Making Open-Source Text LLM Watermarks Durable Against Merging

Luisa Scharff [1] **Thibaud Gloaguen** [1] **Robin Staab** [1] **Martin Vechev** [1]

Abstract

Open-source LLMs (OSMs) are reaching near state-of-the-art performance, prompting prior works to trace the text they generate by embedding text watermarking algorithms directly into their weights. Yet, OSMs are subject to post-training modifications, which has been shown to remove the watermark. Model merging in particular, a prominent method used for combining expert knowledge and preventing catastrophic forgetting, strongly removes such OSMs watermarks. A key question is how to enable OSM watermarks that survive subsequent merging. In this work, we show for the first time how to design an OSM watermark that is durable against model merging. We propose Merge-Adversarial Training, an adversarial training algorithm to distill text watermarks into model weights while being robust to subsequent model merging. Our approach consistently outperforms all baselines (e.g. with SLERP up to +51 percentage points (pp) TPR@1%FPR with +25 pp on average) while preserving downstream capabilities. We also for the first time evaluate OSM watermarks against realistic merge scenarios, representing common use-cases such as combining expert capabilities or preventing catastrophic forgetting, and with 3 prominent merging algorithms. More broadly, our findings suggest that adversarial training is a reliable approach for increasing OSM watermark durability against post-training modifications.

1. Introduction

Large language model (LLM) watermarking is reaching maturity, becoming increasingly integrated into regulatory frameworks (Hamon et al., 2024) and consumer-facing products (Dathathri et al., 2024). Most existing watermarking methods are based on modifying the LLM sampling procedure to imprint a human-invisible but detectable signal in the generated output. Crucially, these methods are *generation-time* and designed for closed-source models served via an API, and are thus inapplicable to open-source models.

Open-source LLM watermarks Given the widespread adoption of capable open-source models (Qwen Team, 2026; DeepSeek-AI, 2026; Team et al., 2026), the community has increasingly shifted its focus to OSM watermarks, which directly embed the watermarking behavior into model weights. Prior work (Gloaguen et al., 2025) has highlighted a new challenge for OSM watermarks: *durability* under common OSM modifications. Among these, *model merging*, which combines two or more trained models to join their capabilities, has proven to be surprisingly adverse to OSM watermarks. This is concerning because model merging is common and cost-effective: merged models routinely rank near the top of leaderboards (Beeching et al., 2023), and over 40 thousand are publicly shared on Hugging Face. Thus, model merging poses a significant operational challenge: even non-adversarial merges of a released watermarked model may inadvertently remove the watermark.

This work: OSM watermarks durable against merging In this work, we introduce the first OSM watermark specifically designed to be durable against model merging. Specifically, building on the popular watermark-distillation approach (Gu et al., 2024), we propose *Merge-Adversarial Training* (MAT), a watermark-distillation objective explicitly designed for robustness to model merging.

MAT builds on an adversarial-training-inspired meta-loss component that directly includes a low-cost merge operation inside the watermark training loop: at each step, we temporarily create a merged model that interpolates the current checkpoint with a reference model, compute the watermark-distillation loss through this merge, and backpropagate gradients only to the current model. Our experimental evaluation shows that MAT yields a significantly more durable watermark compared to both standard watermark-distillation approaches and other OSM watermarks (Block et al., 2025) while retaining the same model performance as standard watermark-distillation.

Beyond our method, we introduce a rigorous evaluation pipeline for OSM durability under model merging. Un-

[1]ETH Zurich. Correspondence to: Luisa Scharff <lscharff@ethz.ch>.

Accepted at the ICML 2026 Workshop on Agents in the Wild: Safety, Security, and Beyond. Copyright 2026 by the author(s).

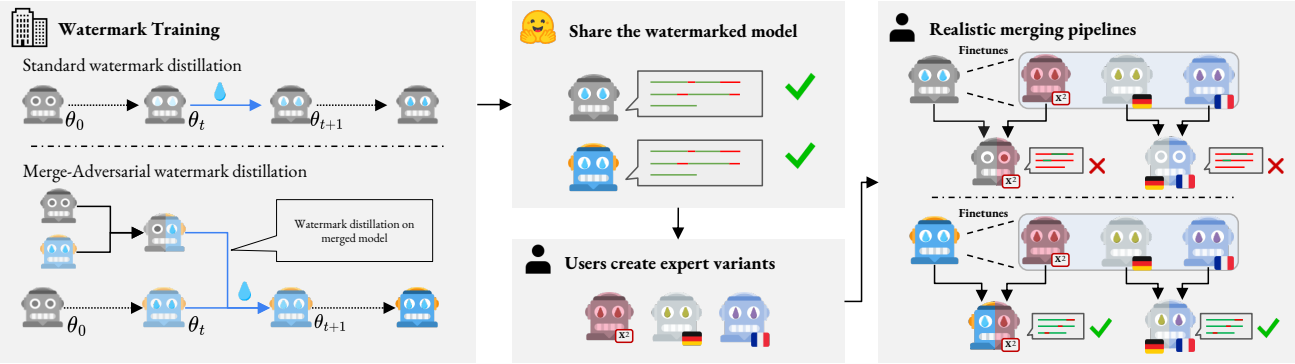

Figure 1. **Overview of our method and evaluation:** (*Left*) Unlike standard watermark distillation, we train the watermark using an adversarial objective: at each step, we simulate a merge operation and optimize for it to remain watermarked. (*Middle*) After training, both standard distillation and our model are watermarked as long as they are not modified. (*Right*) Yet, users might finetune the model and create complex merged models (e.g., to prevent forgetting or to combine capabilities). While standard watermark distillation loses its watermark, our method preserves it consistently.

like prior work, which only merges the watermarked model with the unwatermarked base model, we evaluate durability across realistic scenarios, including multiple fine-tuned models, multi-round merging, and three merging algorithms: LINEAR, SLERP, and TIES. At the same time, our experimental results establish the evaluation of durability under unwatermarked base model merging as an effective worst-case proxy for faster OSM watermarking evaluations.

Our contributions We make the following contributions:

1. We introduce MAT, an adversarial-training-based objective (Sec. 3) that simulates model merging during training, directly optimizing our watermark to be durable against merging.
2. We show that MAT consistently improves watermark detectability under realistic merge scenarios and multiple watermarking schemes, significantly outperforming standard watermark distillation while maintaining downstream performance.
3. We demonstrate that OSM watermarks can be trained in a way that provides additional downstream benefits (e.g., durability against merging).

2. Background and Related Work

In this section, we review prior work on LLM watermarking, with a particular focus on approaches for open-source models. We further provide the relevant background on model merging.

Generation-time LLM watermarking LLM watermarking aims to embed a statistically detectable signal into generated text, enabling later attribution to a particular model or provider. Most existing schemes operate at generation time: given a private key, they modify the sampling procedure to insert a detectable key-dependent signal into the model output. The most prominent class is the Red-Green family (KGW) (Kirchenbauer et al., 2023a), where a pseudorandom function partitions the vocabulary into green and red tokens and biases generation toward green tokens. Other popular schemes include KTH (Kuditipudi et al., 2024), AAR (Aaronson, 2023), and SynthID (Dathathri et al., 2024). Unlike Red-Green, these schemes are distortion-free: they preserve the model distribution in expectation over the private key. However, all methods require modifying the LLM sampling procedure, making them incompatible with the user-controlled open-source setting.

Watermarks for open-source models Watermarking an open-source model requires embedding the signal into the model weights, so that standard inference already produces watermarked text (i.e., text with a detectable watermark signal). This setting has only recently received systematic attention (Gloaguen et al., 2025). Existing approaches broadly fall into gradient-based methods (Gu et al., 2024; Xue et al., 2025; Gloaguen et al., 2026), which train the model to internalize a watermark, and weight-editing methods (Block et al., 2025), which directly modify model parameters. Watermark distillation (Gu et al., 2024) is the most widely used gradient-based approach: a student model is trained to imitate the output distribution of a watermarked teacher, embedding the watermark behavior into its weights. Meanwhile, weight-editing methods inject the watermark via direct weight modifications requiring no additional training (Block et al., 2025). Crucially, while Xue et al. (2025) improves durability against finetuning, no prior work directly improves OSM watermark durability against other common post-training modifications such as merging.

Robustness, security, and durability Prior work on LLM watermark reliability has studied several distinct failure modes. Text-level robustness asks whether a watermark

remains detectable after edits to generated text, such as paraphrasing, translation, deletion, or insertion (Kirchenbauer et al., 2023a; Kuditipudi et al., 2024; Kirchenbauer et al., 2023b). Security work studies adversarial attribution manipulation, including spoofing attacks that generate text falsely detected as watermarked and scrubbing attacks that remove watermarks from generated text (Sadasivan et al., 2023; Jovanović et al., 2024; He et al., 2024; Pang et al., 2024). These directions primarily target generation-time watermarks and text-level adversaries.

Open-source watermarking introduces an additional requirement: *durability* under model modifications. Because OSMs are routinely finetuned, quantized, and merged after release, useful OSM watermarks should remain detectable after such transformations. Importantly, recent work evaluating existing OSM watermarks under common post-release modifications finds that current methods remain fragile in these settings (Gloaguen et al., 2025): OSM watermarks may survive mild compression-like transformations, but degrade sharply under stronger modifications such as merging.

Model merging Model merging constructs a new model by combining the weights of two or more independently trained checkpoints without additional training. Simple linear interpolation averages model parameters directly, while SLERP (Goddard et al., 2024) interpolates on the hypersphere. Task-vector methods such as Task Arithmetic (Ilharco et al., 2022) interpret finetuning updates as directions in weight space that can be added, subtracted, or combined. TIES (Yadav et al., 2023) and DARE-TIES (Yu et al., 2024b) further modify task-vector merging by resolving sign conflicts, pruning updates, or reducing interference between parent models. These methods are attractive in open-source development because they are inexpensive, require no data or retraining, and can combine specialized finetunes into a single checkpoint.

3. Our Method

In this section, we introduce our training algorithm, *Merge-Adversarial Training* (MAT), specifically designed to make OSM watermarks more durable against merging (Sec. 3.1). In Sec. 3.2, we additionally detail all realistic merge scenarios considered for evaluation.

3.1. Merge-Adversarial Training

Threat model We consider a model-provider that releases an OSM watermarked model. Downstream users can then finetune this model into multiple experts and subsequently merge them freely with one another, additional finetunes, the original base, or community checkpoints. Importantly, once the watermarked model is released, we assume we have no control over how users use it.

Algorithm 1 Merge-Adversarial Training

Require: Frozen base θ_0; watermark transform $w_k(\cdot)$ with top-k gating; dataset $\mathcal{D}$; merge-weight range $[\alpha_{\min}, \alpha_{\max}]$; learning rate η; training steps T

1: $\theta \leftarrow \theta_0$ {Initialize trainable copy}
2: **for** s from 0 to $T-1$ **do**
3: $\quad x \leftarrow \text{Sample}(\mathcal{D})$ {Batch of sequences}
4: $\quad \tilde{p}(\cdot \mid x_{<t}) \leftarrow w_k(p_{\theta_0}(\cdot \mid x_{<t}))$
5: $\quad \alpha \leftarrow \text{Sample}(\mathcal{U}(\alpha_{\min}, \alpha_{\max}))$ {Merge weight}
6: $\quad \theta_M \leftarrow \alpha\,\theta + (1-\alpha)\,\theta_0$ {Linear merge}
7: $\quad \mathcal{L} \leftarrow \sum_{t=1}^{|x|} \mathrm{KL}\big(\tilde{p}(\cdot \mid x_{<t}) \,\big\|\, p_{\theta_M}(\cdot \mid x_{<t})\big)$
8: $\quad g \leftarrow \nabla_\theta \mathcal{L}$
9: $\quad \theta \leftarrow \theta - \eta g$
10: **end for**
11: **return** θ

Training setting For training, we assume that we have access to an unwatermarked base model. Our goal is to distill a generation-time watermark into this model as in (Gu et al., 2024). To embed a watermark that is durable against model merging, we adapt standard watermark-distillation using an adversarial training approach. In particular, we consider model merging as an adversarial action, and regularize our model to be robust against it. In order to avoid any assumptions on later downstream merging behavior, we instantiate our adversarial merge using the unwatermarked base model. We show in Sec. 4.2 that this is, in most cases, a worst-case proxy for watermark durability.

Merge-adversarial training (MAT) We present an overview of our method in Algorithm 1. Let θ_0 denote the frozen base model and θ the trainable copy initialized from θ_0 (line 1). We write θ_M for a merged model formed by linearly combining θ with θ_0. Lastly, $w_k(\cdot)$ corresponds to the generation-time watermarking algorithm, applied to the top-k logits. It maps the next-token probability distribution to its watermarked counterpart. At each step we sample a merge weight $\alpha \sim \mathcal{U}(\alpha_{\min}, \alpha_{\max})$ and form the merged parameters $\theta_M = \alpha\,\theta + (1-\alpha)\,\theta_0$ (lines 5-6). The merged model is trained to mirror the watermarked teacher's next-token distribution using KL divergence (line 7):

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{D}} \sum_{t=1}^{|x|} \mathrm{KL}\big(w_k(p_{\theta_0}(\cdot \mid x_{<t})) \,\big\|\, p_{\theta_M}(\cdot \mid x_{<t})\big)\,, \tag{1}$$

where $p_{\theta_0}(\cdot \mid x_{<t})$ and $p_{\theta_M}(\cdot \mid x_{<t})$ correspond to the frozen base θ_0 and the merged θ_M, respectively. We then backpropagate this loss back to the student model θ (lines 8-9) to complete our training loop.

Watermark-algorithm w_k Like standard watermark-distillation, our method is agnostic to the underlying generation-time algorithm. In Sec. 4, we evaluate it extensively on KGW, with additional experiments on the AAR and KTH families. Unlike Gu et al. (2024), however, we find in App. B.1 that watermark distillation should be performed only on the top-k logits to improve model performance.

Choice of merge adversary Even though we only explicitely train against linear interpolation with θ_0, we find in Sec. 4 that the resulting watermark survives non-linear merges as well, suggesting that robustness learned against the simpler merges consistently transfers to more complex methods. We ablate the merge-weight range $[\alpha_{\min}, \alpha_{\max}]$, the top-$k$ gating, and the watermark strength δ in App. B, and report the full training hyperparameters in App. A.1.

3.2. Merge Scenarios and Evaluation Pipeline

Next, we detail all merge scenarios we consider for evaluation, also illustrated in Figure 2. We first introduce notation for the parent models involved in each merge, then list the configurations themselves, and lastly describe the resulting evaluation pipeline.

Following Sec. 3.1, let B (the watermarked base) correspond to the trained and watermarked θ released by the provider. Let B' be the unwatermarked base θ_0. We further use F for a finetune of B, and F' for an unwatermarked finetune (from the provider).

We group the merges into two families, each corresponding to a real-world deployment scenario. The *watermarked* family (Figure 2 right) covers four configurations in which every parent is derived from B: (i) the two-expert merge FF (German FT $\otimes$ Math FT), corresponding to a user who has trained multiple domain experts and combines them into a single multi-skilled model; (ii) the base-expert merge BF ($B \otimes$ Math FT), the standard recipe for mitigating catastrophic forgetting on the FT-target domain by merging a finetune back with the watermarked base; (iii) the multi-stage cascades F^2BF and FBF, simulating more complex post-training pipelines such as the recursive expert-and-base merges (e.g., used to build BGGPT (Alexandrov et al., 2024)), where intermediate FF or BF checkpoints are themselves merged again with a third parent. We additionally consider F^2F (German FT $\otimes$ Math FT $\otimes$ French FT), as an example of a multi-stage merge of individual finetunes (Sec. 4.1). For every cascaded merge, the inner merge is fixed at $\alpha = 0.5$ (e.g., in F^2BF both the inner FF and the inner BF are constructed at $\alpha = 0.5$) while we sweep the outermost $\alpha \in [0.1, 0.9]$ using 0.1 steps.

The *unwatermarked* family (Figure 2 left) covers the case in which the watermarked base B is merged with a parent that does not carry the watermark: BF' ($B \otimes$ Private Instruct Model Finetune) and BB' ($B\otimes$ Private Instruct Model). While not reflective of typical deployment scenarios, we find that BB' serves as an empirical lower bound on detection for the harder all-watermarked cascades at matched effective dose (Sec. 4.2), making it a cheap single-merge proxy for evaluating these configurations. We note that MAT only optimizes against the linear BB' during training (Sec. 3.1), yet still consistently improves durability across all merge scenarios, including the non-linear ones.

4. Experimental Evaluation

In this section, we show that MAT preserves the performance of standard watermark distillation while being significantly more durable against realistic merges (Sec. 4.1). Following prior work, we also evaluate durability against unwatermarked-parent merges (Sec. 4.2) and find that they serve as an effective worst-case proxy for merge-durability evaluation. In Sec. 4.3, we ablate across watermarking algorithms, model architectures, and other OSM watermarks. We defer additional results to App. A.

Experimental setup We use LLAMA-3.1-8B-INSTRUCT, deferring additional results on QWEN-2.5-3B-INSTRUCT to Sec. 4.3. We compare MAT against standard watermark distillation (KGW-D) using the same hyperparameters ($\delta = 2.3$, $\gamma = 0.25$). Each base model is distilled on a multi-domain mixture spanning English (ALPACA-GPT4 (Peng et al., 2023), OPENWEB-TEXT (Gokaslan & Cohen, 2019)), German (ALPACA-GPT4-DE (LeoLM, 2023), FINEWEB-2 (Penedo et al., 2024)), math (METAMATHQA (Yu et al., 2024a)), and code (CODEALPACA (Chaudhary, 2023)), ensuring watermark learning across domains (Gloaguen et al., 2026). We then derive finetuned versions of each watermarked base using domain-specific datasets: NUMINAMATH-COT (Li et al., 2024) for math, EVOL-INSTRUCT-DE (Chen et al., 2023) for German, and a 58k mix of FRENCH-ALPACA (Pacifico, 2024) with the Lucie corpus (Gouvert et al., 2025) for French.

To measure watermark detectability, we report true positive rate at 1% false positive rate (TPR@1%) averaged over 500 English prompts from C4 (Raffel et al., 2023). Additionally, we measure TPR@1% on German prompts from FineWeb-2 `deu_Latn` (Penedo et al., 2024) and math prompts from GSM8K (Cobbe et al., 2021), ensuring the watermark persists in finetuned-model domains. For quality, we report perplexity under LLAMA-2-13B and `seq-rep-3`, i.e., the fraction of repeated 3-grams in the continuation. Lower perplexity reflects more natural generations under an external reference judge, while `seq-rep-3` captures degenerate repetition that perplexity alone can mask. We further benchmark downstream capabilities with the evaluation harness

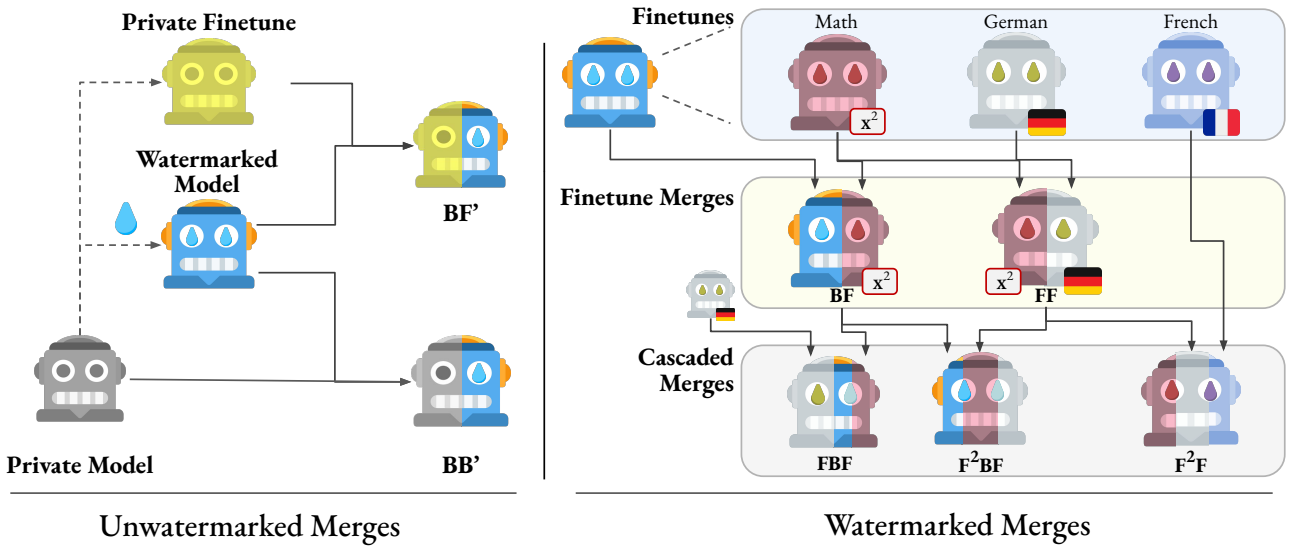

Figure 2. **Merge evaluation pipeline:** The provider releases a watermarked Instruct base (B), and produces watermarked finetunes (F) via SFT. Downstream users merge any combination of these models, yielding the scenarios FF, BF, F^2F, FBF, F^2BF, BB', and BF' that we evaluate.

from Gao et al. (2024) on ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), GSM8K (Cobbe et al., 2021), and MATH (Lewkowycz et al., 2022). We defer the full training hyperparameters and data compositions to App. A.1.

Merge scenarios and evaluation We follow the threat model from Sec. 3.1: we act as the provider of a watermarked base B that users subsequently merge. We organize results into two categories. First, *all-watermarked* merges (Sec. 4.1), where every parent derives from our watermarked release; here, our method consistently improves merge robustness over the KGW-D baseline. Second, we test *unwatermarked-parent* merges (Sec. 4.2), where B is merged with either an external community finetune or the original unwatermarked Instruct base. In particular, we compare BB' directly against cascaded FBF results at matched effective α, and additionally evaluate BF' as a real-world deployment scenario where the watermarked base meets an externally trained community finetune.

4.1. Durability Against Realistic Merge Scenarios

Here, we evaluate watermark durability under a range of realistic scenarios (illustrated in Figure 2). We first consider two merge configurations widely used in practice: combining two domain-expert finetunes (FF) and merging the watermarked base with a single domain finetune to mitigate catastrophic forgetting (BF). We then evaluate three cascaded merges, which stress-test our watermark durability against more complex post-training pipelines.

Table 1. **Watermark evaluation pre-merge:** TPR@1% and generation quality (PPL; lower is better) of the generation-time KGW (KGW), watermark-distillation (KGW-D), and ours.

	TPR@1%			PPL		
Variant	EN	DE	Math	EN	DE	FR
KGW	100.0	100.0	100.0	5.65	5.68	5.59
KGW-D	98.7	98.8	95.8	5.39	5.66	5.93
Ours	98.0	99.0	96.4	5.69	5.59	5.46

Watermark detectability-quality trade-off prior to merging To ensure a fair comparison, in Table 1, we verify that, *prior to merging*, our method has a similar detectability-quality trade-off to KGW-D. This means that our method's improvement in durability against merging is not induced by a stronger watermark. It also shows that our method does not hurt model performance prior to merging compared to KGW-D, both when measuring perplexity (Table 1) and benchmark accuracies (App. A.2.3).

Our method is durable against finetune merges To measure durability against finetune merges (Figure 2, second row), we compare the merged-model TPR@1% to the highest parent TPR@1% on a given domain (called the *single-finetuning baseline*). Intuitively, a non-durable watermark should yield merged-model TPR@1% no higher than this baseline. We also measure watermark detectability on the *expert domains* (i.e., German for the German finetune and math for the math finetune), because, as we show in

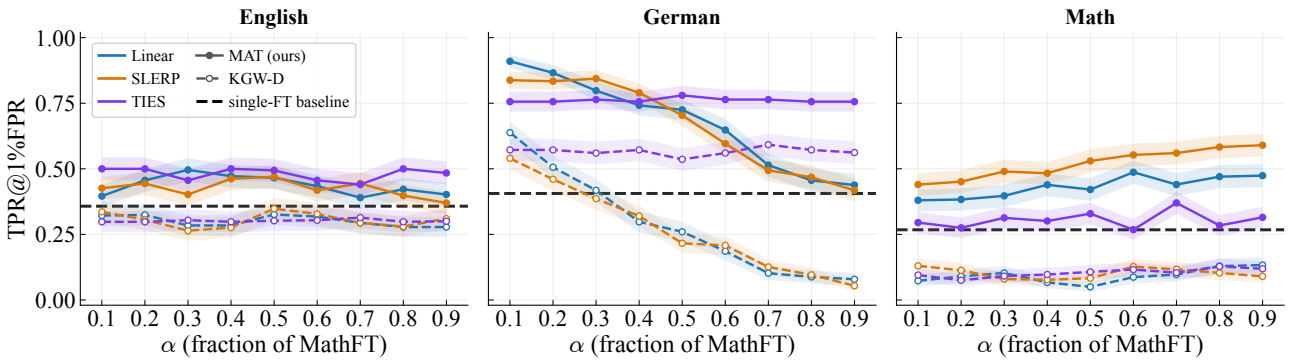

Figure 3. **Watermark detectability for different merge ratios of experts (FF):** We compare watermark detectability (TPR@1%) when merging an α math-finetuned model with a $(1-\alpha)$ German-finetuned model using three different merging algorithms. We average results over 500 samples with English, German, and math prompts, respectively. The dashed line corresponds to the single-finetuning baseline, i.e., the maximum TPR@1% of the finetuned models for the given domain.

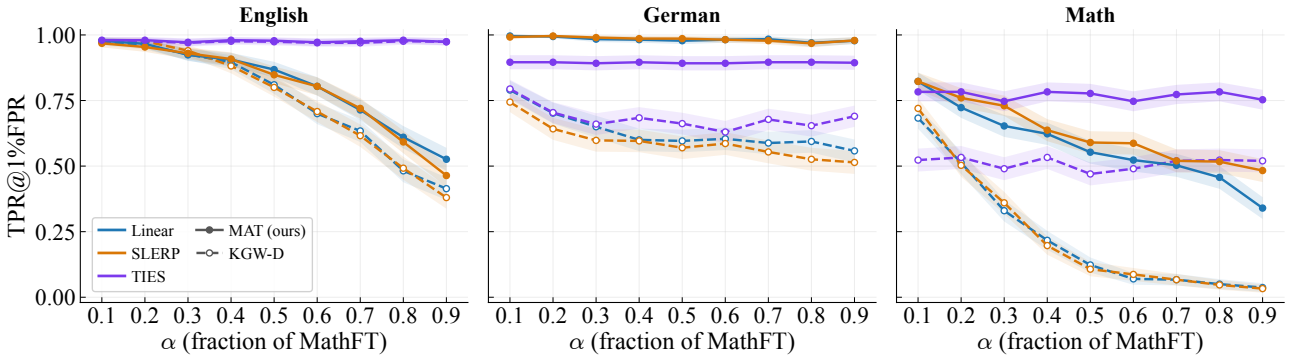

Figure 4. **Watermark detectability for different merge ratios against catastrophic forgetting (BF):** We compare watermark detectability (TPR@1%) when merging an α-weighted math-finetuned model with a $(1-\alpha)$-weighted base (watermarked) model using three different merging algorithms. We average results over 500 samples with English, German, and math prompts, respectively.

Table 5, these are the domains where the watermark of the finetuned experts is the weakest (e.g., on average, TPR@1% on the expert domain drops from $\sim$98% pre-finetuning to $\sim$15–25% post-finetuning, while outside this domain detection only drops to $\sim$28–40%).

In Figure 3, we compare our method with KGW-D in the expert-knowledge combination (FF) scenario. Across English and the expert domains (German and math), our method systematically outperforms KGW-D. More importantly, unlike KGW-D, with our method the merged model always has a higher TPR than the single-finetuning baseline (dashed line in Figure 3), which means that our method effectively improves durability against merging by successfully recovering the watermark degradation caused by finetuning. At the same time, we find in App. A.3.5 that this improvement to durability does not prevent the merged model from successfully combining the expert knowledge.

For the BF merge, corresponding to anti-catastrophic forgetting merges, where one parent is the watermarked base B itself, the watermark is, by construction, better preserved than under FF, since one half of the merged weights still encodes the full distillation signal. Nonetheless, Figure 4 shows that our method systematically outperforms KGW-D, especially in the expert domain where finetuning has weakened the watermark. This confirms that our method is robust to finetune merges.

Our method remains durable against cascaded merges We next evaluate whether durability against finetuned merges extends to cascaded merges (Figure 2, last row). Specifically, we consider an F^2F configuration: the FF merge above (math finetune merged with German finetune) is merged again with a French finetune. Figure 5 shows that MAT remains durable, significantly outperforming KGW-D across all evaluated domains and exceeding the single-finetuning baseline. Full per-method results for the remaining cascaded merges (FBF, F^2BF) are in App. A.3.

Overall, our results show that our method is durable against all tested merging algorithms and realistic scenarios, unlike prior work. Importantly, this comes at no cost to model performance before merging.

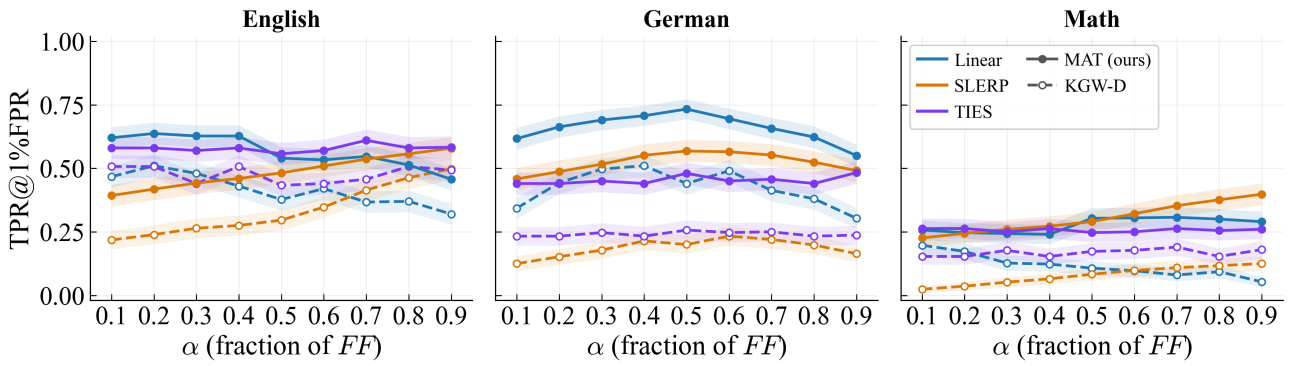

Figure 5. **Watermark detectability with cascaded merges:** We compare watermark detectability (TPR@1%) when merging an α-weighted FF-merged model with a $(1-\alpha)$-weighted French finetuned model using SLERP merging. We average results over 500 samples with English, German, and math prompts, respectively. The dashed line corresponds to the single-finetuning baseline, i.e., the maximum TPR@1% of the FF-merged and finetuned models for the given domain.

4.2. Durability Against Unwatermarked-Parent Merges

Following prior work (Gloaguen et al., 2025), we consider merge scenarios where one parent is unwatermarked. We verify that these merges are effective proxies for realistic merge scenarios, enabling efficient evaluation of OSM watermark durability under model merging. App. A.3.4 further shows that our watermark is durable across unwatermarked-parent scenarios, including BB' merges with the unwatermarked base and BF' merges with finetunes such as FuseChat (Wan et al., 2024), OpenMath (Toshniwal et al., 2024), or Tulu (Lambert et al., 2024).

Unwatermarked-parent merges as a proxy for realistic merges We compare durability against BB' with durability against cascaded merges (FBF). For cascaded merges, let α_{eff} denote the final weight fraction from the watermarked model; e.g., two successive merges at $\alpha = 0.5$ yield $\alpha_{\text{eff}} :- 0.25$. We hypothesize that cascaded-merge durability, measured by TPR@1%, is greater than that of BB' at matched α_{eff}. If so, BB' provides a lower bound for watermark durability under merging and, because it requires no finetuning, serves as an efficient proxy.

In Figure 6, we compare the watermark detectability of FBF with that of BB' at matched α_{eff}. We find that in the overwhelming majority of settings (93% of tested configurations), the TPR@1% of BB' is indeed lower than that of FBF, validating our hypothesis. The lower bound is violated only in the math domain at high α_{eff} (i.e., with a high portion of the math finetuned model). We suspect this is because math finetuning significantly lowered the entropy of the resulting model on the math domain, making it harder to watermark than even the base (unwatermarked) model. As a result, a high fraction of the math finetuned model degrades the watermark faster than a similar fraction of the base (unwatermarked) model. Nonetheless, BB' remains in most cases an effective proxy for evaluating durability against merging. Full results and the per-α_{eff} breakdown are in App. A.3.2.

4.3. Ablations on Watermark Families and Model Architectures

Next, we test whether our gains are specific to KGW-D on LLAMA-3.1-8B-INSTRUCT along two distinct axes: watermark family and base architecture. In particular, we apply the MAT objective to two additional watermark-distillation variants, AAR and KTH, and we also transfer it to a different base model, QWEN-2.5-3B-INSTRUCT. Additionally, we evaluate the weight-space GAUSSMARK (Block et al., 2025) on merge robustness in order to verify that merge-induced degradation is not specific to distillation-based OSM watermarking approaches.

MAT generalizes across watermark families We evaluate durability against BB' across α for AAR and KTH, implemented as in (Gu et al., 2024), on LLAMA-3.1-8B-INSTRUCT. As shown in Figure 7, baseline distillation collapses under merging at low α for both schemes (TPR@1% near zero on Linear and SLERP), while our method preserves detectability consistently across the full α range. Notably, this mirrors the recovery we observe for KGW-D, indicating that the merge-adversary objective is not specific to KGW but also transfers to other watermark families. We provide the full results in App. A.4.

MAT transfers to different base architectures We further apply our method to QWEN-2.5-3B-INSTRUCT, reusing the same hyperparameters as in the LLAMA-3.1-8B-INSTRUCT run. As we show in Figure 7 (bottom-left), despite this direct transfer, our method maintains a consistent improvement over KGW-D across all α's, showing even larger gains for higher α's where the watermarked

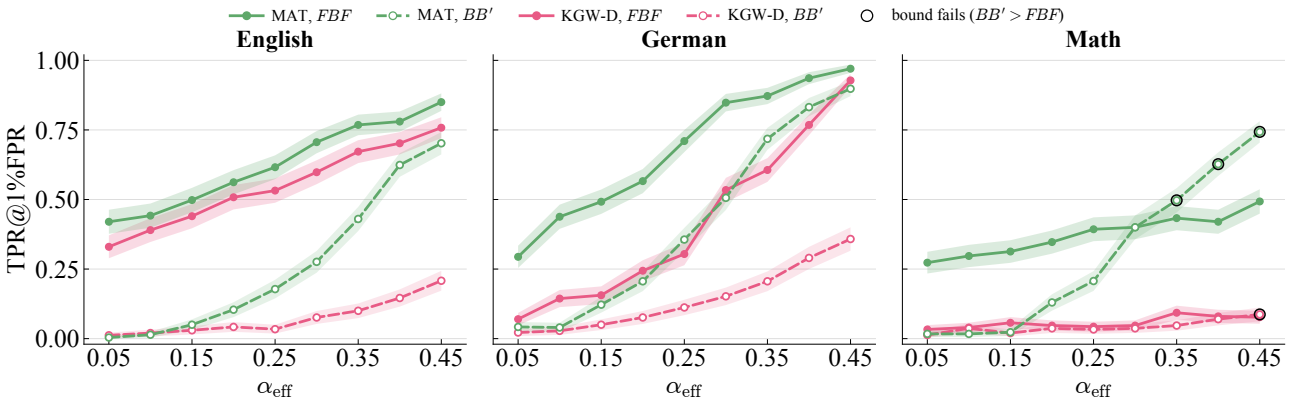

Figure 6. **Watermark detectability for BB' and FBF with effective merge ratios:** We compare watermark detectability (TPR@1%) between the unwatermarked merge BB' and FBF at matched effective merge ratio α_{eff}. We highlight points where BB' has higher TPR@1% than FBF.

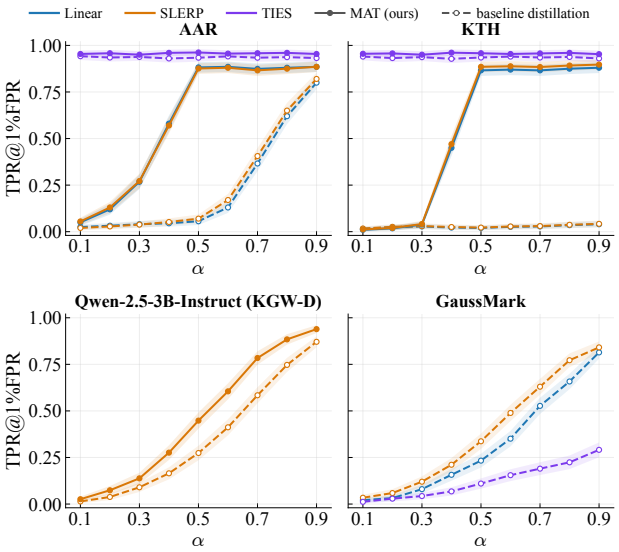

Figure 7. **Durability gain across watermark families and architectures:** Post-merge TPR@1% FPR under BB' on English (C4). *Top:* AAR (left) and KTH (right) on LLAMA-3.1-8B-INSTRUCT. *Bottom:* KGW-D on QWEN-2.5-3B-INSTRUCT (left, SLERP only) and GAUSSMARK on LLAMA-3.1-8B-INSTRUCT (right, baseline only)

base model B dominates. These results indicate that MAT can already provide benefits across base architectures even without requiring model-specific tuning of hyperparameters.

Other watermark schemes also collapse under merging Beyond watermark-distillation, we also evaluate GAUSSMARK, which directly embeds the watermark via key-dependent perturbations of the model weights. The bottom-right panel of Figure 7 shows that GAUSSMARK (like KGW-D) collapses under merging, dropping from $\sim$89% TPR@1% standalone to $\sim$34% at $\alpha = 0.5$ using SLERP. This suggests that the merge-induced collapse is not an artifact of any single watermarking mechanism, motivating future work on adapting MAT to weight-space watermarks.

5. Limitations and Future Work

While we provide an extensive evaluation of MAT, several directions remain open. On the merge side, we evaluate MAT across three popular weight-space algorithms (LINEAR, SLERP, TIES), showing that MAT requires only linear interpolation during training. Although this transfers well to SLERP and TIES, extending MAT to more aggressive trim-and-rescale schemes such as DARE-TIES (Yu et al., 2024b), Model Breadcrumbs (Davari & Belilovsky, 2024), or evolutionary and learned merge operators is an interesting direction for future research. Similarly, while we are the first to evaluate watermark durability in cascaded scenarios, we focus on depth two to three (e.g., F^2F and F^2BF); scaling to longer merge chains that mix more checkpoints is an important direction for real-world model use-cases.

On the watermark side, we train MAT variants for KGW-D, AAR, and KTH, and additionally report the merge-collapse behavior of GAUSSMARK. Adapting MAT to weight-space watermarks like GAUSSMARK would require backpropagating through the detector, which we leave to future work. More broadly, extending MAT to additional schemes, such as SynthID, semantic watermarks (SemStamp, SemaMark), and unbiased schemes (DiPmark), is a promising avenue for future work.

Our main experiments in Sec. 4 use a single objective formulation with stochastic uniform α and a fixed top-k logit gate. Although we ablate the chosen parameters in App. B, alternative, more complex formulations (e.g., curricula over α, worst-case-α adversarial training, multi-step inner adversaries, or merge-aware regularizers) may improve the

robustness-quality trade-off even further (at the cost of simplicity). Finally, our experiments focus on LLAMA-3.1-8B-INSTRUCT and QWEN-2.5-3B-INSTRUCT. We consider scaling MAT, as in Sec. 3, to larger models to be conceptually straightforward.

6. Conclusion

We introduced Merge-Adversarial Training (MAT), a method for improving the durability of open-source model watermarks under model merging. MAT trains the watermarked model against a simulated merge adversary, encouraging the watermark signal to remain detectable even after the released weights are interpolated with other checkpoints. Despite using only linear interpolation during training, MAT generalizes across merge algorithms, consistently improving over standard watermark distillation.

We also introduce a broader evaluation pipeline for merge durability and show that unwatermarked-base merges provide an effective worst-case proxy for faster evaluations. To our knowledge, this is the first work to demonstrate that watermark durability under merging can be explicitly optimized during training without sacrificing model quality. Together, MAT and our evaluation framework provide a practical step toward operationalizing OSM watermarks that remain detectable after common post-release modifications, and highlight that durability should be treated as a first-class requirement for open-source watermarking alongside detectability, quality, and text-level robustness.

References

Aaronson, S. Watermarking of large language models. In *Workshop on Large Language Models and Transformers, Simons Institute, UC Berkeley*, 2023.

Alexandrov, A., Raychev, V., Dimitrov, D. I., Zhang, C., Vechev, M., and Toutanova, K. Bggpt 1.0: Extending english-centric llms to other languages, 2024. URL https://arxiv.org/abs/2412.10893.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open llm leaderboard. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2023.

Block, A., Sekhari, A., and Rakhlin, A. Gaussmark: A practical approach for structural watermarking of language models, 2025. URL https://arxiv.org/abs/2501.13941.

Chaudhary, S. Code Alpaca: An instruction-following LLaMA model for code generation. https://github.com/sahil280114/codealpaca, 2023.

Chen, Z., Yan, S., Liang, J., Jiang, F., Wu, X., Yu, F., Chen, G. H., Chen, J., Zhang, H., Jianquan, L., Xiang, W., and Wang, B. MultilingualSIFT: Multilingual Supervised Instruction Fine-tuning, July 2023. URL https://github.com/FreedomIntelligence/MultilingualSIFT.git.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dathathri, S., See, A., Ghaisas, S., Huang, P.-S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035): 818–823, 2024.

Davari, M. and Belilovsky, E. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pp. 270–287. Springer, 2024.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gloaguen, T., Jovanović, N., Staab, R., and Vechev, M. Towards watermarking of open-source llms. *arXiv preprint arXiv:2502.10525*, 2025.

Gloaguen, T., Staab, R., Jovanović, N., and Vechev, M. LLM fingerprinting via semantically conditioned watermarks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=t38nZqqi3Z.

Goddard, C., Siriwardhana, S., Ehghaghi, M., Meyers, L., Karpukhin, V., Benedict, B., McQuade, M., and Solawetz, J. Arcee's MergeKit: A toolkit for merging large language models. In Dernoncourt, F., Preoţiuc-Pietro, D., and Shimorina, A. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 477–485, Miami, Florida, US, November 2024. Association

for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.36. URL https://aclanthology.org/2024.emnlp-industry.36.

Gokaslan, A. and Cohen, V. OpenWebText corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Gouvert, O., Hunter, J., Louradour, J., Cerisara, C., Dufraisse, E., Sy, Y., Rivière, L., Lorré, J.-P., and community, O.-F. The lucie-7b llm and the lucie training dataset: Open resources for multilingual language generation, 2025. URL https://arxiv.org/abs/2503.12294.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, C., Li, X. L., Liang, P., and Hashimoto, T. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=9k0krNzvlV.

Hamon, R., Sanchez, I., Fernández Llorca, D., and Gómez, E. Generative ai transparency: Identification of machine-generated content. Scientific analysis or review, Ispra (Italy), 2024.

He, Z., Zhou, B., Hao, H., Liu, A., Wang, X., Tu, Z., Zhang, Z., and Wang, R. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4115–4129, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Jovanović, N., Staab, R., and Vechev, M. Watermark stealing in large language models. In *International Conference on Machine Learning*, pp. 22570–22593. PMLR, 2024.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023a.

Kirchenbauer, J., Geiping, J., Wen, Y., Shu, M., Saifullah, K., Kong, K., Fernando, K., Saha, A., Goldblum, M., and Goldstein, T. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023b.

Kuditipudi, R., Thickstun, J., Hashimoto, T., and Liang, P. Robust distortion-free watermarks for language models. *TMLR*, 2024.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tülu 3: Pushing frontiers in open language model post-training. 2024.

LeoLM. Alpaca-GPT4-DE: German translation of the Alpaca-GPT4 instruction dataset. https://huggingface.co/datasets/LeoLM/alpaca_gpt4_de, 2023.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S. C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., and Polu, S. Numinamath. https://huggingface.co/AI-MO/NuminaMath-CoT, 2024.

Pacifico, J. French-alpaca: A French instruction-following dataset (110k). https://huggingface.co/datasets/jpacifico/French-Alpaca-dataset-Instruct-110K, 2024.

Pang, Q., Hu, S., Zheng, W., and Smith, V. No free lunch in llm watermarking: Trade-offs in watermarking design choices. *Advances in Neural Information Processing Systems*, 37:138756–138788, 2024.

Penedo, G., Kydlíček, H., Sabolcěc, V., Messmer, B., Foroutan, N., Jaggi, M., von Werra, L., and Wolf, T. FineWeb-2: A sparkling update with 1000s of languages. https://huggingface.co/datasets/HuggingFaceFW/fineweb-2, 2024.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with GPT-4, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., and Feizi, S. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Team, K., Bai, T., Bai, Y., Bao, Y., et al. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Team, Q. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Toshniwal, S., Du, W., Moshkov, I., Kisacanin, B., Ayrapetyan, A., and Gitman, I. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Wan, F., Zhong, L., Yang, Z., Chen, R., and Quan, X. Fusechat: Knowledge fusion of chat models, 2024. URL https://arxiv.org/abs/2408.07990.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. URL https://arxiv.org/abs/2203.05482.

Xue, J., Zhao, Y., Ghanim, M. A., Gao, S., Sun, R., Lou, Q., and Zheng, M. Pro: Enabling precise and robust text watermark for open-source llms. *arXiv preprint arXiv:2510.23891*, 2025.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. MetaMath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024a. URL https://openreview.net/forum?id=N8N0hgNDRt.

Yu, L., Yu, B., Yu, H., Huang, F., and Li, Y. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024b.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

A. Supplementary Results

A.1. Watermark Distillation and SFT Hyperparameters

A.1.1. Data Mixture

Watermark distillation uses a multi-domain mix: 30% math (MetaMathQA (Yu et al., 2024a)), 30% German (20% Alpaca-GPT4-DE (LeoLM, 2023) + 10% FineWeb-2 (Penedo et al., 2024) `deu_Latn`), 30% English (20% Alpaca-GPT4 (Peng et al., 2023) + 10% OpenWebText (Gokaslan & Cohen, 2019)), and 10% code (CodeAlpaca (Chaudhary, 2023)). Datasets are streamed and interleaved with sampling probabilities equal to the percentages above (fixed seed 42), and each sample is truncated/packed to a sequence length of 512 tokens.

A.1.2. Watermark Distillation

Both KGW-D (replication of (Gu et al., 2024)) and MAT KGW-D are trained from Llama-3.1-8B-Instruct (Grattafiori et al., 2024) in bfloat16, using AdamW (β_1=0.9, β_2=0.999, ε=10^{-8}), cosine LR schedule with 500 warm-up steps and peak LR 1×10^{-5}, a batch size of 64, and gradient checkpointing enabled. We use the same KGW parameters in both runs (γ=0.25, k=1). The bias is δ=2.3 for KGW-D and MAT KGW-D, with the latter additionally using the rank-top-k restriction (k=100). The merge adversary samples $\alpha \sim \mathcal{U}(0.1, 1.0)$ per step and is disabled for the KGW-D baseline. Both KGW-D and MAT KGW-D are trained for 5,000 steps. We run the experiments on NVIDIA RTX Blackwell 6000 Pro GPUs with 96GB VRAM. KGW-D uses 1×4 GPUs under FSDP (full shard, $\sim$ 4 h training time) and MAT KGW-D uses 2×4 GPUs under DeepSpeed ZeRO-2 ($\sim$ 6 h training time). The teacher copy is held on CPU and moved to GPU on demand to fit the merge-adversary forward pass within GPU memory.

A.1.3. Per-Domain SFT

Each domain SFT starts from the KGW-D/MAT KGW-D checkpoint-5,000 and uses AdamW with weight decay 0.01, cosine schedule and `bfloat16` with gradient checkpointing. Hyperparameters per domain are reported in Table 2.

Table 2. **Per-domain SFT hyperparameters:** Each finetune starts from the corresponding KGW-D/MAT KGW-D checkpoint-5,000.

Domain	Dataset	Epochs	LR	Eff. batch	Seq. len
Math	NuminaMath-CoT (Li et al., 2024)	1	1×10^{-5}	128	4096
German	Evol-Instruct-DE (Chen et al., 2023)	2	2×10^{-5}	64	2048
French	Lucie + French-Alpaca (58k) (Gouvert et al., 2025; Pacifico, 2024)	2	2×10^{-5}	64	2048

The French SFT mixes 29k French-Alpaca instructions sampled from (Pacifico, 2024) with 29k continued-pretraining pseudo-pairs sampled from the `RedPajama-fr` subset of the Lucie Training Dataset (Gouvert et al., 2025) for a total of 58k examples, chosen for sample-count parity with the German Evol-Instruct-DE set so both fine-tunes run for an equal number of optimizer steps. All other settings follow the HuggingFace `Trainer` defaults.

A.2. Benchmark Results and Downstream Capabilities

A.2.1. Downstream Capabilities

We evaluate downstream capabilities with the EleutherAI `lm-evaluation-harness` (Gao et al., 2024) using each benchmark's standard few-shot regime: for English/general reasoning we run MMLU (Hendrycks et al., 2021) (5-shot), HellaSwag (Zellers et al., 2019) (10-shot), ARC-Challenge (Clark et al., 2018) (25-shot). For math we run GSM8K (Cobbe et al., 2021) (8-shot) and Minerva-MATH (MATH) (Lewkowycz et al., 2022) (0-shot). Because two of our merge-parent fine-tunes target non-English domains (German and French), we additionally evaluate on MMLU-DE (Hendrycks et al., 2021), HellaSwag-DE (Zellers et al., 2019) and the analogous `fra_Latn` suite.

A.2.2. Generation Quality

We measure generation quality using the reference-model perplexity protocol established by Gu et al. (2024) and used in Gloaguen et al. (2025): from the watermarked model, we generate, unless otherwise mentioned, $n = 500$ continuations conditioned on natural-text prefixes drawn from a domain-matched corpus (C4-RealNewsLike (Raffel et al., 2023) for English, FineWeb-2 `deu_Latn` (Penedo et al., 2024)/`fra_Latn` (Penedo et al., 2024) for German/French, and GSM8K

(Cobbe et al., 2021) prompts for math). We compute their perplexity with LLAMA-2-13B (Touvron et al., 2023) and report the median PPL. Alongside reference perplexity, we report *seq-rep-3*, the fraction of repeated trigrams in the generated continuation, which captures the degenerate repetition failure mode that low perplexity alone can mask.

A.2.3. BASELINE MODEL BENCHMARKS

Table 3 (BASES block) reports the downstream performance of the base LLAMA-3.1-8B-INSTRUCT model and both watermarked variants (KGW-D and MAT KGW-D) prior to any finetuning or merging. Watermark distillation incurs limited quality loss: reference perplexity rises by $\sim$0.9–1.9 PPL and MMLU, HellaSwag, and MATH scores remain within $\sim$2 pp of the unwatermarked baseline, confirming that the watermark embedding itself does not substantially degrade model utility.

A.2.4. FINETUNE BENCHMARKS

The MATH FT, GERMAN FT, and FRENCH FT blocks of Table 3 compare the per-language finetunes derived from each watermarked variant. All three finetunes achieve comparable on-domain quality regardless of whether the parent model was trained with KGW-D or MAT: German finetunes reach similar perplexity on held-out German text, math finetunes show nearly equivalent MATH accuracy, and French finetunes match on French perplexity and MMLU-FR.

Table 3. **Per-task downstream accuracy across the unwatermarked LLAMA-3.1-8B-INSTRUCT base, the two watermarked bases (KGW-D, MAT KGW-D), and the Math, German, and French finetunes derived from each watermarked base, no merging:** Cells colored relative to the Instruct baseline: $\geq$95%, $\geq$90%, $<$90%. Per-language perplexity and seq-rep-3 are reported separately in Table 4.

		Accuracy [%]						
Model	Scenario	ARC	MMLU	HeSw	GSM8K	MATH	MMLU_DE	MMLU_FR
BASES	Instruct	61.9	68.8	76.7	85.0	44.0	57.5	59.3
	KGW-D	58.5	67.5	74.7	82.5	42.0	56.1	58.5
	MAT KGW-D	59.3	67.6	75.8	81.9	42.3	56.6	58.9
MATH FT	KGW-D	53.8	63.2	71.3	81.9	47.1	52.3	54.1
	MAT KGW-D	52.7	62.1	70.5	82.4	48.3	51.2	55.4
GERMAN FT	KGW-D	56.6	63.1	70.4	76.8	36.8	58.3	53.4
	MAT KGW-D	56.0	62.3	70.2	75.9	37.4	57.9	54.0
FRENCH FT	KGW-D	52.6	60.9	71.1	74.2	37.7	43.1	58.9
	MAT KGW-D	51.5	54.4	70.4	75.4	36.0	42.1	58.7

Table 4. **Per-language generation quality (no merging):** Reference perplexity (PPL, computed under LLAMA-2-13B) and seq-rep-3 of LLAMA-3.1-8B-INSTRUCT under each watermarking variant. Per-task downstream accuracy is reported separately in Table 3.

		PPL			seq-rep-3		
Model	Scenario	EN	DE	FR	EN	DE	FR
BASES	Instruct	3.81	3.98	4.55	0.058	0.093	0.077
	KGW-D	5.39	5.66	5.93	0.049	0.095	0.076
	MAT KGW-D	5.69	5.59	5.46	0.058	0.094	0.089
MATH FT	KGW-D	5.41	5.71	5.87	0.051	0.097	0.081
	MAT KGW-D	5.70	5.73	5.40	0.059	0.096	0.090
GERMAN FT	KGW-D	5.40	3.53	5.89	0.061	0.078	0.087
	MAT KGW-D	5.68	3.65	5.42	0.054	0.080	0.092
FRENCH FT	KGW-D	5.43	5.69	4.09	0.048	0.091	0.072
	MAT KGW-D	5.71	5.71	4.14	0.057	0.096	0.070

Table 5. **Standalone single-model TPR@1% FPR detection (no merging):** Bases are evaluated before any finetuning; the finetune blocks are evaluated post-SFT. LOGIT PROC. KGW is the decoding-time KGW reference.

Scenario	Method	English	German	Math
BASES	Logit proc. KGW	100.0	100.0	100.0
	KGW-D	98.7	98.8	95.8
	MAT KGW-D (ours)	98.0	99.0	96.4
GERMAN FT	KGW-D	35.4	14.0	25.0
	MAT KGW-D	33.8	17.1	28.5
MATH FT	KGW-D	34.9	41.4	21.4
	MAT KGW-D	36.1	39.8	24.4
FRENCH FT (UNSEEN)	KGW-D	52.0	42.2	21.0
	MAT KGW-D	51.4	53.8	27.0

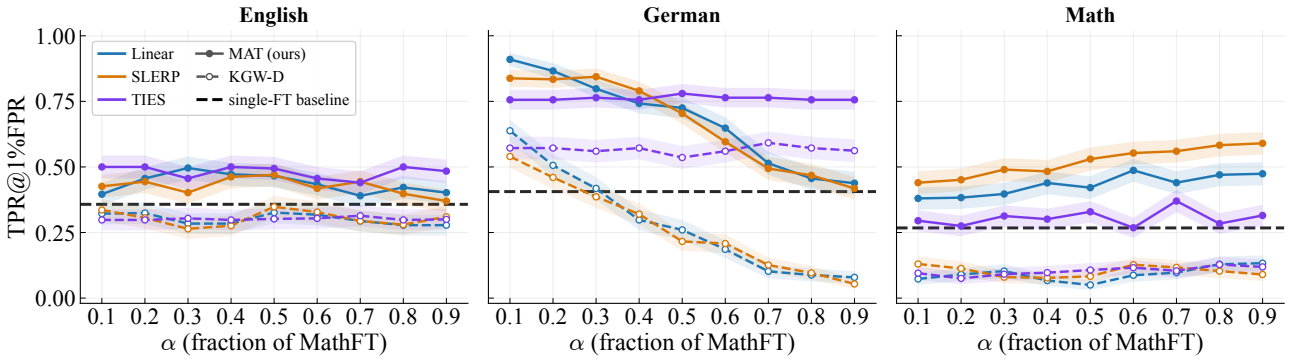

Figure 8. FF **watermark detection:** TPR@1% as a function of α = fraction of Math FT across Linear, SLERP, and TIES.

A.3. Supplementary Results for All-Watermarked Merges

Figure 8, Figure 9, Figure 10, and Figure 11 provide the full TPR@1% sweeps across the all-watermarked merge family (FF, BF, F^2BF, FBF), all merge methods (Linear, SLERP, TIES), and all three detection domains (English, German, Math) as recovery curves. We report 95% confidence intervals on every TPR@1% with $n = 500$ prompts per cell. The shaded band around each curve in Figures 8–11 and the unwatermarked-parent recovery figures depicts this interval.

A.3.1. MERGE METHODS AND HYPERPARAMETERS

We evaluate three merge methods, all parameterized by a single mixing coefficient $\alpha \in [0, 1]$. LINEAR (Wortsman et al., 2022) averages the two parents in weight space: $\theta_{\text{merge}} = \alpha\,\theta_A + (1-\alpha)\,\theta_B$. SLERP (Goddard et al., 2024) interpolates the two parents on the unit hypersphere with the same coefficient, normalizing and renormalizing each parameter tensor before/after the spherical interpolation. TIES (Yadav et al., 2023) is computed via MERGEKIT (Goddard et al., 2024) with parent A acting as the task-vector base: we form a task vector $\tau_B = \theta_B - \theta_A$, retain the top-$d$ fraction of $|\tau_B|$ per parameter tensor (density $d = 0.5$ throughout), and combine as $\theta_{\text{merge}} = \theta_A + \alpha\,\widetilde{\tau}_B$, where $\widetilde{\tau}_B$ is the magnitude-trimmed and renormalised task vector. We sweep $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ for all three methods in the per-merge results in Figures 8–11.

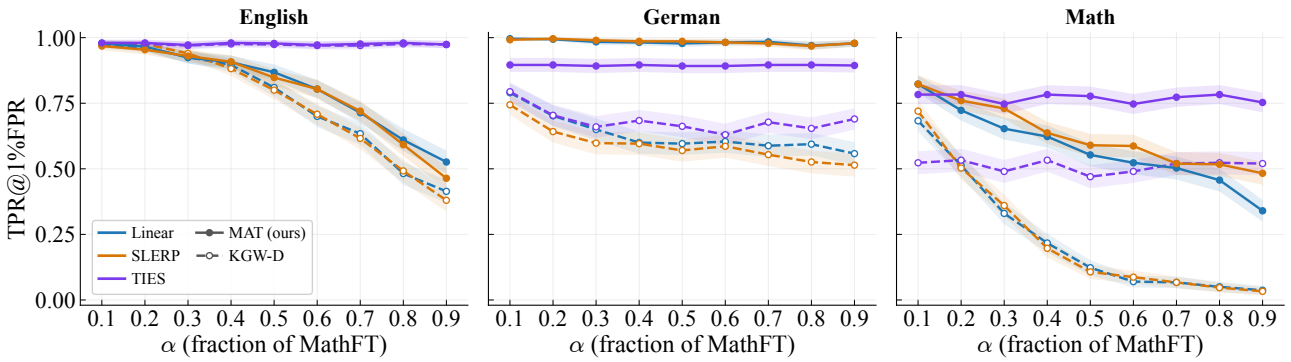

Figure 9. BF **watermark detection:** TPR@1% as a function of α = fraction of Math FT across Linear, SLERP, and TIES.

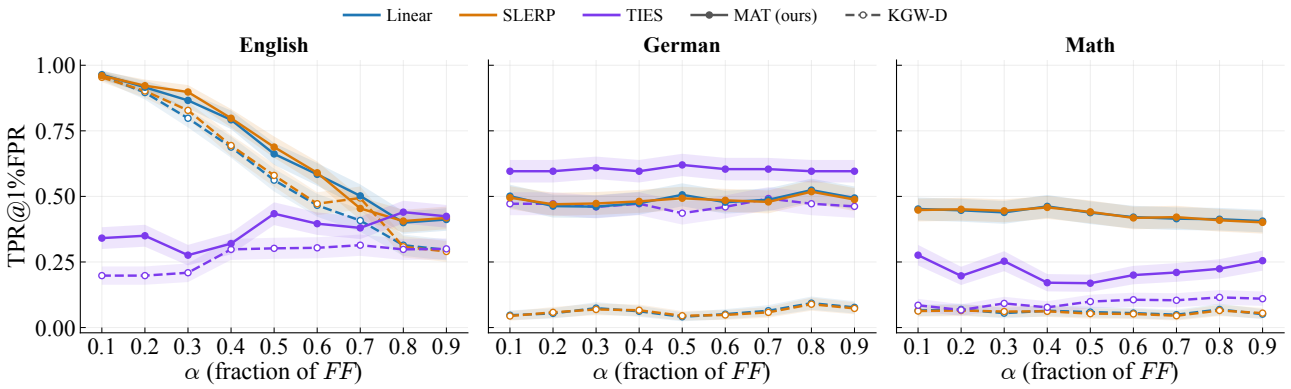

Figure 10. F^2BF **watermark detection:** TPR@1% as a function of α = fraction of FF across Linear, SLERP, and TIES.

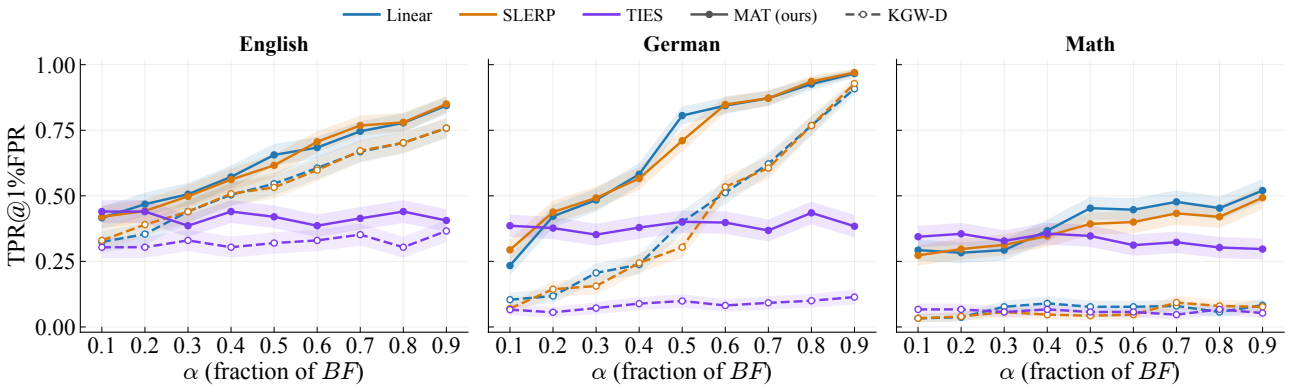

Figure 11. FBF **watermark detection:** TPR@1% as a function of α = fraction of BF across Linear, SLERP, and TIES.

A.3.2. BB' AS A LOWER BOUND FOR FBF

In Sec. 4.2, we introduce BB' as an empirical lower bound on FBF at matched effective watermark dose. Here we present the per-α sweep behind that claim (Figure 12). The matched dose is the B-fraction in the merged weights: $\alpha_{\text{eff}}(FBF) = 0.5 \cdot \alpha_{FBF}$, since BF contains $0.5\,B$. The bound holds where the FBF curve lies above the BB' curve at the same α_{eff}.

The bound holds across nearly all $(\alpha_{\text{eff}}, \text{domain})$ configurations. The few failures all fall in the Math domain at high α_{eff}: the Math FT parent on its own retains very little watermark signal, so when it dominates the FBF merge it contributes less detectable signal than BB''s simple mix of B with the unwatermarked Instruct. The same conclusion holds under Linear

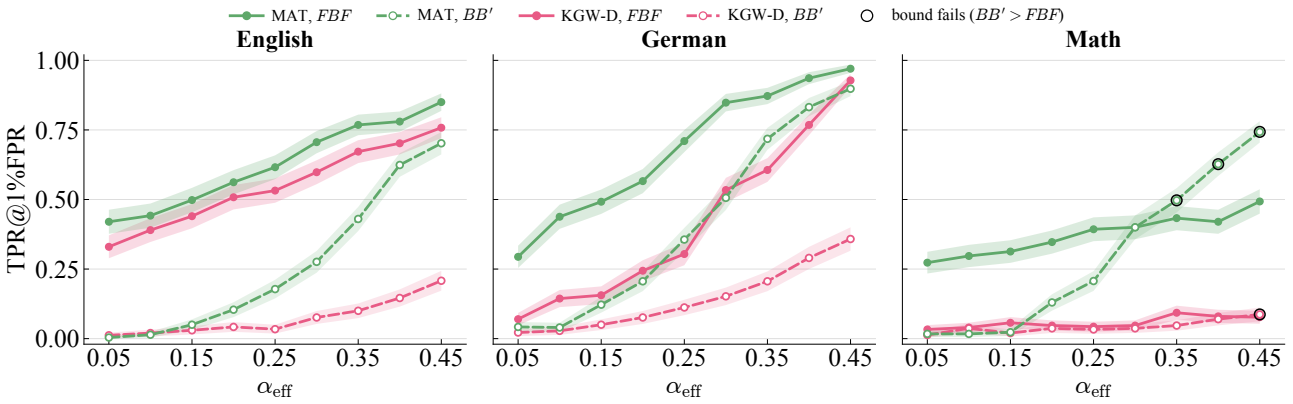

Figure 12. FBF **vs.** BB' **lower-bound test on TPR@1%, full SLERP sweep** $\alpha_{\text{eff}} \in \{0.05, 0.10, \ldots, 0.45\}$**:** The bound holds where the solid line sits above the dashed line; configurations where it fails ($BB' > FBF$) are circled.

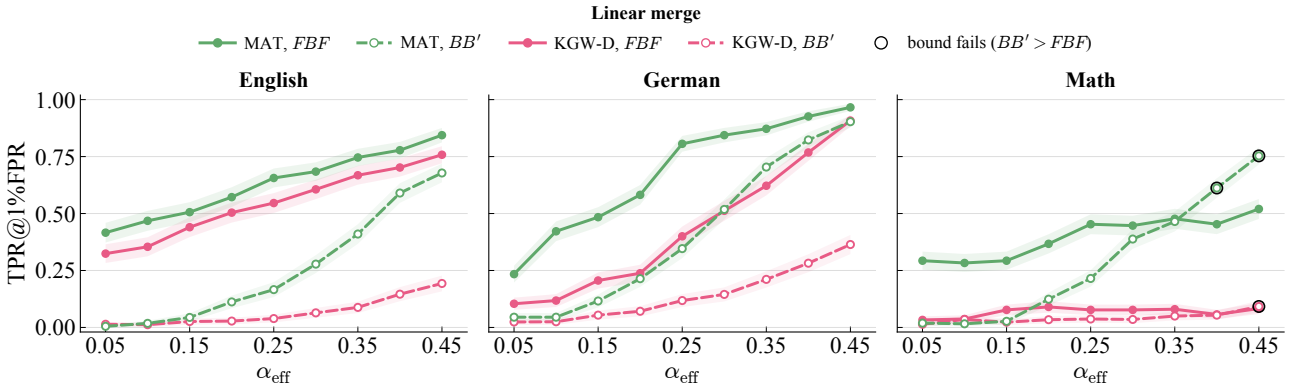

Figure 13. FBF **vs.** BB' **lower-bound test on TPR@1%, full Linear sweep** $\alpha_{\text{eff}} \in \{0.05, 0.10, \ldots, 0.45\}$**:** Same conventions as Figure 12: the bound holds where the solid line sits above the dashed line; configurations where it fails ($BB' > FBF$) are circled.

merging (Figure 13).

The lower-bound test does not extend to TIES at density $d = 0.5$: TIES keeps the dominant per-parameter values from each parent rather than averaging them, so the watermarked B side of BB' is largely preserved and TPR@1% remains high ($\sim$80–90% on English, $\sim$82–91% on German, $\sim$70–88% on Math) regardless of α_{eff}. BB' therefore sits above FBF at every α_{eff} and the proxy is uninformative; we omit the corresponding figure and instead report the raw values in Table 6.

Table 6. **TIES variant of the** FBF **vs.** BB' **lower-bound test (density** $d = 0.5$**):** Values are TPR@1% FPR (in %) at $\alpha = 0.5$. BB' remains substantially higher than FBF in every cell, so the lower-bound interpretation does not apply on TIES.

	FBF		BB'	
Domain	KGW-D	MAT	KGW-D	MAT
English	32.0	42.0	79.7	90.0
German	9.9	40.1	81.7	91.3
Math	5.7	34.7	70.0	87.7

A.3.3. FF VS. THE SINGLE-FT BASELINE

Table 7 compares the merged-model TPR@1% against the better of the two single-finetune baselines (max of Math FT and German FT, by domain) for both KGW-D and MAT across Linear, SLERP, and TIES at $\alpha \in \{0.3, 0.5, 0.7\}$. MAT sits above the single-FT baseline (green) in every Linear and SLERP cell, and dominates on the German domain across all three merge methods. On TIES, MAT clears the baseline on English and German across all three α, and on Math at

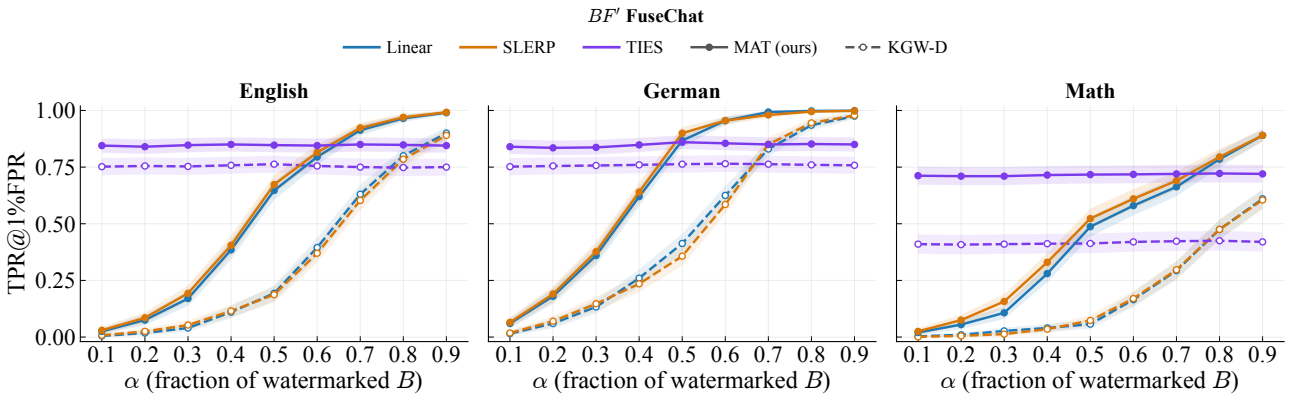

Figure 14. BF' **FuseChat (Wan et al., 2024) recovery:** TPR@1% as a function of α = fraction of the watermarked B in $B \otimes$ FuseChat, where FuseChat is a community finetune of LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024).

$\alpha \in \{0.5, 0.7\}$; the only orange cell (TIES Math at $\alpha = 0.3$, within 1 SE of the baseline) is unsurprising given that the watermark on Math is intrinsically weak post-SFT ($\sim$25–29% standalone), leaving little headroom for the merge to add signal. KGW-D, by contrast, falls below or within 1 standard error in 24 of 27 cells and only exceeds the baseline (green) on the TIES German rows. This is structural to TIES rather than a merge-recovery effect: TIES keeps the dominant per-parameter values from each parent, so the strong watermark signal in B that survives at standalone is preserved through the merge on the German domain, while Linear and SLERP dilute it. In other words, KGW-D cannot recover the signal beyond the single-finetune baseline under Linear and SLERP, while MAT consistently preserves or recovers it under Linear and SLERP and on the TIES German rows.

Table 7. FF **vs. single-FT baseline:** FF over Linear, SLERP, TIES at $\alpha \in \{0.3, 0.5, 0.7\}$. Values are TPR@1% FPR (in %). Cells colored relative to the *better* of the two single-finetune baselines per (method, domain) (i.e., the max of the MATH FT and GERMAN FT rows in Table 5): above by > 1 SE , within ± 1 SE , below by > 1 SE , where SE is the per-column binomial standard error of TPR@1% on $n = 500$ prompts.

		English		German		Math	
Method	α	KGW-D	MAT	KGW-D	MAT	KGW-D	MAT
Linear	0.3	28.4	49.6	41.8	79.8	10.3	39.7
	0.5	32.6	46.6	26.0	72.5	5.0	42.1
	0.7	29.4	39.0	10.2	51.4	9.7	44.0
SLERP	0.3	26.4	40.2	38.6	84.4	8.0	49.0
	0.5	34.8	47.0	21.6	70.4	8.3	53.0
	0.7	29.4	44.4	12.6	49.4	11.7	56.0
TIES	0.3	30.4	45.6	56.0	76.4	9.1	31.3
	0.5	30.2	49.4	53.6	78.0	10.7	32.9
	0.7	31.4	44.0	59.2	76.4	10.4	37.0

A.3.4. UNWATERMARKED-PARENT MERGES (BF' AND BB')

Figures 14–16 report the per-domain BF' recovery sweeps for the three community finetunes (FuseChat (Wan et al., 2024), OpenMath (Toshniwal et al., 2024), Tulu (Lambert et al., 2024)) and Figure 17 the BB' recovery against the unwatermarked LLAMA-3.1-8B-INSTRUCT base. Under Linear and SLERP, MAT keeps the watermark detectable down to smaller fractions of the watermarked base B than KGW-D in every domain. TIES behaves differently: on BF' it remains in the moderately-detectable regime for both methods (KGW-D $\sim$40–76%, MAT $\sim$70–86% across domains), with MAT consistently $\sim$8–12 pp above KGW-D; on BB' TIES stays high but no longer fully saturated, with KGW-D at $\sim$70–82% and MAT at $\sim$88–91%.

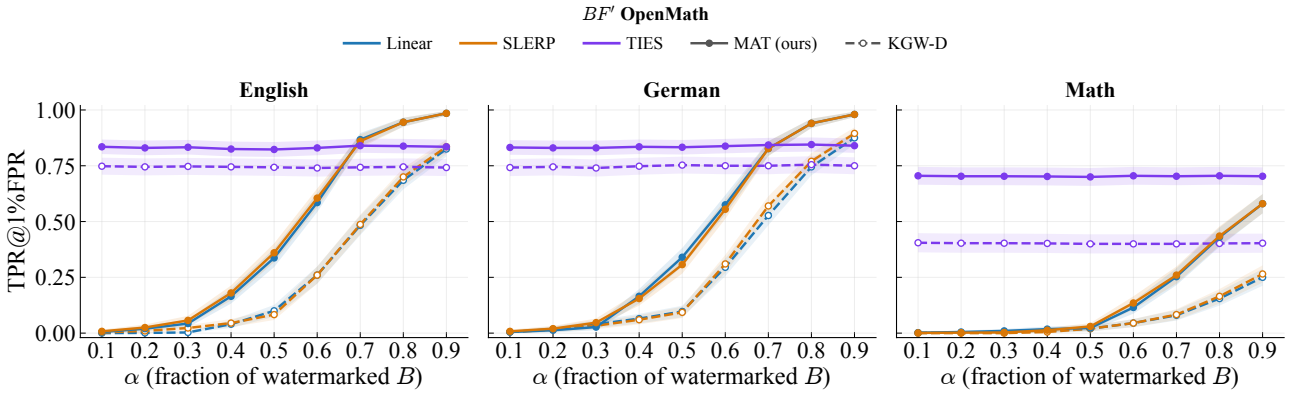

Figure 15. BF' **OpenMath (Toshniwal et al., 2024) recovery:** TPR@1% as a function of α = fraction of the watermarked B in $B \otimes$ OpenMath, where OpenMath is a community finetune of the LLAMA-3.1-8B base model (Grattafiori et al., 2024). Same conventions as Figure 14.

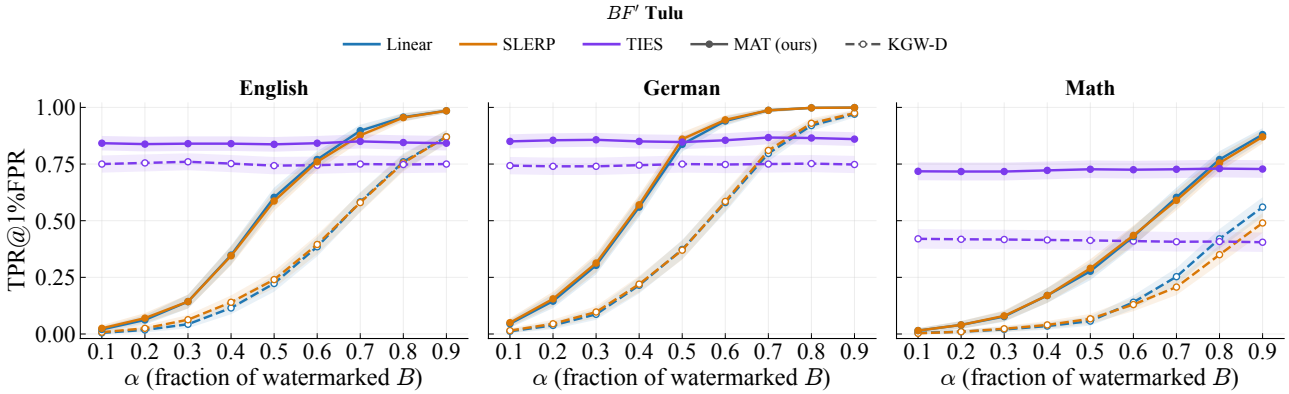

Figure 16. BF' **Tulu (Lambert et al., 2024) recovery:** TPR@1% as a function of α = fraction of the watermarked B in $B \otimes$ Tulu, where Tulu is a community finetune of the LLAMA-3.1-8B base model (Grattafiori et al., 2024). Same conventions as Figure 14.

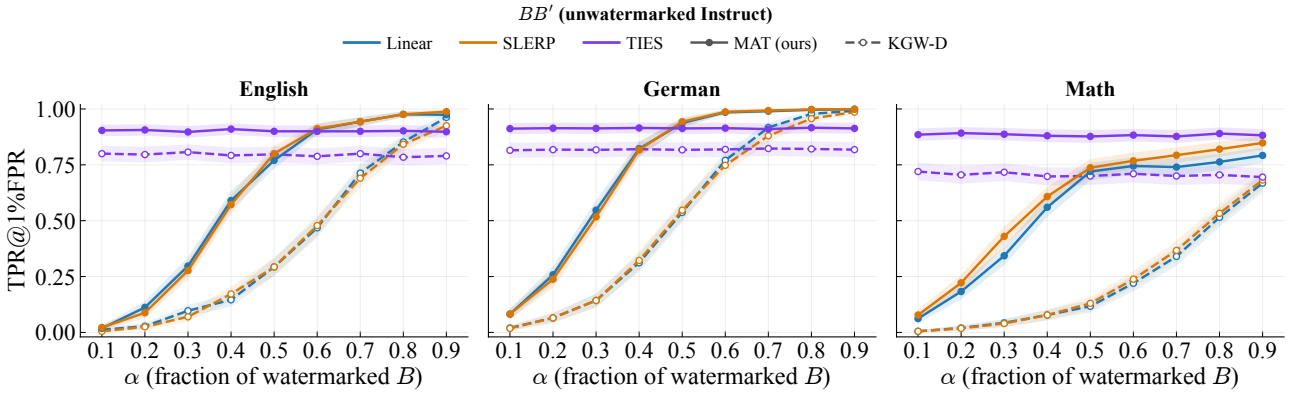

Figure 17. BB' **recovery:** TPR@1% as a function of α = fraction of the watermarked B in $B \otimes B'_{\text{unwm}}$, where B'_{unwm} is the unwatermarked LLAMA-3.1-8B-INSTRUCT base (Grattafiori et al., 2024).

A.3.5. POST-MERGE BENCHMARK RESULTS FOR FF AND BF

Table 3 reports lm-eval-harness accuracies for the unwatermarked base, the two watermarked bases (KGW-D, MAT KGW-D), and the per-domain finetunes *prior to merging*. Table 8 reports the same suite of benchmarks on the *merged* checkpoints in the two main all-watermarked merge configurations from Sec. 4.1: FF (Math FT $\otimes$ German FT) and

BF (the watermarked base B paired with each per-domain finetune). The objective is to verify that our merge-adversary objective does not degrade downstream capability of the merged checkpoint relative to the KGW-D baseline at matched merge configuration.

Table 8. **Post-merge benchmark results for FF and BF:** Per-task downstream accuracy and per-language reference perplexity of the merged models for the FF and BF scenarios at SLERP $\alpha \in \{0.5, 0.7\}$, for both KGW-D and MAT KGW-D. Accuracy cells are colored relative to the unwatermarked Instruct baseline (Table 3, BASES block): $\geq 95\%$, $\geq 90\%$, $< 90\%$. PPL is measured under the LLAMA-2-13B reference model on the same per-language corpora used in Table 4

Scenario	α	Variant	Accuracy [%]							PPL		
			ARC	MMLU	HeSw	GSM8K	MATH	MMLU_DE	MMLU_FR	EN	DE	FR
FF (MathFT $\otimes$ GermanFT)	0.5	KGW-D	56.7	65.5	71.4	76.7	42.0	54.7	53.8	5.42	4.65	5.86
		MAT KGW-D	56.7	64.8	71.1	76.5	43.0	54.7	54.7	5.68	4.55	5.42
	0.7	KGW-D	55.8	65.2	70.9	77.7	44.0	54.7	53.5	5.41	5.06	5.88
		MAT KGW-D	55.8	64.5	70.5	77.0	45.0	54.7	54.0	5.69	5.11	5.41
BF (B $\otimes$ MathFT)	0.5	KGW-D	56.2	65.3	72.7	82.1	44.5	54.3	56.4	5.40	5.68	5.90
		MAT KGW-D	56.0	64.8	73.0	82.4	45.5	53.9	57.0	5.69	5.65	5.45
	0.7	KGW-D	55.7	64.5	72.2	82.1	45.9	53.4	55.4	5.40	5.70	5.89
		MAT KGW-D	55.9	64.0	72.1	82.3	46.9	53.3	56.5	5.70	5.69	5.42

A.4. Generalization Across Watermark Families and Architectures

A.4.1. WATERMARK FAMILIES

Table 9 summarizes standalone (un-merged) detection and quality for the AAR and KTH watermark variants on English. Figure 18 and Figure 19 show the post-merge TPR@1% recovery on English under the BB' merge across Linear, SLERP, and TIES for AAR and KTH, respectively: in both schemes, the baseline distillation (AAR-D, KTH-D), distilled as in (Gu et al., 2024), collapses post-merge while MAT preserves detectability across α. For GAUSSMARK (Block et al., 2025) we report the baseline detection collapse under merging and discuss it separately in App. A.4.3. The QWEN-2.5-3B-INSTRUCT (Team, 2024) ablation under BB' is reported separately in Table 10.

Both AAR-D and KTH-D reproduce the merge-collapse pattern we observed for KGW-D: at $\alpha = 0.5$ under SLERP / Linear BB', post-merge TPR@1% drops from $\sim$87–90% standalone to $\sim$7% (AAR-D) and $\sim$2% (KTH-D). For MAT AAR and MAT KTH we apply only the merge-adversary training on top of the standard distillation; the top-k logit restriction we use for MAT KGW-D is not applied here. Even without that additional component, the merge-adversary objective alone recovers detection to $\sim$88% (MAT AAR) and $\sim$87% (MAT KTH) at $\alpha = 0.5$, a $+80$ to $+85$ pp gain, while leaving standalone detection and generation quality essentially unchanged (Table 9). The merge-induced collapse is therefore not specific to KGW and the merge-adversary objective transfers to other watermark families.

Table 9. **Standalone watermark-family ablation on LLAMA-3.1-8B-INSTRUCT, English (C4):** Pre-merge detection and quality for the AAR and KTH watermark variants. Post-merge results under BB' across Linear, SLERP, and TIES are in Figure 18 (AAR) and Figure 19 (KTH). GaussMark is reported separately in App. A.4.3.

Scheme	Variant	Detection	Quality	
		TPR@1%	PPL	rep-3
AAR	AAR-D	87.7	5.27	0.073
	MAT AAR (ours)	84.7	5.68	0.093
KTH	KTH-D	89.7	6.8	0.090
	MAT KTH (ours)	88.5	6.2	0.097

A.4.2. QWEN-2.5-3B-INSTRUCT MODEL

Table 10 reports the same merge-recovery effect on a different base architecture: QWEN-2.5-3B-INSTRUCT (Team, 2024) ($\delta = 2.3$, $\gamma = 0.25$; logit-processor reference 100% TPR@1%, PPL = 11.67, rep-3 = 0.041), evaluated on English (C4) under the BB' merge with the unwatermarked QWEN-2.5-3B-INSTRUCT.

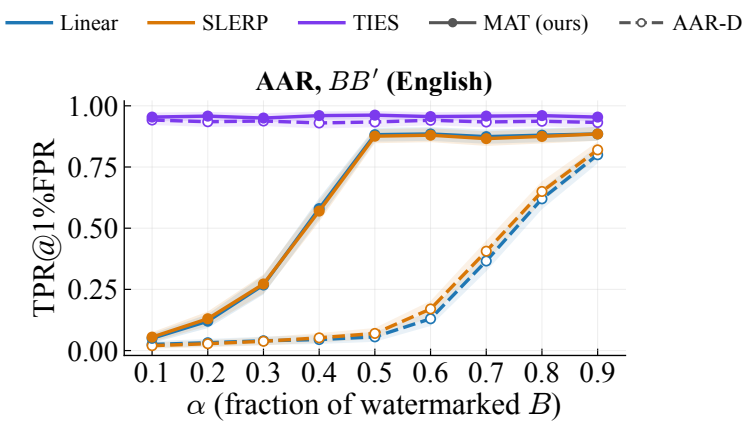

Figure 18. **AAR BB' scenario on English domain:** TPR@1% as a function of α = fraction of the watermarked AAR base B in BB', across Linear, SLERP, and TIES. AAR-D (dashed, open markers) collapses on Linear/SLERP at $\alpha \leq 0.5$; MAT AAR preserves detectability down to $\alpha = 0.3$. TIES is saturated for both variants.

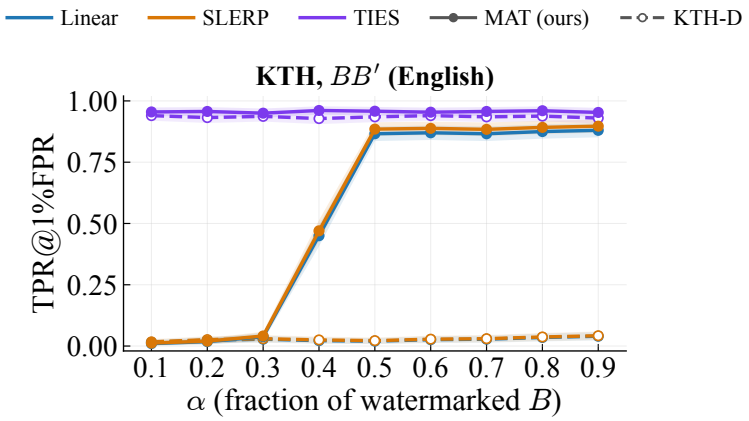

Figure 19. **KTH BB' scenario on English domain:** TPR@1% as a function of α = fraction of the watermarked KTH base B in BB', across Linear, SLERP, and TIES. KTH-D (dashed, open markers) collapses on Linear/SLERP at $\alpha \leq 0.4$; MAT KTH (solid, filled markers) preserves detectability from $\alpha = 0.4$ upward. TIES is saturated for both variants.

We did not retune any of the MAT hyperparameters when porting them from LLAMA-3.1-8B-INSTRUCT: the same δ, γ, top-k restriction, and merge-adversary α schedule were applied directly. Even with this transfer, MAT KGW-D improves post-merge English TPR@1% under BB' by $+17$ to $+20$ pp at $\alpha \in \{0.5, 0.7\}$ over the KGW-D baseline (Table 10). The gain shrinks at $\alpha = 0.3$ where the unwatermarked Qwen parent dominates the merge, but the qualitative recovery pattern matches the LLAMA base, suggesting the merge-adversary objective does not require base-specific tuning to provide a benefit.

Table 10. **Qwen-2.5-3B-Instruct BB' merge ablation:** TPR@1% under SLERP BB' across $\alpha \in \{0.3, 0.5, 0.7\}$ on English (C4). KGW-D = standard KGW logit distillation applied to Qwen-2.5-3B-Instruct; MAT KGW-D = the same with merge-adversary training and top-k. Δ = MAT KGW-D − KGW-D (in pp); cells colored $\geq$+10 pp, $\geq$+3 pp, uncolored <+3 pp.

		TPR@1%		
Merge	α	KGW-D	MAT KGW-D (ours)	Δ
BB' (SLERP)	0.3	9.0	13.8	**+4.8**
	0.5	27.4	44.8	**+17.4**
	0.7	58.4	78.4	**+20.0**

A.4.3. GAUSSMARK

GAUSSMARK (Block et al., 2025) is a weight-space watermark: instead of biasing logits or modifying sampling, it perturbs a fixed subset of the model parameters with low-amplitude Gaussian noise drawn from a key-dependent distribution, and detection runs a likelihood-ratio test on those parameters.

We apply GAUSSMARK to LLAMA-3.1-8B-INSTRUCT as described in Block et al. (2025), and then run the same BB' merge sweep used for KGW-D. Standalone (no merge), the watermarked model reaches $\sim$89% TPR@1% on English (Table 11). Under BB' at $\alpha = 0.5$ under SLERP, it drops to $\sim$34% TPR@1% (Figure 20), the same collapse pattern we observe for KGW-D, AAR-D, and KTH-D.

The GaussMark panel in Figure 7 reports the baseline collapse and is included to underline that the collapse is not an artifact of any single watermarking mechanism.

Table 11. **Standalone GaussMark ablation on LLAMA-3.1-8B-INSTRUCT, English (C4):**

	Detection	Quality	
Variant	TPR@1%	PPL	rep-3
GAUSSMARK	89.0	4.06	0.061

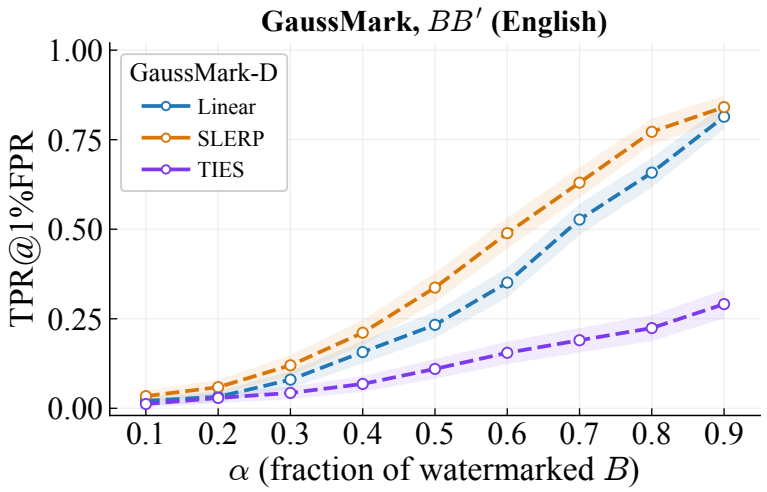

Figure 20. **GaussMark BB' merge scenario on English domain:** TPR@1% under the BB' merge as a function of α = fraction of the watermarked GaussMark base B, across Linear, SLERP, and TIES.

B. Ablations

In this section, we ablate the components of our method presented in Sec. 3.1.

Experimental Setup All ablations are run on LLAMA-3.2-1B-INSTRUCT for 7 500 steps (LR = 1e−5, 500-step warm-up, cosine decay). The hyperparameters are otherwise identical to the full LLAMA-3.1-8B-INSTRUCT training described in App. A.1. We vary one hyperparameter at a time from the baseline settings ($k = 200$, $\alpha_{\min} = 0.1$, $\delta = 2.0$) and then validate the joint best in a second pass. The hyperparameters used in Sec. 3 correspond to the bolded row in each table: $\delta = 2.3$, $k = 100$, $\alpha_{\min} = 0.1$. Each row reports standalone (un-merged) TPR@1% on C4 with PPL and seq-rep-3, alongside post-merge TPR@1% under Linear merging at $\alpha = 0.5$ as the headline robustness number.

B.1. Top-k Gating (k)

k controls how many top-ranked teacher tokens the watermark delta is applied to. Table 12 sweeps $k \in \{50, 100, 200, 300, 500\}$. Post-merge robustness broadly trends upward with k at the cost of PPL; $k = 100$ gives the

best PPL/robustness trade-off (lower PPL than the base while matching its post-merge TPR), staying within 1 pp of the base standalone TPR, so we move it into the joint search. Compared to the no-gating baseline of Gu et al. (2024), restricting watermark distillation to the top-k logits lowers PPL while leaving standalone TPR essentially unchanged (within $\sim$2 pp).

Table 12. **Top-k ablation:** Vary k; hold $\alpha_{\min} = 0.1$, $\delta = 2.0$. Standalone and post-merge TPR@1% in %; PPL on LLAMA-3.2-1B. Best checkpoint values reported.

k	TPR@1% (st.)	PPL	TPR@1% ($\alpha = 0.5$)	seq-rep-3
50	95.7	5.58	55.7	0.097
100	**95.3**	**5.90**	**60.0**	**0.094**
200 (base)	96.0	6.25	59.7	0.083
300	97.3	6.28	65.7	0.094
500	96.7	6.45	67.0	0.089

B.2. Watermark Strength (δ)

δ is the green-list bias added by the teacher. Higher δ produces a stronger watermark signal but degrades quality. Table 13 sweeps $\delta \in \{1.0, 2.0, 2.3, 2.5, 2.7\}$. $\delta = 1.0$ is too weak (standalone TPR $\sim 55\%$); $\delta = 2.3$ gives the cleanest robustness/quality trade-off, substantially improving post-merge TPR over the base without the PPL inflation we see at $\delta = 2.5$ / 2.7.

Table 13. **Watermark-strength ablation:** Vary δ; hold $k = 200$, $\alpha_{\min} = 0.1$. Best checkpoint values reported.

δ	TPR@1% (st.)	PPL	TPR@1% ($\alpha = 0.5$)	seq-rep-3
1.0	55.3	4.98	11.7	0.077
2.0 (base)	96.0	6.25	59.7	0.083
2.3	**98.3**	**6.69**	**76.7**	**0.074**
2.5	99.0	7.11	89.3	0.086
2.7	99.0	7.82	89.7	0.068

B.3. Merge-Adversary Lower Bound ($\alpha_{\min}$)

The merge adversary samples $\alpha \sim \mathcal{U}(\alpha_{\min}, 1)$ at each step. Lower $\alpha_{\min}$ exposes the student to more aggressive merges. Table 14 sweeps $\alpha_{\min} \in \{0.1, 0.3, 0.5\}$. Robustness drops sharply as $\alpha_{\min}$ rises: at $\alpha_{\min} = 0.5$, post-merge TPR@1% drops by a third vs the base, because the model never trains against $\alpha < 0.5$. $\alpha_{\min} = 0.1$ wins clearly and we keep it for the joint config.

Table 14. **Merge-adversary lower-bound ablation:** Vary $\alpha_{\min}$; hold $k = 200$, $\delta = 2.0$. Best checkpoint values reported.

$\alpha_{\min}$	TPR@1% (st.)	PPL	TPR@1% ($\alpha = 0.5$)	seq-rep-3
0.1 (base)	**96.0**	**6.25**	**59.7**	**0.083**
0.3	97.3	6.46	54.0	0.085
0.5	97.7	6.37	40.3	0.083

B.4. Joint Search

Table 15 validates the per-parameter winners ($k = 100$, $\alpha_{\min} = 0.1$) jointly with a δ sweep, fixing the other parameters at base. The headline trade-off is δ vs PPL: $\delta = 2.5$ recovers 88.3% at $\alpha = 0.5$ but pays $+0.90$ PPL over $\delta = 2.0$, while $\delta = 2.3$ recovers 80.3% at $\alpha = 0.5$ for only $+0.26$ PPL. We choose $\boldsymbol{\delta = 2.3,\ k = 100,\ \alpha_{\min} = 0.1}$ as the main-paper config: best PPL/robustness frontier without the seq-rep-3 risk of higher δ. Increasing k from 100 back to 200 at the same δ slightly hurts both robustness (-3.3 pp) and PPL, confirming $k = 100$ as the joint-best.

Table 15. **Joint search:** Combine the per-parameter winners ($\alpha_{\min} = 0.1$ throughout) with a δ sweep; the row in **bold** is the main-paper configuration.

k	δ	TPR@1% (st.)	PPL	TPR@1% ($\alpha = 0.5$)	seq-rep-3
100	2.0	96.7	6.32	60.3	0.086
100	2.2	98.0	6.56	79.7	0.080
100	**2.3**	**97.0**	**6.58**	**80.3**	**0.082**
200	2.3	97.7	6.82	77.0	0.078
100	2.5	97.3	7.22	88.3	0.081

B.5. Importance of Sampling α

We test whether sampling α matters at all by replacing the uniform adversary with α *fixed* at 0.3. Table 16 shows the fixed-α run sits below the sampled baseline at *every* evaluated α, including 0.3 itself; robustness collapses away from the training point (2.7% at $\alpha = 0.1$, 40.0% at $\alpha = 0.5$, -19.7 pp vs. baseline) and PPL rises sharply (9.35 vs. 6.25) as the student over-fits the single training point. Uniform α sampling outperforms the fixed-α schedule at every α and at standalone PPL, which is why we adopt it as our default.

Table 16. **Sampled vs. fixed merge-adversary α:** Post-merge TPR@1% across the full α range (Linear merging on C4). Top row: the default merge adversary, which samples $\alpha \sim \mathcal{U}(0.1, 1)$ at every training step (same configuration as $\alpha_{\min} = 0.1$ in Table 14).

Merge-adversary α	**Post-merge TPR@1% at eval $\alpha = \cdot$**					**Standalone TPR@1%**	**PPL**
	0.1	0.3	0.5	0.7	0.9		
$\alpha \sim \mathcal{U}(0.1, 1)$ (sampled, default)	12.7	38.7	59.7	81.0	93.7	96.0	6.25
Fixed $\alpha = 0.3$ (no sampling)	2.7	10.0	40.0	70.0	91.3	96.7	9.35

C. Existing Assets and Licenses

We use the following existing datasets and models. We list the license names as reported by the corresponding dataset or model cards, repositories, or project pages.

Datasets

- ALPACA-GPT4: Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).
- OPENWEBTEXT: Creative Commons Zero v1.0 Universal (CC0 1.0) for the dataset packaging; the dataset creators state that they do not own the underlying extracted text.
- ALPACA-GPT4-DE: Apache License 2.0.
- FINEWEB-2: Open Data Commons Attribution License v1.0 (ODC-BY 1.0).
- METAMATHQA: MIT License.
- CODEALPACA: Creative Commons Attribution 4.0 International (CC BY 4.0) for the dataset release.
- NUMINAMATH-COT: Apache License 2.0.
- EVOL-INSTRUCT-DE: Apache License 2.0.
- FRENCH-ALPACA: Apache License 2.0.
- Lucie Training Dataset: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).
- C4: Open Data Commons Attribution License v1.0 (ODC-BY 1.0).

- GSM8K: MIT License.
- ARC-CHALLENGE: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).
- MMLU: MIT License.
- HELLASWAG: MIT License.
- MATH: MIT License.

Models

- LLAMA-3.1-8B-INSTRUCT: Llama 3.1 Community License.
- QWEN-2.5-3B-INSTRUCT: Qwen Research License Agreement.
- LLAMA-2-13B: Llama 2 Community License Agreement.
- FUSECHAT-LLAMA-3.1-8B-INSTRUCT: Apache License 2.0 as listed by its model card; because it is derived from LLAMA-3.1-8B-INSTRUCT, we also treat it as subject to the Llama 3.1 Community License.
- OPENMATH2-LLAMA3.1-8B: Llama 3.1 Community License.
- LLAMA-3.1-TULU-3-8B: Llama 3.1 Community License.

D. Broader Impacts

Our work aims to make provenance signals for open-source LLMs more durable under common downstream modifications. A positive impact is that robust watermarking can help model providers, auditors, and researchers attribute generated text after benign post-training workflows such as finetuning and model merging. This can support accountability and make deployed open-source systems easier to monitor.

The same capability also has limitations and risks. Watermark detection can be misinterpreted if users ignore false-positive rates, key-management assumptions, or the possibility that downstream transformations fall outside the evaluated threat model. More robust watermarks may also strengthen unilateral control over models after release, so deployments should pair detection claims with transparent operating thresholds, appeals or audit procedures where relevant, and clear communication about the detector's scope.