E2E data-to-text generation on 500 held-out examples.** Entries are point estimates ± one bootstrap standard error; **bold** marks the best available value. The weighting signal transfers to both task formats: MWeiST cuts E2E perplexity well below direct fine-tuning at 270M (42.5 against 47.1) and at 1B (31.0 against 36.7), and at 4B it leads JFLEG on perplexity, ChrF, and BLEU. On 4B E2E, MWeiST leads on CCE while ReST leads on perplexity and ChrF; the paired bootstrap over the 500-example pool separates MWeiST from every baseline on CCE ($p < 10^{-4}$ in all five comparisons).

		JFLEG				E2E NLG			
Model	**Method**	PPL↓	CCE↓	ChrF↑	BLEU↑	PPL↓	CCE↓	ChrF↑	BLEU↑
Gemma3-270m	MWeiST (ours)	24.890±0.723	**0.9570±0.0289**	12.907±1.347	1.915±0.357	42.535±0.691	2.4158±0.0254	**27.171±0.258**	**1.513±0.076**
	LoRA	24.552±0.704	0.9668±0.0292	**13.199±1.368**	1.970±0.364	47.114±0.779	2.4220±0.0257	26.796±0.258	1.482±0.074
	DoRA	24.698±0.710	0.9639±0.0292	13.104±1.366	1.966±0.364	46.210±0.759	**2.4128±0.0257**	26.831±0.260	1.483±0.074
	Confidence Filter	24.734±0.717	0.9622±0.0290	12.997±1.347	1.927±0.359	42.730±0.694	2.4544±0.0255	26.882±0.256	1.462±0.073
	ReST	**24.551±0.710**	0.9624±0.0290	12.927±1.343	1.883±0.351	**42.211±0.684**	2.4469±0.0255	26.858±0.256	1.463±0.073
	L2RW	26.808±0.797	0.9708±0.0292	13.068±1.345	**2.063±0.381**	48.767±0.819	2.4367±0.0258	26.544±0.256	1.438±0.072
Gemma3-1b	MWeiST (ours)	19.636±0.514	0.7612±0.0220	12.581±1.383	3.147±0.580	31.041±0.526	**1.9803±0.0252**	27.323±0.277	1.684±0.084
	LoRA	21.609±0.588	0.7595±0.0234	**12.823±1.388**	3.147±0.574	36.703±0.638	2.0137±0.0259	27.104±0.279	1.617±0.082
	DoRA	21.341±0.577	**0.7568±0.0233**	12.814±1.382	3.125±0.569	36.443±0.632	2.0121±0.0259	27.155±0.280	1.630±0.082
	Confidence Filter	19.655±0.514	0.7692±0.0220	12.665±1.382	3.129±0.575	31.880±0.535	2.0524±0.0252	26.826±0.265	1.627±0.082
	ReST	19.546±0.508	0.7746±0.0218	12.597±1.380	3.148±0.581	31.032±0.520	2.0283±0.0251	27.056±0.269	1.658±0.084
	L2RW	21.967±0.603	0.7597±0.0233	12.682±1.391	3.153±0.576	36.998±0.648	2.0165±0.0260	27.131±0.281	1.640±0.085
Gemma3-4b	MWeiST (ours)	**12.311±0.281**	0.6598±0.0205	**12.751±1.291**	**7.189±1.248**	15.658±0.225	**1.6226±0.0229**	26.749±0.254	1.519±0.074
	LoRA	20.585±0.568	0.6595±0.0208	12.186±1.320	3.991±0.731	21.820±0.358	1.6504±0.0229	26.709±0.259	**1.613±0.077**
	DoRA	20.983±0.590	**0.6592±0.0207**	12.129±1.320	4.077±0.746	21.788±0.357	1.6503±0.0230	26.709±0.259	1.610±0.077
	Confidence Filter	17.233±0.450	0.6706±0.0206	12.412±1.325	3.883±0.711	15.733±0.224	1.6574±0.0232	26.572±0.256	1.512±0.075
	ReST	16.912±0.439	0.6688±0.0205	12.373±1.327	3.816±0.702	**14.506±0.202**	1.6326±0.0230	**26.844±0.257**	1.537±0.075
	L2RW	21.167±0.600	0.6597±0.0207	12.294±1.319	3.597±0.660	21.858±0.360	1.6524±0.0230	26.724±0.264	1.600±0.076

C.4 COMPUTATIONAL EFFICIENCY AND RUNTIME ANALYSIS

The asymptotic analysis in Appendix B isolates the cost of computing a meta-gradient; Table 10 reports the measured cost of running the pipelines end to end. The meta-loop is paid for in wall-clock

time rather than in memory. Peak memory is almost unchanged at 270M and rises visibly only for the 3B mixture-of-experts model, which is consistent with the overhead being dominated by the inner-loop passes rather than by adapter size, so MWeiST fits wherever direct fine-tuning already does.

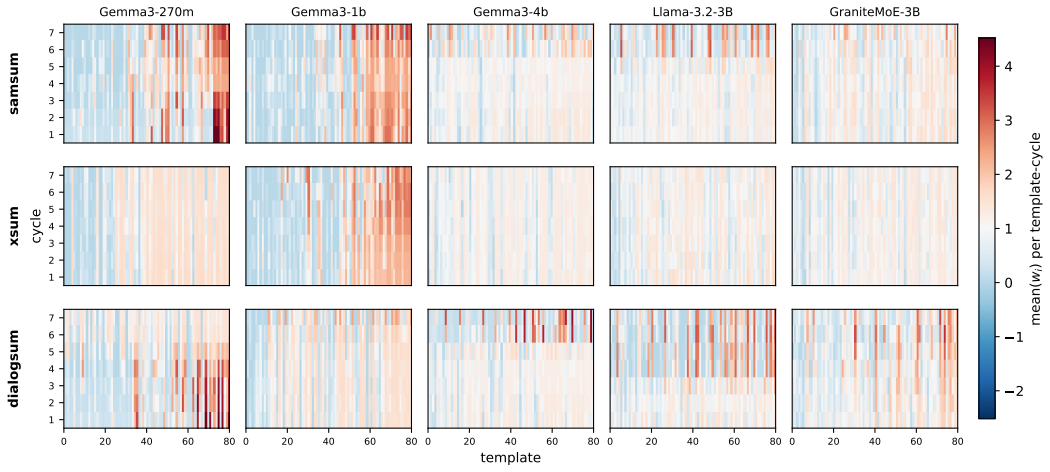
Table 10: **Wall-clock runtime and peak GPU memory across methods on SAMSum.** Each run uses one GPU: an RTX PRO 6000 Blackwell for Granite-MoE 3B or an RTX 6000 Ada for Gemma-3 270M. The Total column includes held-out scoring. LoRA and DoRA perform this scoring inside the training script, so their Train and Total values coincide. MWeiST costs $3.9\times$ LoRA at 270M and $6.8\times$ at 3B, buying 1.2 PPL and 0.03 nats CCE at 270M. Gemma-3 270M time and memory average the two splits with matched evaluation cadence, and its PPL/CCE column is the three-split mean of Table 7.

Model	Method	Train (h)	Total (h)	Peak mem (GiB)	PPL / CCE
GraniteMoE-3B	MWeiST (ours)	7.01	8.85	29.9	8.797 / 1.3359
	LoRA	1.30	1.30	24.9	8.918 / 1.3388
	DoRA	1.28	1.28	24.5	8.914 / 1.3389
Gemma3-270m	MWeiST (ours)	2.55	2.90	27.3	19.833 / 2.2184
	Uniform ST	0.67	1.02	27.2	20.051 / 2.2933
	LoRA	0.74	0.74	27.1	21.066 / 2.2463
	DoRA	0.77	0.77	27.1	21.053 / 2.2428

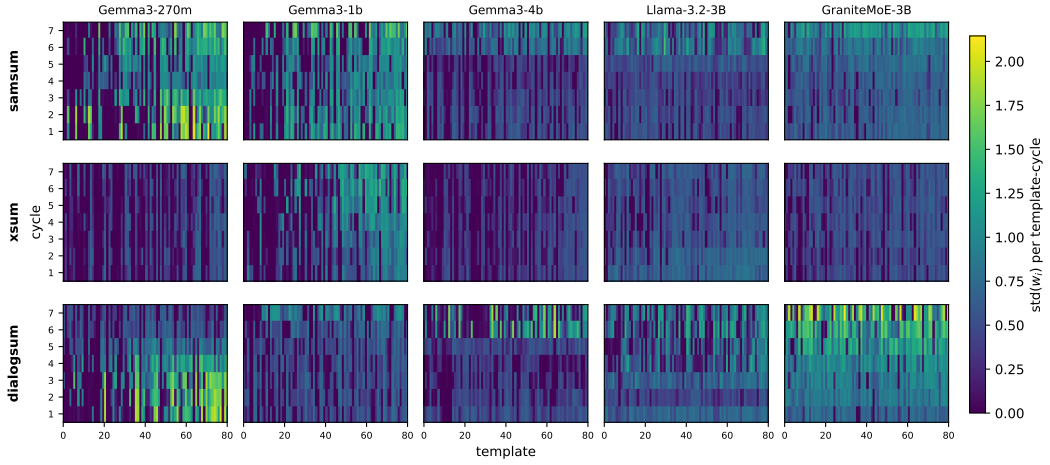
D Analysis of Learned Weights

Figure 3 visualises the learned per-template importance weights w_i across cycles. The heatmaps reveal which templates are consistently up- or down-weighted and provide qualitative evidence that MWeiST recovers a non-trivial weighting structure rather than collapsing to the uniform solution. Two complementary observations make this concrete. First, the between-template spread of mean weights (Fig. 3a) is large: the standard deviation of per-template mean weights across the 80 templates typically falls in the range 0.4-1.2 on a scale where the global mean is normalised to 1.0, so different templates receive systematically different mean weights rather than clustering around $w \approx 1$. Second, the within-template spread (Fig. 3b) is comparable in magnitude: the mean per-template standard deviation across the n_{per} synthetic completions of a single template typically lies in 0.3-0.8, indicating that even within one template's completions some samples are upweighted strongly while others are suppressed. If MWeiST were merely filtering at the template level (labelling some prompts "good" and others "bad"), within-template variance would collapse toward zero; if it were producing near-uniform weights, both spreads would. Neither degeneracy is observed, in any (model, dataset) cell. The vertical structure in the mean-weight heatmaps indicates that MWeiST's per-template preferences are stable across cycles: the same templates are consistently favored or disfavored throughout training, while the substantial within-template variance confirms that this stability does not collapse into mere template-level filtering.

Figure 4 shows the marginal distribution of learned weights at the end of training, which is consistent with the spread statistics above. The weighting structure is consistent with per-completion quality assessment guided by the validation signal, which is precisely the methodological claim of Section 3.



(a) Mean per-template weight across cycles.



(b) Std of per-template weights across cycles.

Figure 3: **Per-template importance-weight statistics across cycles**. For each (model, dataset) we show the per-cycle mean (a) and standard deviation (b) of the learned importance weights w_i aggregated over the k synthetic samples drawn from each of the n_{tr} templates. Rows within each sub-grid are datasets; columns are models.

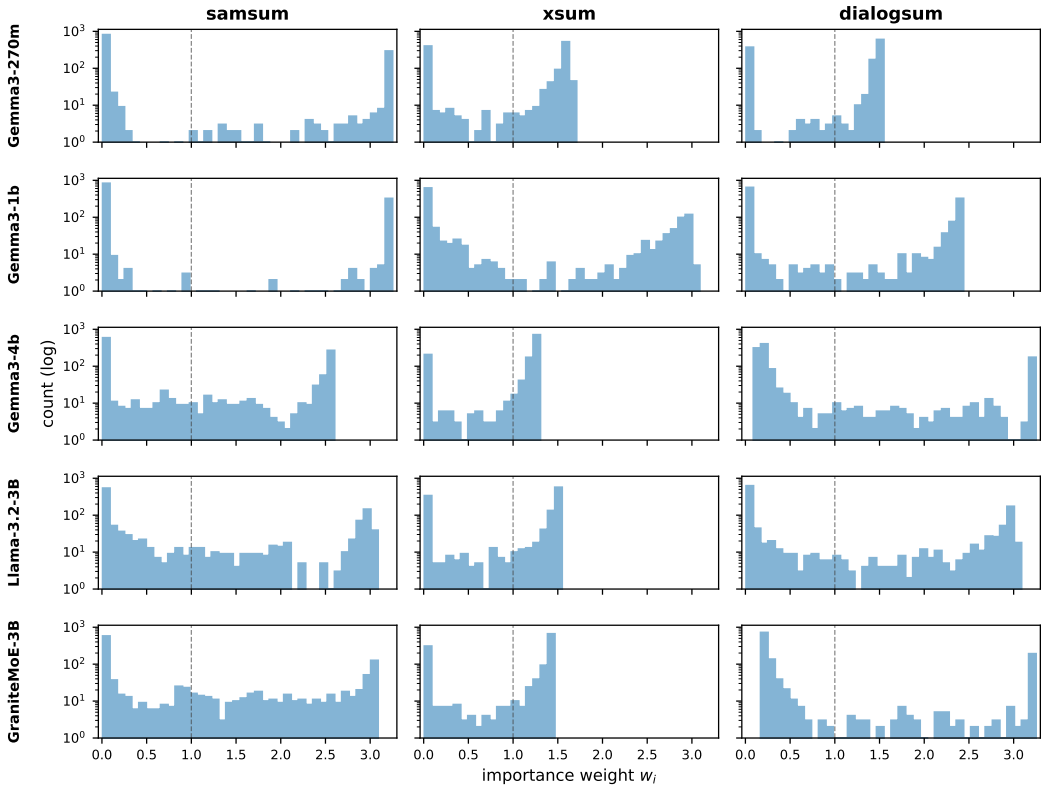


Figure 4: **Distribution of final per-example importance weights**. Each panel shows the histogram of learned importance weights w_i over the N synthetic training samples at the end of the last generation cycle. Rows are datasets, columns are models.

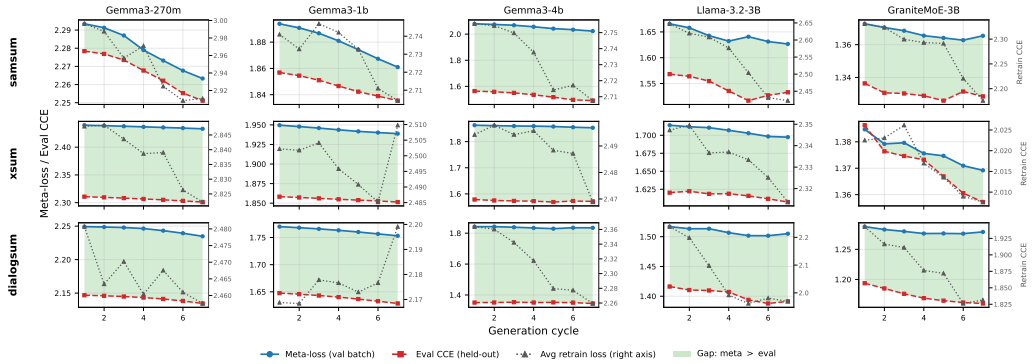


Figure 5: **Meta-learning's prediction vs. post-retrain reality**. Per-cell, end-of-cycle meta-loss (solid, blue) is the causal CE the look-ahead inner loop would attain on the meta batch; eval CCE (dashed, red) is the held-out CCE on the 20-example monitoring set after the full retrain step; and avg retrain loss (dotted, grey; right axis) is the synthetic-batch training CCE. Rows are datasets, columns are models.

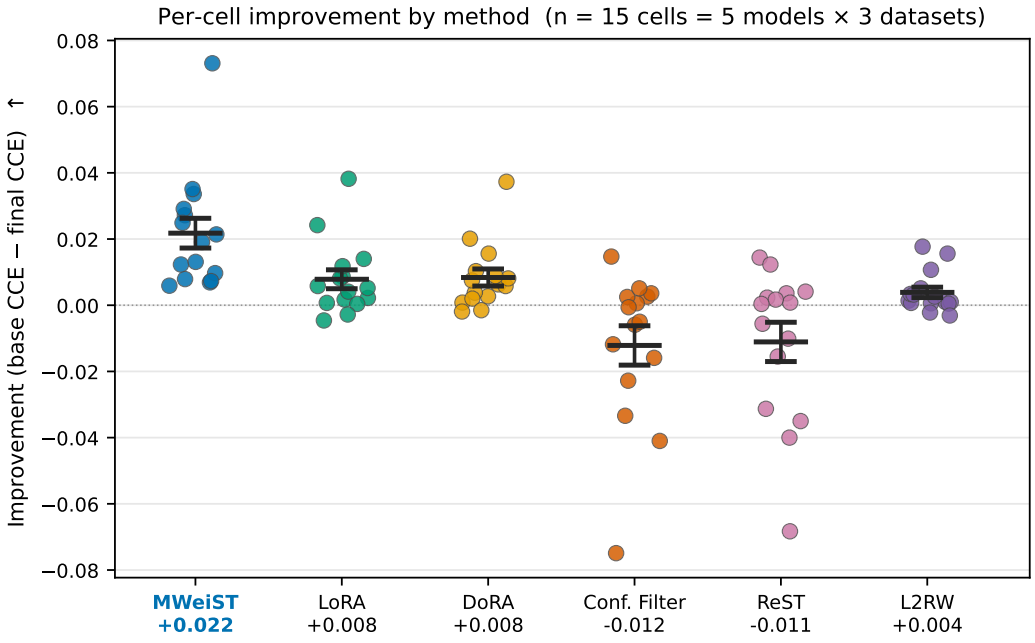


Figure 6: Per-cell CCE improvement by method (n=15 cells, 5 models × 3 datasets).

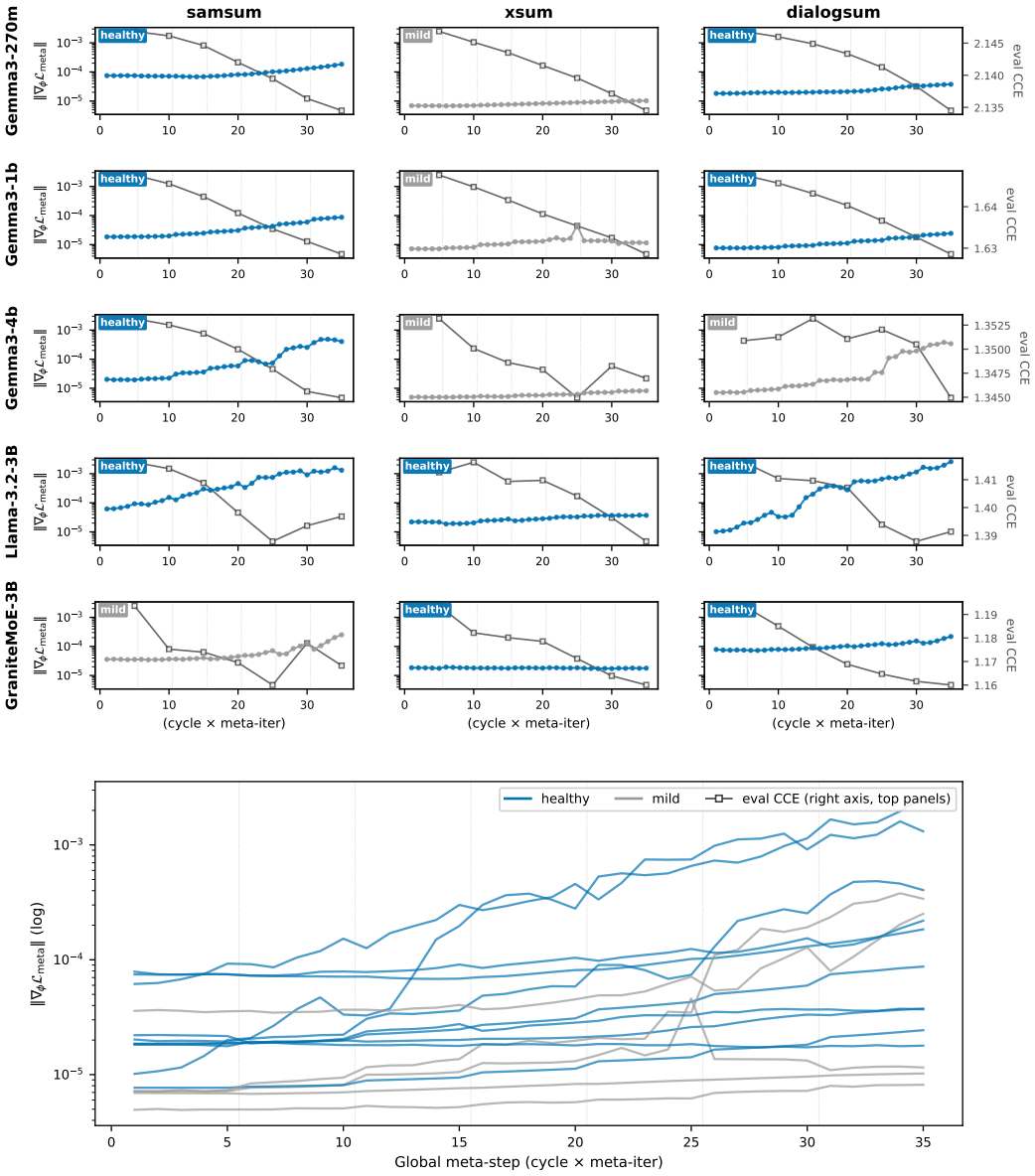


Figure 7: Meta-gradient norm $\|\nabla_\phi \mathcal{L}_{\text{meta}}\|$ across the 35 inner-loop steps (M cycles $\times$ K meta-iters) for the 15 main-grid MWeiST runs. Top: per-(model, dataset) small multiples on a shared log-y axis; right axis (rightmost column) shows per-cycle eval CCE. Bottom: overlay of all 15 trajectories. Cells are tagged *healthy* or *mild* based on the sign of the cycle-1$\rightarrow$ cycle-7 eval-CCE delta. For most models, healthy learning (blue) coincides with a meta-grad norm that rises across training, while mild cells (grey) stay flat. GraniteMoE-3B breaks this pattern - two of its three cells are healthy yet the meta-grad norm remains in the lowest band ($\sim 10^{-5}$).

SAMSum / Gemma-3 1B. Elizabeth had an argument with Blackett today, but she doesn't want William to intermeddle.

SAMSum / Gemma-3 1B. Blake believes that Alexis is trying to turn their son, Steven, against him and has bad influence on Steven. Alexis believes Blake is self-centered and often attacks her and accuses her of things she didn't do.

CNN/DM / Gemma-3 4B. Colony collapse disorder has killed millions of bees . Scientists suspect a virus may combine with other factors to collapse colonies . Disorder first cropped up in2004, as bees were imported from Australia . $15 billion in U.S. crops each year dependent on bees for pollination .

CNN/DM / Gemma-3 4B. Orme, Tennessee, has running water from6 p.m. to9 p.m. Town's spring ran dry in midst of Southeast drought . Donated water is trucked in from Alabama . Completion of pipeline will solve problem for good .

CNN/DM / Gemma-3 4B. Amount almost double what was spent in2004 election cycle . Lower TV production costs help more candidates advertise . Mitt Romney leads presidential candidates in TV spending . Advertisers face challenge of cutting through clutter of ads, analyst says .

Figure 8: **Per-token contribution to the MWeiST meta-loss**. Each gold-completion token is shown with a red background whose darkness reflects its cross-entropy $-\log p_\phi(b_t \mid a, b_{<t})$ under the cycle-7 LoRA-adapted model. Intensities are normalised *within each example*, so shading is best interpreted as relative within-row contrast rather than absolute surprisal. In the CNN/DM examples, higher-loss tokens are frequently content-bearing words, including proper nouns (*Romney*, *Tennessee*, *Australia*), domain verbs (*combine*, *cropped*, *cutting*), and numeric quantities (*$15 billion*), rather than nearby function words. A similar, though less pronounced, pattern appears in the SAMSum examples, where some function words also receive relatively high loss (e.g. *with* in Example 1). Sentence-initial tokens can also receive higher loss, likely because the model is transitioning from a completed sentence to a new topic; the observed variation should therefore be read as partly positional rather than purely lexical. Displayed examples come from an 80-example sweep and were chosen by content-vs-function-word surprisal ratio. Sub-word BPE pieces are merged into a single highlight per word, with intensity defined as the maximum over sub-pieces.

E TOP VS BOTTOM TEMPLATES BY MWEIST WEIGHT

To inspect what MWeiST learns to up-weight versus down-weight, we display the three templates with the highest and three with the lowest per-template weights at cycle 7 (the final generation cycle) for two representative (model, dataset) cells. Each card pairs the truncated prompt with its gold reference and the synthetic completion produced by the cycle-7 model; the index-aligned weight w is shown in the card's title bar. Top-weighted templates (green) are those whose synthetic generation the meta-objective favoured; bottom-weighted (red) are suppressed.

Llama-3.2-3B / DialogSum

− Most-favoured templates

Template 73 $w = 3.008$

Prompt:	Dialogue: #Person1#: What kind of music do you like listening to? #Person2#: I like music that has a fast beat and is lively, like dance music. You know, I go to a disco almost every week. Sometimes…
Gold:	#Person2# likes dance music while #Person1# prefers classical music. #Person1# suggests #Person2# listen to more classical music because it can make the brain more active and reduce stress. #Person2# tells #Person1# cla…
Synthetic:	Dialogue between two persons about their preferences regarding different types musical styles The dialogue deals mainly around topic related towards differences among individuals tastes concerning various genres such ty…

Template 46 $w = 3.008$

Prompt:	Dialogue: #Person1#: Would you talk to me about taking one of Dr. Miller's classes? #Person2#: Yes. Have you ever been in one of his classes? #Person1#: I'm thinking about taking his class next semes…
Gold:	#Person1# is considering taking Dr. Miller's class but worries about grades. #Person2# tells #Person1# Dr. Miller was an easy and interesting teacher and gaining knowledge is as important as grades.
Synthetic:	Person #1 wants advice from person #2 on whether or not they should take an upcoming course offered at their school taught by Professor John Mille... The following content is provided under creative commons license. cli…

Template 28 $w = 3.005$

Prompt:	... Review. My name is Richard Clarke and I am joined today by the very erudite David Watson. #Person2#: Thank you Dick. Today we are going to talk about the movie Lion King. Tell me Dave, what is yo…
Gold:	Richard Clarke and David Watson discuss Lion King on Premier Movie Review. Richard thinks it's a fine movie as it has strong animation, great score and songs, and an enchanting story, while David thinks the movie lacks…
Synthetic:	"Lion King" (1994) - Full Movie In 1989 Disney released its first animated feature since their classic Cinderella over twenty years earlier; Snow White & The Seven Dwarfs had come up with something new again but…

− Most-disfavoured templates

Template 3 $w = 0.011$

Prompt: Dialogue: #Person1#: Hey, man. What's up? #Person2#: Ah, first of all, I put a buck in the vending machine for a seventy-five cent candy bar, and the thing got stuck here in the machine. Then, I pres...

Gold: #Person2# tells #Person1# he put a buck in the vending machine but the thing got stuck. After hitting the machine, #Person2# gets the candy bar.

Synthetic: A person has problems with their coin-operated candy dispenser, because they can't get any coins or cash from him/her A friend is having trouble getting his/her quarters/cash/gifts etc., which are used as currency when...

Template 2 $w = 0.011$

Prompt: Dialogue: #Person1#: Have you had any experience with sales work? #Person2#: Yes, I have. In fact, I had worked for a foreign trade company for nearly two years, where I established business ties wit...

Gold: #Person1# is interviewing #Person2# and asks #Person2# several questions, including experience with sales work, the work at the previous company, and why #Person2# left there.

Synthetic: The dialogue shows us how Person 2 decides on joining another firm after leaving his former one because he found little room for growth here while lacking adequate contacts outside the area; but then again by becoming p...

Template 5 $w = 0.011$

Prompt: Dialogue: #Person1#: Why do you want to spend your summer days in a mountain village? #Person2#: I think a mountain village is splendid! #Person1#: I don't agree with you. It's too small to live. #Pe...

Gold: #Person2# thinks spending summer days in a mountain village is splendid while #Person1# prefers a big town.

Synthetic: Person 1 thinks living in mountains would be boring because there aren't many things going on around them while person two disagrees saying he loves being surrounded by nature which makes him feel peaceful inside himsel...

Gemma3-1B / XSum

– Most-favoured templates

Template 39 $w = 3.015$

Prompt: Article: The 23-year-old has had surgery for what the Pro12 club describe as "an ongoing shoulder complaint". Edinburgh estimate that the former Scotland Under-20 and Scotland Sevens player will be s...

Gold: Edinburgh scrum-half Sean Kennedy has been ruled out for the rest of the season through injury.

Synthetic: Former England international Chris Paterson says he's been left 'devastated' by news of his friend Jamie Kennedy being ruled out for at least three months after suffering an injury in training this week.

Template 47 $w = 2.992$

Prompt:	Article: The destination of a stimulus package worth nearly £39m was agreed as part of the Welsh government's budget deal with the Liberal Democrats. Finance Minister Jane Hutt said it would help "ge…
Gold:	Funding to improve school buildings and create apprenticeships has been announced to help revive the economy.
Synthetic:	Cardiff City Council leader says 'we need to do better than this'. (Credit Image:... * Share or comment on this article latest

Template 63 $w = 2.988$

Prompt:	Article: Weaknesses in the way mice swapped data with computers left them vulnerable, said security firm Bastille Networks. Attackers could spoof poorly protected signals letting them use PCs as if t…
Gold:	Hackers could gain access to home and corporate networks via security flaws in wireless mice, suggests research.
Synthetic:	The authors say hackers can easily steal information transmitted over USB-connectedw

– Most-disfavoured templates

Template 0 $w = 0.011$

Prompt:	Article: The Scottish capital's trams began running on Saturday, after six years of disruption and cost increases. Mr Salmond cited "considerable public concern" over the Â£776m project, which he sai…
Gold:	Edinburgh's troubled tram project will be investigated by a judge-led inquiry, Scottish First Minister Alex Salmond has announced.
Synthetic:	A new tram system opened today (12 March) linking central London's West End district with Heathrow airport – but not without controversy The UK Government announced plans last year to build Britain's largest ever urban…

Template 2 $w = 0.011$

Prompt:	Article: The IPC opened proceedings against the National Paralympic Committee of Russia after a report claimed the country had operated a widespread doping programme. A decision on any ban will come…
Gold:	The International Paralympic Committee will make its decision on Russia's participation in the Rio Games "in the interest of sport", president Sir Philip Craven has told BBC Sport.
Synthetic:	A meeting between representatives of both organisations took place at an early stage yesterday morning before being adjourned until tomorrow afternoon when decisions were expected to be made regarding whether the Russia…

Template 5 $w = 0.011$

Prompt:	Article: Prosecutors accuse Fifa president Sepp Blatter of making a "disloyal payment" of $2m (Â£1.6m) to Mr Platini. Mr Platini has provided information to the investigation but said he did so as a…
Gold:	Michel Platini, the head of European football's governing body Uefa, is being treated as "in between a witness and an accused person" by Swiss prosecutors investigating corruption at the world football authority Fifa.
Synthetic:	* The former French football chief Jean Pierre Escalettes died yesterday aged 74 after suffering heart failure following surgery remove

Llama-3.2-3B / CNN/DM

Most-favoured templates

Template 36 $w = 1.720$

Prompt: ... point that some scholars speculate the song was actually written "for Therese," as in Therese Malfatti, one of several women who turned down a marriage proposal from the notoriously lovesick maes...

Gold: Several famous songs written by men for a woman . Beethoven's "Für Elise" may have been written "for Therese" Songwriters make up stories about who is behind "867-5309/Jenny"

Synthetic: There are many songs about love but few say anything meaningful or profound. Love Songs About Love There's nothing like listening to romantic music when your heart needs soothing up; there really isn't! I mean, we've go...

Template 10 $w = 1.718$

Prompt: ... first elected female president. Pratibha Patil, the ruling coalition's 72-year-old nominee for the mainly ceremonial post, easily beat opposition-backed challenger and vice president, Bhairon Sin...

Gold: India elects first female president, official results show Saturday . Pratibha Patil's supporters are calling victory a boost for women's rights . Bitter election campaign was marked by scandal . 72-year-old Patil was t...

Synthetic: ...first elected female president. Pratibha Patil, the ruling coalition's President-elect Pratibha Devisingh Rau (or Devi) Patil will be sworn in at noon today before President K.R.Narayanan amid tight security arrangem...

Template 30 $w = 1.717$

Prompt: ... third place in the closing race, but saw his 12-point lead cut to four by McLaren team-mate Fernando Alonso in Shanghai. The Spaniard finished second behind Ferrari's Kimi Raikkonen, who also kep...

Gold: Lewis Hamilton fails to clinch world title after spinning out of the Chinese GP . Briton's lead cut to four points by McLaren team-mate Fernando Alonso . Alonso finished second in Shanghai behind Ferrari's Kimi Raikkone...

Synthetic: ... third place in the closing race, but saw his 12 -point lead cut to four by McLaren team- mate Fernando Alonso... World Rally Championship | Formula One FormulaOne.com is your source for all things related to formula...

Most-disfavoured templates

Template 0 $w = 0.009$

Prompt: ... opposite second-rower Shaw was noted by citing commissioner Dennis Wheelahan. Chabal started the match on the substitutes' bench, but was brought on in the 26th minute to replace the injured Fabi...

Gold: France lock Sebastien Chabal cited for a dangerous tackle on Simon Shaw . Chabal faces disciplinary hearing on Monday after incident against England . Sale forward will miss the third and fourth-place play-off is he is...

Synthetic: A former international referee says justice has been done over an incident involving Wales player Dwayne Peel. MOSCOW (Reuters) – Former FIFA president Sepp Blatter walked free from detention Monday as Russian authoriti...

Template 1 $w = 0.009$

Prompt:	... city went through as insurgents battled for control of the city. Mohammed stops at his cousin's grave. "I feel an ache when I think of her. Every time I remember her, I cry," he told CNN at a vis...
Gold:	Boy, 12, lost leg to car bombing in Iraq; cousin was killed by blast . He is now heading to the United States to get a prosthetic leg . Boy: "I want to ... meet this person that gave me my future back" Charity director...
Synthetic:	... is better than what we had here," said Mohamed Al Saleh who has been living there since July last year. Summary: A man whose sister-in-law returned missing after being kidnapped by al Qaeda militants asked Wednesday...

Template 5 $w = 0.009$

Prompt:	... r – now at 3,114 – is expected to rise yet further as the South Asian nation continues to assess the damage. The Bangladesh Red Crescent fears the death toll could be 5,000 – perhaps even reac...
Gold:	Survivors of Cyclone Sidr in Bangladesh waiting for aid to arrive . Red Crescent Society official says deaths could rise to 5,000-10,000 . Bangladeshi president distributes aid; angry survivors go empty-handed . Thousan...
Synthetic:	... nbsp;Bangladesh is bracing itself against what will likely become one of Asia’s worst natural disasters ever when Cyclonic Storm Sidra hits its southwestern coastline tonight or tomorrow. Keywords: bangkok hon...

F Other Ablations

F.1 Comparison with instruct-tuned models

Table 11: Per-token cross-entropy (CCE, lower is better). For each row we compare the MWeiST-trained cycle-7 checkpoint against the base model and three progressively stronger off-the-shelf instruct-tuned baselines (raw prefix, chat template, and chat + 8-shot ICL). The lowest CCE in each row is **bolded**.

Dataset	Model	Base (cycle 0)	MWeiST (cycle 7)	Instruct (raw)	Instruct (chat)	Instruct (chat + 8-shot)
SAMSum	Gemma-3 270M	2.2794	**2.2479**	2.4299	3.2738	3.3274
	Gemma-3 1B	1.8586	**1.8374**	2.4916	6.3515	4.6055
	Gemma-3 4B	1.5636	**1.4909**	2.1665	8.6205	3.5216
	Llama-3.2 3B	1.5672	**1.5330**	1.6935	2.4194	2.5239
	Granite-MoE 3B	1.3354	**1.3284**	1.7561	2.4374	1.6276
XSum	Gemma-3 270M	2.3118	**2.3011**	3.7734	4.8971	4.7722
	Gemma-3 1B	1.8594	**1.8514**	3.5579	8.5202	6.8955
	Gemma-3 4B	1.5760	**1.5711**	3.0679	10.1114	5.4414
	Llama-3.2 3B	1.6211	**1.6072**	1.9569	2.8898	2.6261
	Granite-MoE 3B	1.3542	**1.3376**	2.0380	2.5125	1.7693
CNN/DM	Gemma-3 270M	2.7093	**2.7035**	3.5034	4.4020	3.8082
	Gemma-3 1B	2.3860	**2.3824**	3.3434	8.4643	5.2806
	Gemma-3 4B	2.1531	**2.1386**	2.8314	8.0105	3.3962
	Llama-3.2 3B	2.2293	**2.2283**	2.5364	3.3259	2.5104
	Granite-MoE 3B	1.6839	**1.6749**	2.3715	2.6064	1.9131

Table 12: Hugging Face paths for the five base models used throughout the main paper and their instruct-tuned counterparts used as the chat-template and 8-shot baselines.

Model	Base	Instruct
Gemma-3 270M	`google/gemma-3-270m`	`google/gemma-3-270m-it`
Gemma-3 1B	`google/gemma-3-1b-pt`	`google/gemma-3-1b-it`
Gemma-3 4B	`google/gemma-3-4b-pt`	`google/gemma-3-4b-it`
Llama-3.2 3B	`meta-llama/Llama-3.2-3B`	`meta-llama/Llama-3.2-3B-Instruct`
Granite-MoE 3B	`ibm-granite/granite-3.0-3b-a800m-base`	`ibm-granite/granite-3.0-3b-a800m-instruct`

Throughout the main paper we focus on the pre-trained (base) versions of the five models in our grid (see Table 12 for the exact Hugging Face checkpoints used here and for the matching instruct-tuned counterparts). Table 11 reports per-token CCE on the held-out summarisation splits. For each base model we compare the MWeiST-trained cycle-7 checkpoint against three progressively stronger off-the-shelf instruct-tuned baselines on the same eval prefix. The *raw prefix* variant scores the literal `Dialogue:\n...\nSummary:` (or `Article: ...\nSummary:`) prefix that MWeiST trains on. The *chat template* variant wraps that prefix as a user turn via `tokenizer.apply_chat_template(..., add_generation_prompt=True)` and scores the reference summary as the assistant turn, reflecting the instruct model's actual deployment surface. The *chat + 8-shot* variant prepends eight hand-picked (article, summary) demonstrations drawn from the same 80 train examples MWeiST sees, in chat-format alternating turns, plus an explicit per-dataset style instruction.

Across all 15 (model, dataset) cells, MWeiST cycle-7 achieves the lowest per-token CCE on the held-out reference summaries, beating both the base checkpoint and all three instruct baselines. The 8-shot ICL variant meaningfully closes the chat-template gap on the larger Gemma-3-it models, e.g. Gemma-3 4B / SAMSum drops from 8.62 (chat) to 3.52 (chat + 8-shot), confirming that demonstrations recover much of the style information that chat conditioning alone obscures, but not enough to match a model fine-tuned on the same 80 examples. The tightest MWeiST-vs-ICL margins ($\sim$0.24–0.30 nats) appear on the Llama and Granite cells where the chat-template tax was smallest to begin with; the widest margins ($>$1 nat) appear on the Gemma-3-it cells whose chat conditioning is most aggressive. We report a single seed.

F.2 K AND T TRADE-OFF

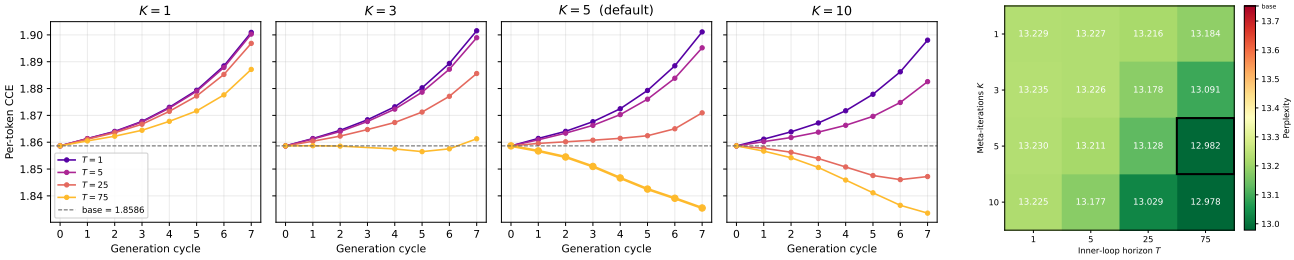


Figure 9: **Sensitivity of MWeiST to the meta-iteration count K and inner-loop horizon T on Gemma-3 1B / SAMSum**. *Left:* per-token CCE per generation cycle, faceted by K with one curve per T. *Right:* cycle-7 perplexity (lower = better).

F.3 DEPENDENCE ON THE NUMBER OF UNIQUE TEMPLATES n_{tr}

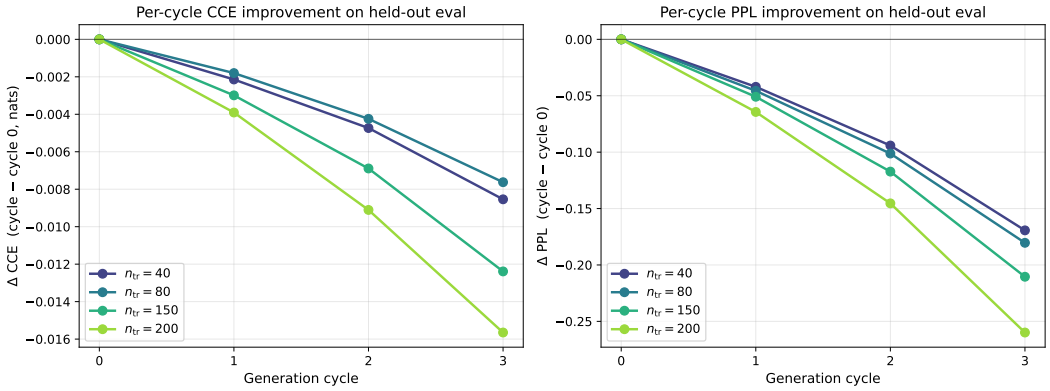


Figure 10: **Behaviour of MWeiST across the data-scarce regime ($n_{\text{tr}} \in \{40, 80, 150, 200\}$).** Gemma-3 1B on SAMSum, with n_{tr} varying and every other knob held fixed: synthetic pool $|\mathcal{D}_{\text{syn}}| = 1200$ regardless of n_{tr}, $K = 5$ meta-iterations per cycle, $T = 75$ inner-SGD steps, $M = 3$ generation cycles. *Left:* Per-cycle Δ CCE = $\text{CCE}_m - \text{CCE}_0$. Every cell descends monotonically across cycles; none of the four data regimes flips the sign of MWeiST's effect, and the magnitude of the gain grows monotonically with n_{tr}: $\Delta\,\text{CCE} \approx -0.008$ nats at $n_{\text{tr}} \in \{40, 80\}$, -0.012 at $n_{\text{tr}} = 150$, and -0.016 at $n_{\text{tr}} = 200$. *Right:* Per-cycle Δ PPL on the same held-out eval. The PPL trajectories track the CCE story: monotonic improvement across cycles in every cell, with the largest pool ($n_{\text{tr}} = 200$) descending fastest. The default $n_{\text{tr}} = 80$ that we apply in the rest of the paper sits inside the regime where MWeiST works: gains persist at half the data ($n_{\text{tr}} = 40$) and grow at $2.5\times$ the data ($n_{\text{tr}} = 200$). We frame this experiment as a robustness check across the data-scarce regime rather than as an ablation of MWeiST: n_{tr} is a property of the problem setting, not a method hyperparameter.

F.4 RESETTING θ EACH META-ITERATION

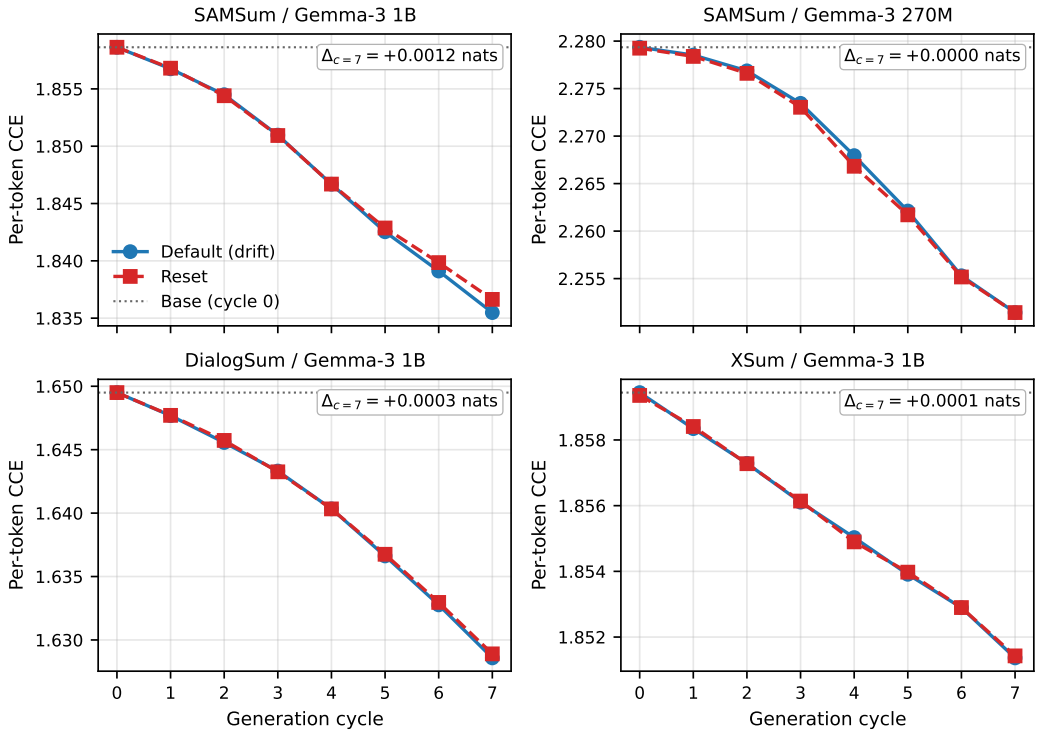


Figure 11: **Robustness of MWeiST to resetting the inner-loop parameters θ between meta-iterations.** Held-out per-token CCE across cycles 0 to 7 for four (model, dataset) configurations at the default setting (K=5 meta-iterations, T=75 inner-SGD steps, M=7 generation cycles), under two regimes. *Default* (blue, solid): θ is snapshotted once at the start of each cycle and is allowed to drift across the K meta-iterations as each metagrad probe extends the inner trajectory. *Reset* (red, dashed): θ is re-snapshotted to its cycle-start value at the start of every meta-iteration, so that each of the K probes constitutes an independent T-step trajectory from the same initialisation. The dotted line denotes the base model at cycle 0. The annotation in each panel reports $\Delta_{c=7}$, the per-token CCE difference between Default and Reset at the final cycle. In all four configurations the difference does not exceed $+0.0012$ nats, and is exactly 0 for SAMSum / Gemma-3 270M; the two trajectories are visually overlapping for the early cycles in every panel. These differences lie well within the seed-noise band calibrated by the $K\times T$ grid ablation, in which the K=5 $\rightarrow$ K=10 step at T=75 already shifts cycle-7 CCE by $+0.0019$ nats. We therefore adopt the no-reset variant in the main method for implementation simplicity. The MAML-style reset rationale does not transfer to the present setting: MAML resets θ to a shared initial value θ_0 in order to align gradient information across parallel tasks, whereas the K meta-iterations in MWeiST constitute sequential refinements of a single weight vector w evaluated on the same synthetic data and the same held-out meta-loss, and therefore there are no parallel tasks to align. In addition, the cycle-end θ is re-derived by retraining from the cycle-start θ using the final w_K, irrespective of the trajectory followed by the meta-loop, so any inner-loop drift can influence subsequent cycles only indirectly through w_K. The present figure confirms that this indirect pathway carries no material signal in the LoRA setting studied here, which is what makes the no-reset variant the safe simplification.

F.5 WEIGHTING AND OPTIMIZER CONTROLS

The most direct ablation fixes every synthetic weight to one. Table 13 shows a clear meta-weighting benefit on SAMSum and DialogSum, and a smaller but consistent one on XSum, where MWeiST still leads perplexity in all five cells.

Table 13: **Meta-weighting isolated against uniform self-training.** In the 14 matched cells, both arms share prompts, generation settings, sampling budget, optimiser, and schedule; the intervention is whether synthetic examples receive meta-learned or uniform weights. Values are from cycle 7 on the 20-example monitoring set. MWeiST leads on PPL in 14/14 matched cells and CCE in 13/14.

		PPL↓		CCE↓	
Model	**Dataset**	MWeiST	Uniform ST	MWeiST	Uniform ST
Gemma3-270m	SAMSum	**17.422**	17.598	**2.2511**	2.3176
	XSum	**16.248**	16.273	**2.3011**	2.3061
	DialogSum	**10.136**	10.178	**2.1345**	2.1547
Gemma3-1b	SAMSum	**12.982**	13.237	**1.8355**	1.9030
	XSum	**10.639**	10.660	**1.8514**	1.8569
	DialogSum	**7.410**	7.451	**1.6286**	1.6444
Gemma3-4b	SAMSum	**10.513**	10.740	**1.4909**	1.5610
	XSum	**8.124**	8.137	1.5711	**1.5679**
	DialogSum	**5.987**	6.131	**1.3449**	1.3493
Llama-3.2-3B	SAMSum	**10.469**	10.698	**1.5581**	1.6687
	XSum	**8.455**	8.491	**1.6072**	1.6169
	DialogSum	6.235	**5.742**	**1.3914**	1.7311
GraniteMoE-3B	SAMSum	**7.901**	8.153	**1.3397**	1.4045
	XSum	**6.134**	6.137	**1.3572**	1.3631
	DialogSum	**4.889**	5.026	**1.1600**	1.2225

Reward weights as an alternative signal. To contrast meta-learned weighting with an alternative paradigm, we examine Group Relative Policy Optimization (GRPO) as a representative reward-driven baseline. A second control replaces the meta-gradient weight with a group-relative advantage derived from a scalar reward, leaving the rest of the self-training loop untouched: the same frozen backbone with rank-8 LoRA adapters, the same $G = 15$ completions per prompt and $N = 1200$ synthetic samples per cycle, the same sampling temperature and repetition penalty, and the same SGD optimiser with Nesterov momentum 0.9 at each model's retrain learning rate, re-initialised at the start of every cycle. Writing $\widetilde{Y}_{i,j}$ for the j-th completion sampled from prompt X_i and Y_i for the corresponding real completion, the reward and advantage are

$$r_{i,j} = \mathrm{Reward}\big(\widetilde{Y}_{i,j},\, Y_i\big) \in [0,1], \qquad \hat{A}_{i,j} = \mathrm{clip}\left(\frac{r_{i,j} - \mu_i}{\sigma_i + \varepsilon_{\mathrm{norm}}},\, -5,\, +5\right), \tag{12}$$

Here μ_i and σ_i are the mean and standard deviation of $\{r_{i,j}\}_{j=1}^{G}$ within the group generated from X_i, and $\varepsilon_{\mathrm{norm}} = 10^{-4}$ stabilises the normalisation when reward variance is nearly zero. We use two reward functions: ROUGE-L F1 against Y_i, and sentence-level ChrF against Y_i rescaled to $[0,1]$. The real completion Y_i enters only through this scalar and is never used as a token target, so the arm receives no teacher forcing on real completions. Each advantage is broadcast to every sampled token of its completion and enters the clipped policy-gradient surrogate

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{\sum_{i,j,t} m_{i,j,t} \min\big(\rho_{i,j,t}\hat{A}_{i,j},\ \mathrm{clip}(\rho_{i,j,t},\, 1-\varepsilon_{\mathrm{clip}},\, 1+\varepsilon_{\mathrm{clip}})\,\hat{A}_{i,j}\big)}{\sum_{i,j,t} m_{i,j,t}}, \tag{13}$$

in which $m_{i,j,t}$ masks prompt and padding positions and $\rho_{i,j,t}$ is the per-token importance ratio of the current policy against a snapshot taken immediately after generation. We use $\varepsilon_{\mathrm{clip}} = 0.1$, one inner epoch per cycle, and no KL penalty towards the base policy; cycles stop once held-out perplexity fails to improve twice in succession, which every one of the 90 runs reached within fourteen cycles. The contrast with MWeiST is what the multiplier measures: a reward scores a completion as an output, whereas a meta-gradient scores it as training data, and the two need not agree.

The control covers SAMSum and DialogSum, the two benchmarks on which both arms were scored under an identical evaluation protocol. Table 14 reports how often MWeiST beats the better of the two reward variants and Table 15 gives the underlying scores: MWeiST leads on perplexity in all ten cells and on completion cross-entropy in nine of ten. Scoring a completion as an output and scoring it

Table 14: **Reward weights instead of meta-gradients.** We replace the meta-gradient weights inside the same self-training loop with GRPO-style ROUGE-L or ChrF rewards, following Eq. (12) and Eq. (13); each reward variant uses three seeds across five models. Each cell counts the models in which the submitted MWeiST run beats the better mean reward variant on the 20-example monitoring set; Table 15 gives the underlying scores. Both methods draw a fresh synthetic pool from their current policy at every cycle; their controlled difference is the weighting signal.

Dataset	**PPL↓**	**CCE↓**	**ChrF↑**	**ROUGE-L↑**
SAMSum	5/5	4/5	4/5	3/5
DialogSum	5/5	5/5	4/5	2/5
Both datasets	10/10	9/10	8/10	5/10

as training data are different judgements, and on these benchmarks the second is the one that transfers.

Finally, Table 16 compares persistent retrain configurations used after the meta-weight update. The result motivated the adopted Nesterov configuration: selecting Adam by whole-sequence PPL would obscure deterioration in the target-completion objective.

G Broader Impact

MWeiST adapts language models from very small amounts of real data, lowering the cost of customising models for new tasks, languages, and domains and reducing the data-collection burden in settings where collecting more supervision is expensive or invasive. As a self-training method, it inherits the well-known risk that repeated training on model-generated data can drift away from the underlying real distribution and amplify biases in the base model (Shumailov et al., 2023). The teacher-forced meta-loss on a fixed real pool partially counteracts this drift by anchoring updates to real data, but it does not eliminate the failure mode, and the learned weights themselves can encode and amplify biases present in the small real pool. We caution against deployment in settings where the real pool is not representative of the target use of the model.

Table 15: **Raw scores behind the reward-weighting control.** Point scores for the submitted MWeiST run and for both GRPO reward variants on the 20-example monitoring set. GRPO entries are mean ± sample standard deviation over three seeds; MWeiST is a single run. The win counts of Table 14 can be read directly off this table: MWeiST is compared against the better of the two reward variants, the more favourable reading for GRPO, and still leads perplexity in every cell. Seed spread on the likelihood metrics is at most 0.025 PPL and 0.017 nats CCE.

		SAMSum		DialogSum	
Model	**Method**	PPL↓	CCE↓	PPL↓	CCE↓
Gemma3-270m	MWeiST (ours)	17.422	2.2511	10.136	2.1345
	GRPO (ROUGE-L reward)	18.586±0.013	2.2590±0.0120	10.579±0.009	2.1468±0.0018
	GRPO (ChrF reward)	18.578±0.013	2.2597±0.0136	10.571±0.002	2.1423±0.0004
Gemma3-1b	MWeiST (ours)	12.982	1.8355	7.410	1.6286
	GRPO (ROUGE-L reward)	13.758±0.025	1.8206±0.0172	7.592±0.004	1.6292±0.0075
	GRPO (ChrF reward)	13.745±0.013	1.8522±0.0115	7.603±0.005	1.6511±0.0009
Gemma3-4b	MWeiST (ours)	10.513	1.4909	5.987	1.3449
	GRPO (ROUGE-L reward)	11.333±0.012	1.5561±0.0082	6.153±0.005	1.3504±0.0009
	GRPO (ChrF reward)	11.341±0.012	1.5720±0.0039	6.152±0.003	1.3527±0.0029
Llama-3.2-3B	MWeiST (ours)	10.608	1.5330	6.235	1.3914
	GRPO (ROUGE-L reward)	10.944±0.002	1.5744±0.0014	6.671±0.003	1.4160±0.0007
	GRPO (ChrF reward)	10.932±0.006	1.5731±0.0003	6.662±0.002	1.4143±0.0014
GraniteMoE-3B	MWeiST (ours)	7.926	1.3284	4.889	1.1600
	GRPO (ROUGE-L reward)	8.032±0.007	1.3325±0.0003	5.161±0.005	1.1993±0.0009
	GRPO (ChrF reward)	8.040±0.009	1.3352±0.0009	5.160±0.008	1.1969±0.0042

		SAMSum		DialogSum	
Model	**Method**	ChrF↑	R-L↑	ChrF↑	R-L↑
Gemma3-270m	MWeiST (ours)	28.42	0.1254	25.53	0.0998
	GRPO (ROUGE-L reward)	26.18±0.04	0.1190±0.0005	24.95±0.13	0.0963±0.0008
	GRPO (ChrF reward)	26.07±0.14	0.1187±0.0002	25.09±0.21	0.0972±0.0001
Gemma3-1b	MWeiST (ours)	27.87	0.1026	25.21	0.0938
	GRPO (ROUGE-L reward)	26.04±0.35	0.1178±0.0083	24.80±1.19	0.0957±0.0038
	GRPO (ChrF reward)	25.67±0.07	0.1076±0.0017	24.48±0.25	0.0953±0.0022
Gemma3-4b	MWeiST (ours)	28.29	0.1282	24.67	0.0980
	GRPO (ROUGE-L reward)	25.53±0.07	0.1233±0.0009	24.27±0.05	0.0991±0.0002
	GRPO (ChrF reward)	25.52±0.11	0.1187±0.0034	24.28±0.02	0.0987±0.0011
Llama-3.2-3B	MWeiST (ours)	23.48	0.0708	18.50	0.0484
	GRPO (ROUGE-L reward)	20.24±0.02	0.0755±0.0005	20.10±0.15	0.0601±0.0012
	GRPO (ChrF reward)	20.13±0.03	0.0758±0.0000	20.16±0.05	0.0602±0.0004
GraniteMoE-3B	MWeiST (ours)	23.30	0.0829	24.30	0.0907
	GRPO (ROUGE-L reward)	23.43±0.03	0.0825±0.0002	24.23±0.02	0.0895±0.0001
	GRPO (ChrF reward)	23.43±0.05	0.0824±0.0001	24.27±0.01	0.0898±0.0005

Table 16: **Persistent retrain-configuration comparison on Gemma-3 1B/SAMSum.** All four arms use meta-learning and share the same 75-step plain-SGD look-ahead, and differ only in the persistent retrain optimizer and step budget listed in the table. Adam attains the lowest PPL but raises the target-completion CCE, which is why we adopt SGD with Nesterov momentum and select by CCE.

	Retrain	**PPL↓**		**CCE↓**	
Retrain optimizer	**steps**	cycle 7	Δ from cycle 0	cycle 7	Δ from cycle 0
SGD	75	13.717	−0.035	1.8570	−0.0016
SGD + Nesterov 0.9 (adopted)	75	12.983	−0.769	1.8356	−0.0231
Adam	5	11.648	−2.104	1.8778	+0.0191
Adam	10	12.355	−1.397	2.3285	+0.4699

Cycle-0 base model: PPL 13.752, CCE 1.8586.

H AI USE, ETHICS AND REPRODUCIBILITY STATEMENTS

AI USE STATEMENT

AI tools were used to assist with text polishing and routine tasks. All scientific content, claims, and conclusions remain the responsibility of the authors.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets and publicly released pretrained checkpoints; it involves no human subjects, no collection of personal data and no annotation studies. The method lowers the cost of adapting language models from very little real data; its benefits and risks, including the propagation of model errors through self-training, are discussed in Appendix G.

REPRODUCIBILITY STATEMENT

Section 3 specifies the algorithm and Section 4.1 the experimental protocol. Appendix A gives the JAX implementation of the double-gradient GradAlign estimator, the complete hyperparameter settings (Table 2), and the fixed master seed and what it governs; Table 12 lists the exact pretrained checkpoints, and Tables 3 and 10 report wall-clock and memory costs. Appendix B validates GradAlign against exact BPTT. Appendix C describes the held-out evaluation protocol, the paired bootstrap confidence intervals, and the seed and split sensitivity study (Table 7).