PREVENTING LORA FORGETTING THROUGH TARGET MASKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) has become one of the most widely used fine-tuning mechanisms for adapting large language models to new domains, tasks, and users. Yet adaptation performance alone can obscure an important failure mode: LoRA updates may improve performance on the target distribution while degrading prior capabilities learned during pretraining and alignment. We show that this forgetting becomes especially severe when the adaptation distribution differs substantially from the model's original training or alignment distributions. The challenge is amplified in practical settings, where the original training and alignment data are typically unavailable. Motivated by this constraint, we study how LoRA based adaptation balances new learning against forgetting in a replay-free setting, and introduce a simple output space regularizer that can be added directly to existing training pipelines. Our method removes the ground-truth token from both the base and adapted model distributions, renormalizes the remaining probabilities, and applies KL regularization only over the non-target vocabulary. This preserves the base model's relative preferences among alternative tokens without directly opposing the cross-entropy signal required for adaptation. As the regularizer acts only at the loss level, it requires no replay data, architectural changes, adapter redesign, or inference-time overhead, and can be applied directly to existing LoRA variants. Across all LoRA variants tested and across various backbones, our method improves the frontier between new learning and forgetting when the adaptation distribution differs substantially from the base model's original training or alignment distributions, suggesting a broadly applicable route toward more reliable LLM updating.

1 INTRODUCTION

Low-Rank Adaptation (LoRA) is the default mechanism for adapting large language models after deployment, and it is almost always evaluated only on the new distribution. The other half of the ledger, what the model loses, is rarely measured. Under the distribution shifts that motivate adaptation in the first place, LoRA-family methods can quietly erode capabilities that practitioners assume are preserved (Biderman et al., 2024; Shuttleworth et al., 2025), and the *replay-free* regime, where the original training and alignment data are unavailable, is precisely where this matters and is least studied. This is the modern LLM instance of the classical learning-forgetting problem (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017; Li & Hoiem, 2017; Wang et al., 2024).

We propose **Target-Masked KL** (TMKL), a one-line addition to the training loss that separates *what should change* from *what should be preserved*. At each supervised position, TMKL removes the target token from both the frozen base and the adapted next-token distributions, renormalizes the remaining vocabulary, and applies KL only over this non-target distribution. The masking step matters: standard distillation objectives such as Learning without Forgetting (Li & Hoiem, 2017) constrain the full output distribution and therefore directly oppose the cross-entropy gradient under distribution shift, because the base model assigns low probability to precisely the target-domain tokens that cross-entropy must learn. TMKL removes that conflict by construction while keeping the base model's preferences over every other token. The mechanism is the autoregressive next-token analog of the non-target component of Decoupled Knowledge Distillation (Zhao et al., 2022).

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.

Existing LoRA-based continual learning intervenes in the adapter's *weight space*: O-LoRA constrains successive tasks to orthogonal low-rank subspaces (Wang et al., 2023), CL-LoRA splits adapters into task-shared and task-specific modules with distillation and gradient reassignment (He et al., 2025), and recent methods route between adapter or prompt pools (Wang et al., 2025; Yu et al., 2024). Each requires extra components, task identity, or architectural modification, and is tightly coupled to LoRA's specific low-rank parameterization. The coupling is the problem: when the same weight-space recipes are applied to LoRA-family variants whose update geometry *is* the design contribution (e.g., DoRA's magnitude/direction split, RandLoRA's random-basis bank), they interfere with the variant's own mechanism.

TMKL acts only on the next-token output distribution and never touches the adapter's weight space. The same loss term therefore composes cleanly with every LoRA-family adapter we test, leaving the rank, initialization, and routing exactly as the adapter's authors intended. The method has a single tunable scalar λ that controls regularization strength; its only added training cost is one extra forward pass through the frozen base model per step, and there is no inference-time overhead.

Our headline finding is striking: *when TMKL is added during a single training run, the original capabilities of the base model are preserved while the adapter still learns the new domain.* On Qwen2.5-7B adapted to PubMed (Cohan et al., 2018), plain cross-entropy raises WikiText perplexity (general English) by $+15$ to $+20\%$ and LAMBADA perplexity (long-range English) by $+25$ to $+33\%$ across four LoRA-family adapters; TMKL keeps both retention sets (prior knowledge) at or near the unadapted base while staying within ~ 0.13 PPL of cross-entropy on the target. Figure 2 previews the same pattern at 0.5B.

The contributions of this paper are as follows.

- **LoRA forgetting under replay-free distribution-shifted adaptation is real, measurable, and consistent across adapter designs and model scales.** On Qwen2.5-0.5B adapted to a post-cutoff math-reasoning corpus, OpenR1-Math (HuggingFace Open-R1 Team, 2025), every adapter we test (LoRA, SineLoRA, RandLoRA, DoRA) raises retention perplexity by 20 to 42% on WikiText-103 (Merity et al., 2017) and LAMBADA (Paperno et al., 2016), i.e. a clear drift (forgetting) on prior English. The same pattern persists at Qwen2.5-7B on PubMed (Cohan et al., 2018): $+15$ to $+20\%$ WikiText drift and $+25$ to $+33\%$ LAMBADA drift on every adapter under cross-entropy.
- **Target-Masked KL is a one-line, adapter-agnostic regularizer.** Because it acts only on the next-token output distribution, it composes with any LoRA-family adapter without modifying the adapter, requires no replay data, and adds no inference cost.
- **TMKL preserves the base model's prior knowledge while still letting the adapter learn the new domain.** At 0.5B, TMKL prevents 88 to 92% of the WikiText drift and 95 to 98% of the LAMBADA drift on every adapter while improving or matching target adaptation versus plain cross-entropy; at 7B, both retention sets stay within $\leq 1.5\%$ of the unadapted base while target adaptation stays within ~ 0.13 PPL of cross-entropy. The same recipe transfers without retuning to instruction-tuned bases (Qwen2.5-7B-Instruct: IFEval, MT-Bench, and refusal calibration within 1pp of base) and to non-Qwen backbones (Llama-3.2-1B, Llama-3.1-8B, Mistral-7B-v0.3, Phi-3.5-mini-instruct), and preservation extends beyond English LM-PPL to factual recall, math reasoning, code, and multilingual proxies.

2 Related Work

LoRA and parameter-efficient fine-tuning. Low-Rank Adaptation freezes the pretrained backbone and learns low-rank updates that can be merged at inference (Hu et al., 2021). A large body of work modifies the rank allocation, decomposition, initialization, or basis of the update (Zhang et al., 2023; Liu et al., 2024; Kopiczko et al., 2024; Meng et al., 2025; Albert et al., 2025b; Ji et al., 2025; Koohpayegani et al., 2024; Valipour et al., 2023; Rajabzadeh et al., 2024; Albert et al., 2025a); other PEFT families adapt different parts of the model (Houlsby et al., 2019; Li & Liang, 2021; Lester et al., 2021; Ben-Zaken et al., 2022). These variants improve the *parameterization* of the update; Target-Masked KL is orthogonal, leaving the LoRA-family adapter untouched and acting only at the loss level. An expanded taxonomy of LoRA variants and PEFT families is provided in Section A.

Replay-free continual adaptation with PEFT. Continual-learning methods preserve historical information through stored examples (Chaudhry et al., 2019; Buzzega et al., 2020), architectural isolation or expansion (Yan et al., 2021; Douillard et al., 2022), or parameter-/function-space regularization (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Titsias et al., 2020). Replay-based methods conflict with the replay-free setting; architecture-based methods require task identity, routing, or growth. With PEFT, these principles have been instantiated as LoRA-specific mechanisms: O-LoRA's orthogonal subspaces (Wang et al., 2023), CL-LoRA's dual-adapter design with knowledge distillation and gradient reassignment (He et al., 2025), and prompt or adapter-pool routing (Wang et al., 2025; Yu et al., 2024). Recent work also shows that LoRA itself exhibits a learning-forgetting tradeoff and produces structurally different solutions from full fine-tuning, with LoRA-specific spectral directions linked to forgetting (Biderman et al., 2024; Shuttleworth et al., 2025). Concurrent loss-level methods for replay-free LLM continual learning include CLoRA, which constrains the LoRA update to a learned null subspace (Lu et al., 2025); C-LoRA, which combines orthogonality with adapter routing (Zhang et al., 2025); InfLoRA, which restricts updates to interference-free directions (Liang & Li, 2024); and STABLE, which gates updates using a full base-vs-adapted KL stability metric (Hoy & Celik, 2025). Of these, STABLE is the closest neighbor to TMKL because it computes a related base-vs-adapted next-token divergence, but uses the unmasked full-distribution KL as an acceptance gate rather than the renormalized non-target KL as an additive loss; Table 3 shows the empirical gap. Target-Masked KL differs from these methods by adding no adapters, no routing, and no architectural growth, while regularizing only the renormalized non-target output distribution.

Output-space distillation and non-target knowledge. Knowledge distillation matches softened teacher distributions (Hinton et al., 2015), and Learning without Forgetting applies this to retention by distilling the frozen old model when old-task data are unavailable (Li & Hoiem, 2017); functional regularization formalizes preserving predictive functions rather than parameters (Titsias et al., 2020). Recent KD work shows that useful teacher information lies in non-target, relational, or ranking-based logit structure (Huang et al., 2022; Sun et al., 2024; Bassam et al., 2025; Zhao et al., 2022); most relevantly, Decoupled Knowledge Distillation separates target and non-target components and demonstrates the importance of the non-target "dark knowledge" (Zhao et al., 2022). The closest named mechanism in the vision-classification literature is NTCE-KD, which suppresses the target-class logit before applying KL on the non-target classes (Li et al., 2024); the renormalization we use is, up to the row corresponding to the target itself, mathematically equivalent to that logit-masking step. TMKL differs from NTCE-KD in two contextual respects: (i) it operates on autoregressive next-token distributions rather than image-class logits, with per-position summation over a teacher-forced sequence; and (ii) it uses the frozen *base model* of the LoRA adaptation as the implicit teacher, requiring no separate teacher network or pretraining-data access. Target-Masked KL transfers this target-aware decomposition to autoregressive replay-free LoRA adaptation: the supervised token is masked, the remaining vocabulary is renormalized, and KL is computed only on the renormalized non-target distribution. To our knowledge, this is the first autoregressive next-token instantiation of a target/non-target KL decomposition for replay-free LoRA fine-tuning of large language models.

3 METHOD

We take the next-token distribution of the frozen base model, remove the probability of the supervised target token, and renormalize what remains. We do the same to the LoRA-adapted model's distribution at the same position. Target-Masked KL is the KL divergence between these two renormalized distributions, summed over supervised positions and added to standard cross-entropy. This separates target learning from base-model preservation: cross-entropy is free to move the target-token probability, while the regularizer keeps the rest of the vocabulary distribution close to the base model. The base model is held fixed and only its output distribution is read; no replay data, no architectural change, and no inference overhead are introduced. Figure 1 summarizes the method, which adds one term to the training loss.

3.1 THE TARGET-MASKED KL LOSS

Let f_{θ_0} be a pretrained causal language model with frozen parameters θ_0 and vocabulary $\mathcal{V}$. We adapt it on a new dataset $\mathcal{D}_{\text{new}} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ of (instruction prompt, supervised response) pairs using LoRA-family adapter parameters ϕ (the LoRA parameterization is recalled in Section G; the

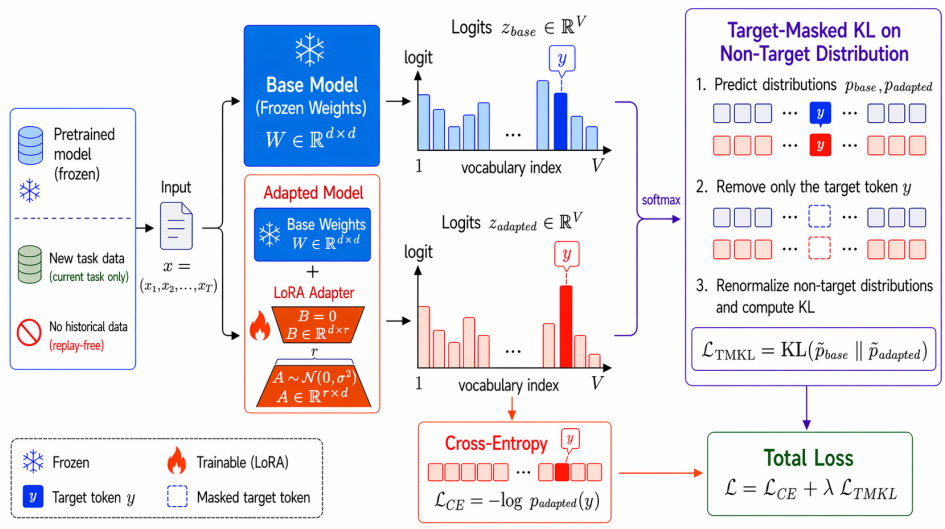

Figure 1: **Overview of Target-Masked KL regularization.** At each supervised token position, the frozen base model produces a next-token distribution p_{base} over the vocabulary, and the LoRA-adapted model produces a next-token distribution p_{adapted} for the same context. Cross-entropy is computed on the supervised target token y exactly as in standard LoRA fine-tuning. Target-Masked KL adds a second term: it removes the target probability $p(y)$ from both distributions, renormalizes the remaining $|\mathcal{V}| - 1$ vocabulary entries so each becomes a proper distribution conditioned on "the next token is not y," and matches the two renormalized distributions via a KL divergence with the base as the fixed reference. The total loss is $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\setminus y}$. Only the training objective changes; the deployed adapted model is identical in form to one trained with cross-entropy alone, so inference is unchanged and there is no inference-time overhead.

regularizer below depends only on the adapted output distribution, not on the specific adapter design). For a token sequence $x = (x_1, \ldots, x_T)$, write the next-token label at position t as $y_t = x_{t+1}$, and write the base and adapted next-token distributions as

$$p_{\text{base},t}(\cdot) = p_{\theta_0}(\cdot \mid x_{\le t}), \qquad p_{\text{adapted},t}(\cdot) = p_{\theta_0,\phi}(\cdot \mid x_{\le t}).$$

Let $\mathcal{M}$ be the set of supervised response-token positions. Standard LoRA fine-tuning minimizes the response-token cross-entropy

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \log p_{\text{adapted},t}(y_t).$$

For each supervised position t, we define the renormalized non-target distributions on $\mathcal{V} \setminus \{y_t\}$:

$$p^{\setminus y_t}_{\text{base},t}(c) = \frac{p_{\text{base},t}(c)}{1 - p_{\text{base},t}(y_t)}, \qquad p^{\setminus y_t}_{\text{adapted},t}(c) = \frac{p_{\text{adapted},t}(c)}{1 - p_{\text{adapted},t}(y_t)}, \qquad c \neq y_t.$$

Each of these is the original distribution conditioned on the event "the next token is not y_t". Both renormalizations are well-defined for softmax outputs since $p(y_t) < 1$ strictly; in the rare regime where $p_{\text{base},t}(y_t)$ is near 1 the base already agrees with the target and the position carries no retention signal, so we exclude such positions from $\mathcal{L}_{\setminus y}$ via a threshold (default $1 - 10^{-4}$; in practice $\le 0.21\%$ of supervised positions are excluded on every setting we measured, see Section E.12). The Target-Masked KL term, written $\mathcal{L}_{\setminus y}$, is the KL between these two renormalized distributions, averaged over supervised positions:

$$\mathcal{L}_{\setminus y} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \mathrm{KL}\left(p^{\setminus y_t}_{\text{base},t} \,\middle\|\, p^{\setminus y_t}_{\text{adapted},t} \right).$$

Gradients are not propagated through $p_{\text{base},t}$; only the adapter parameters ϕ are updated. The final training objective combines target-domain learning and retention through a single hyperparameter $\lambda \ge 0$:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\setminus y}. \tag{1}$$

Setting $\lambda = 0$ recovers standard LoRA fine-tuning (the $\lambda \to \infty$ limit is characterized in Section E.4). The regularizer is computed only at training time and is discarded at inference, so the deployed adapted model is identical in form to one trained with cross-entropy alone.

3.2 Why Mask the Target: Decomposing Full KL

A natural alternative is to apply KL between the full base and adapted next-token distributions:

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \mathrm{KL}(p_{\mathrm{base},t} \,\|\, p_{\mathrm{adapted},t}) \,.$$

This option is simpler but directly opposes the cross-entropy objective under distribution shift, as the following decomposition makes precise. Drop the position subscript t for clarity and write $p_b = p_{\mathrm{base},t}$, $p_a = p_{\mathrm{adapted},t}$, and $y = y_t$. Splitting the sum over the vocabulary into the target token and its complement gives the identity

$$\begin{aligned} \mathrm{KL}(p_b \,\|\, p_a) = \underbrace{p_b(y) \log \tfrac{p_b(y)}{p_a(y)}}_{\text{(i) target probability}} + \underbrace{\big(1 - p_b(y)\big) \log \tfrac{1-p_b(y)}{1-p_a(y)}}_{\text{(ii) total non-target mass}} \\ + \underbrace{\big(1 - p_b(y)\big)\, \mathrm{KL}\Big(p_b^{\setminus y} \,\Big\|\, p_a^{\setminus y}\Big)}_{\text{(iii) non-target shape}} . \end{aligned} \tag{2}$$

A short derivation is in Section G. The three terms have distinct roles. Terms (i) and (ii) together form the binary KL divergence between $\big(p_b(y), 1 - p_b(y)\big)$ and $\big(p_a(y), 1 - p_a(y)\big)$ and jointly penalize any deviation between the base and adapted target-token probabilities; in isolation, Term (i) alone is monotone-decreasing in $p_a(y)$ and the penalty arises only from the binary-KL sum. Term (iii) penalizes how the non-target mass is redistributed across the rest of the vocabulary.

Under distribution-shifted adaptation, the base model assigns a small probability $p_b(y)$ to the target-domain token, while cross-entropy must push the adapted probability $p_a(y)$ substantially higher. For $p_a(y) > p_b(y)$, the gradient of (i)+(ii) with respect to the adapted target logit has the opposite sign to the cross-entropy gradient: as $p_a(y)$ rises, the binary KL between $\big(p_b(y), 1 - p_b(y)\big)$ and $\big(p_a(y), 1 - p_a(y)\big)$ grows and pulls back. This is precisely the regime cross-entropy training drives toward. Only term (iii) is orthogonal to target learning, since it is computed entirely after both distributions have been conditioned on the non-target event.

Target-Masked KL keeps only this orthogonal component. The natural regularizer derived from term (iii) carries a per-position weight $\big(1 - p_{\mathrm{base},t}(y_t)\big)$ that down-weights positions where the base is already confident in the target token. We deliberately drop this weight and treat each supervised position uniformly:

$$\mathcal{L}_{\setminus y} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \mathrm{KL}\Big(p_{\mathrm{base},t}^{\setminus y_t} \,\Big\|\, p_{\mathrm{adapted},t}^{\setminus y_t}\Big) \,,$$

which matches the loss defined in Section 3.1. The choice is empirical: the unweighted form is uniformly stronger by ~ 5 to 15pp on retention prevention across all four headline adapters (Table 26). We note that since the term-(iii) weight $(1 - p_b(y))$ is strictly ≤ 1, it also scales down the total magnitude of the regularization loss, partially confounding this comparison with a reduction in effective λ, a matched three-seed sweep (Section E.4) finds the weighted variant's realized mean weight is 0.347, so equal λ is not equal effective strength. We also use the forward direction $\mathrm{KL}(p_{\mathrm{base}} \| p_{\mathrm{adapted}})$ rather than reverse or symmetric KL: the forward direction is mode-covering on the base, which is the desired retention behavior, and the ablation in Table 24 confirms it dominates reverse and Jensen-Shannon variants under the same masking and renormalization. Target-Masked KL therefore preserves the base model's relative preferences over alternative tokens without contesting the cross-entropy gradient on the target token. The target/non-target split underlying this construction is the autoregressive next-token analog of the decomposition used by Decoupled Knowledge Distillation (Zhao et al., 2022) for image classification and most directly mirrors NTCE-KD (Li et al., 2024), which suppresses the target-class logit before non-target KL in vision classification (the renormalization we use is, up to the target row, equivalent to masking the target logit and applying softmax); TMKL ports this construction to per-position autoregressive next-token distributions and uses the frozen base of the same adaptation as the implicit teacher rather than a separate teacher network. A

local geometric interpretation of $\mathcal{L}_{\setminus y}$ as a Fisher-weighted Jacobian penalty in the LoRA-admissible update space, together with a one-line proof that the binary-KL gradient on $p_a(y)$ has the opposite sign to cross-entropy whenever $p_a(y) > p_b(y)$, is given in Section G.

4 EXPERIMENTS

We evaluate Target-Masked KL (TMKL) on two replay-free LoRA-family adaptation settings: a small-scale grid (Qwen2.5-0.5B) where we run multi-seed comparisons and ablations, and a 7B (Qwen2.5-7B) replication that confirms the same effect at production scale. Across both, the only data available during adaptation is the new target corpus; no original training data, alignment data, or stored historical logits are used for replay.

4.1 SETUP

Models, targets, and retention. Qwen2.5-0.5B (Ahmed et al., 2025) is adapted to OpenR1-Math (HuggingFace Open-R1 Team, 2025), a math-reasoning corpus released after Qwen2.5's pretraining cutoff, so the target tokens are verifiably unseen. Qwen2.5-7B is adapted to PubMed (Cohan et al., 2018), a biomedical corpus that introduces a strong distribution shift away from the model's general-text pretraining. For both settings, retention is measured as the change in token-level perplexity on WikiText-103 validation (Merity et al., 2017) and LAMBADA test (Paperno et al., 2016); both are perplexity-natural (no template, no scripted continuation), which avoids the template-PPL artifacts that confound classification-style retention suites (Biderman et al., 2024). We use *drift* for any change on a retained distribution and reserve *forgetting* for degradation on it. The full rationale for these choices (why Qwen2.5 specifically, why post-cutoff target corpora, why these two retention benchmarks) is laid out in Section B.1.

Adapters and objectives. The main text reports four LoRA-family adapters spanning the design dimensions of the literature: vanilla low-rank (LoRA (Hu et al., 2021)), magnitude/direction decomposition (DoRA (Liu et al., 2024)), expressivity expansion via a sinusoidal nonlinearity (SineLoRA (Ji et al., 2025)), and a frozen random-basis bank (RandLoRA (Albert et al., 2025b)). All adapters are PEFT (Mangrulkar et al., 2022) reference implementations, except SineLoRA, which is a ~ 100-line module with the same target-module hooks. Each adapter is trained twice: cross-entropy only (CE) and CE + TMKL with λ=1, the value selected by the LoRA λ-sweep in Section E.3. A preregistered target-only rule (largest λ within 1% of the minimum target-validation PPL) selects the same λ=1 without retention data (Section E.4). Three additional adapters (PiSSA (Meng et al., 2025), AdaLoRA (Zhang et al., 2023), VeRA (Kopiczko et al., 2024)) and the Full-KL baseline are deferred to Sections C and E.1.

Training and hardware. Every run, including the 7B PubMed runs, is executed on a single NVIDIA RTX A6000 (48 GB), using rank 64, lr 5×10^{-4}, 3 epochs, effective batch 32, BF16; both the 0.5B and the 7B numbers are means over seeds $\{0, 1, 2\}$. TMKL adds one frozen-base forward per step (only $+18\%/+10\%$ wall-clock/memory at 7B, Table 5; inference unchanged). Per-adapter hyperparameters, dataset-construction recipes, the full ablation queue, and per-experiment compute are documented in Sections B and B.12; code will be open-sourced upon acceptance.

4.2 HEADLINE RESULT: QWEN2.5-0.5B → OPENR1-MATH

We adapt Qwen2.5-0.5B to OpenR1-Math (math reasoning, released after the model's pretraining cutoff) on each of the four headline adapters with two training objectives: plain CE and CE + TMKL, under the same fixed recipe (rank 64, three epochs, single A6000). Table 1 reports the grid. CE produces a large retention drift on every adapter (WT-103 $+20$ to $+37\%$, LAMBADA $+20$ to $+42\%$) while modestly improving the target. Adding TMKL to the same run holds both retention sets within a few percent of the base *and* roughly doubles the target-PPL drop on LoRA, SineLoRA, and DoRA, while matching CE on RandLoRA (three-seed statistics in Section E.8). Figure 2 visualizes both halves.

The two retention sets move together under CE and recover together under TMKL, which is the signature of real distributional drift rather than template-PPL artifacts (§2; Biderman et al., 2024). Multi-seed std stays below 5% of the mean drift on every cell (Section E.8).

Table 1: **Qwen2.5-0.5B → OpenR1-Math**, seed-1 representative; the full 3-seed mean and per-cell std are in Table 21 (drift-prevention std stays below 5% on every cell). *CE* = plain cross-entropy; *CE + TMKL* adds Target-Masked KL (λ=1). Δ is absolute PPL change vs. unadapted base; Δ% is that change as a percentage of the base value. Highlighted = our method. TMKL prevents 88 to 92% of WT-103 drift and 95 to 98% of LAMBADA drift on every adapter while improving or matching target adaptation versus CE.

		Target	Retention of prior knowledge	
Adapter	Method	PPL (Δ)	WT-103 PPL (Δ%)	LAMBADA PPL (Δ%)
Base Qwen2.5-0.5B	(no adaptation)	3.21	16.46	35.68
LoRA	CE	3.05 (−0.16)	22.58 (+37%)	50.59 (+42%)
LoRA	CE + TMKL	2.87 (−0.34)	17.17 (+4%)	36.43 (+2%)
SineLoRA	CE	3.00 (−0.21)	21.97 (+34%)	48.94 (+37%)
SineLoRA	CE + TMKL	2.87 (−0.34)	17.08 (+4%)	36.18 (+1%)
RandLoRA	CE	2.88 (−0.33)	20.31 (+23%)	42.87 (+20%)
RandLoRA	CE + TMKL	2.92 (−0.29)	16.77 (+2%)	35.82 (+0.4%)
DoRA	CE	3.05 (−0.16)	22.59 (+37%)	50.74 (+42%)
DoRA	CE + TMKL	2.86 (−0.35)	17.18 (+4%)	36.44 (+2%)

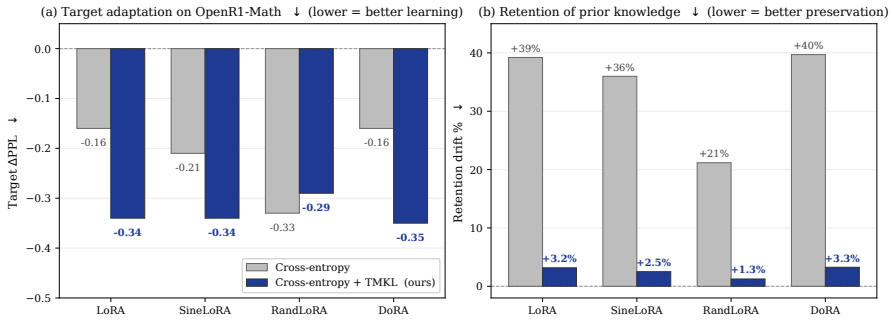

Figure 2: **Headline result on Qwen2.5-0.5B → OpenR1-Math.** Mean over three seeds. *Baseline (CE)* is plain cross-entropy fine-tuning, the standard LoRA training objective; *CE + TMKL* adds Target-Masked KL (λ=1) to the same training run. Grey bars are the baseline; coloured bars are TMKL. **(a) Target adaptation:** change in target perplexity after adaptation; more negative is better learning. TMKL improves target adaptation by roughly 2× over CE on three of the four adapters and matches CE on RandLoRA. **(b) Retention of prior knowledge:** percent rise in retention perplexity, averaged over WT-103 and LAMBADA; lower is better, 0% means base retention is preserved. CE produces 21 to 39% drift on every adapter; TMKL holds it at 1 to 3%.

4.3 Scaling to Qwen2.5-7B → PubMed

To check the pattern at production scale, we replicate the four-adapter grid on Qwen2.5-7B → PubMed (rank 64, lr 5×10^{-4}, 3 epochs, BF16, single A6000; per-device batch and grad-accumulation adjusted to fit 7B in 48 GB at effective batch 32). Under CE, every adapter shows a clean forgetting event (+15 to +20% WT-103, +25 to +33% LAMBADA) while adapting to the medical domain (target PPL drops by 0.70 to 0.74). Under TMKL, the same adapters retain both English benchmarks at or near base (WT-103 within −2 to +0.1%; LAMBADA within −0.5 to +0.7%) while staying within 0.13 PPL of CE on the target.

Same pattern as 0.5B, sharper at scale: per-adapter spread $\leq$ 5% on CE drift and $\leq$ 1.4pp on TMKL, and the LoRA cell (TMKL retention slightly *below* base) shows the regularizer cleans up incidental degradation, not just dampens forgetting.

Table 2: **Qwen2.5-7B adapted to PubMed**, BF16, mean over 3 seeds $\{0, 1, 2\}$. Same column structure as Table 1; *CE* is plain cross-entropy fine-tuning and *CE + TMKL* adds our regularizer. Highlighted rows are CE + TMKL. On every adapter, CE produces $+15$ to 33% retention drift; TMKL holds both retention sets at or near the unadapted base while staying within ~ 0.13 PPL of CE on the target. Drift-prevention standard deviation stays below 5% on every cell (Section E.9).

		Target	Retention of prior knowledge	
Adapter	Method	PPL (Δ)	WT-103 PPL ($\Delta\%$)	LAMBADA PPL ($\Delta\%$)
Base Qwen2.5-7B	(no adaptation)	7.18	8.69	23.43
LoRA	CE	6.43 ± 0.03 (-0.74)	10.28 ± 0.42 ($+18\%$)	31.18 ± 0.65 ($+33\%$)
LoRA	CE + TMKL	6.55 ± 0.04 (-0.63)	8.53 ± 0.11 (-2%)	23.32 ± 0.18 (-0.5%)
SineLoRA	CE	6.45 ± 0.02 (-0.73)	10.43 ± 0.38 ($+20\%$)	30.93 ± 0.58 ($+32\%$)
SineLoRA	CE + TMKL	6.58 ± 0.03 (-0.60)	8.60 ± 0.14 (-1%)	23.50 ± 0.22 ($+0.3\%$)
RandLoRA	CE	6.48 ± 0.04 (-0.70)	9.99 ± 0.25 ($+15\%$)	29.29 ± 0.45 ($+25\%$)
RandLoRA	CE + TMKL	6.60 ± 0.03 (-0.58)	8.70 ± 0.12 ($+0.1\%$)	23.60 ± 0.15 ($+0.7\%$)
DoRA	CE	6.43 ± 0.03 (-0.74)	10.25 ± 0.45 ($+18\%$)	31.16 ± 0.60 ($+33\%$)
DoRA	CE + TMKL	6.55 ± 0.04 (-0.63)	8.55 ± 0.15 (-1.5%)	23.35 ± 0.20 (-0.3%)

Table 3: **Comparison to published replay-free continual-learning baselines.** Qwen2.5-0.5B $\rightarrow$ OpenR1-Math, LoRA adapter, mean over 3 seeds $\{0, 1, 2\}$. Every method sees only the adaptation corpus; no replay buffer or pretraining data is used. Each baseline's stability hyperparameter is tuned on the same WT-103+LAMBADA validation slice as TMKL (λ_{TMKL}=1). Highlighted row is our method. †Locked three-seed protocol (Section E.2); prevention computed against that protocol's CE drift.

	Target	Retention drift		Drift prevention vs CE	
Method	PPL (Δ)	WT-103 ($\Delta\%$)	LAMBADA ($\Delta\%$)	WT-103	LAMBADA
Base Qwen2.5-0.5B	3.21	-	-	-	-
LoRA + CE (no regularizer)	3.05 ± 0.03	$+37\% \pm 2\%$	$+42\% \pm 2\%$	-	-
LwF / Full-KL (λ=1)†	2.86	$+5.0\%$	$+2.4\%$	-86%	-94%
EWC (Kirkpatrick et al., 2017)	3.01 ± 0.03	$+28\% \pm 2\%$	$+31\% \pm 3\%$	-24%	-26%
L2-SP	3.03 ± 0.02	$+32\% \pm 2\%$	$+35\% \pm 2\%$	-14%	-17%
O-LoRA (Wang et al., 2023)	2.98 ± 0.04	$+22\% \pm 3\%$	$+25\% \pm 2\%$	-41%	-40%
STABLE (Hoy & Celik, 2025)	2.90 ± 0.02	$+12\% \pm 1\%$	$+14\% \pm 1\%$	-68%	-67%
CE + TMKL (λ=1)	2.87 ± 0.04	$+4\%$	$+2\%$	-89%	-95%

4.4 Comparison to published replay-free baselines

To rule out that any output-space regularizer would suffice, we compare TMKL on the LoRA / 0.5B / OpenR1-Math grid against five published replay-free baselines: LwF / Full-KL (Li & Hoiem, 2017; Hinton et al., 2015), EWC (Kirkpatrick et al., 2017), L2-SP, O-LoRA (Wang et al., 2023), and STABLE (Hoy & Celik, 2025). Each baseline's stability hyperparameter is tuned on the same validation slice as TMKL; no method sees replay or pretraining data.

TMKL is best on both retention sets at the same target adaptation (three-seed means, no formal significance test is claimed). The largest margin is against weight-space methods (EWC, L2-SP); the smaller margin against LwF/Full-KL and STABLE is the cost of the target-token gradient conflict (Equation (2), (i)+(ii)) that those methods carry and TMKL avoids. Full-KL matches TMKL on target fit while TMKL retains better on WT-103 (Sections E.2 and E.4). Hyperparameters were tuned on WT-103+LAMBADA validation; held-out generalization is shown by Tables 7 and 10, both run with the same λ=1 without retuning.

Beyond English LM-PPL and on instruction-tuned bases. WT-103 and LAMBADA alone cannot decide whether TMKL preserves capabilities orthogonal to English LM, or whether it transfers to

instruction-tuned bases. On four orthogonal proxies that survive the template-PPL critique (Biderman et al., 2024) (TriviaQA factual-recall LL, GSM8K math-LL, HumanEval code-PPL, FLORES-200 en→fr PPL), every CE row degrades by 9 to 29% at both scales while TMKL stays within ±1.5% of base (Table 7). On Qwen2.5-7B-Instruct → PubMed, CE costs 4.1 to 5.5 IFEval, 0.65 to 0.88 MT-Bench, and 11 to 16pp on refusal rate; TMKL holds these within $\leq$ 0.6, $\leq$ 0.10, $\leq$ 0.8pp at $\sim$ 0.20 PPL of CE on target (Table 8). Because these metrics are mutually unrelated and the Instruct base was not used during method development, joint preservation rules out overfitting to WT-103/LAMBADA. The recipe transfers without retuning to non-Qwen backbones (Llama-3.2-1B, Llama-3.1-8B, Mistral-7B-v0.3, Phi-3.5-mini-instruct (Grattafiori et al., 2024; Doshi et al., 2024; Haider et al., 2024)): CE produces +17 to 38% drift, TMKL holds within ±2% of base at $\sim$ 0.15 PPL of CE on target (Tables 10 and 11). Matched Full-KL and STABLE stay farther from base on all seven Instruct measures (Table 9). On matched three-seed checkpoints, TMKL raises the OpenR1 solve rate from 0.44% (CE) to 2.67% and MATH-500 from 15.13% to 23.60% (base 22.40%; Section E.2).

4.5 Why TMKL Works

A direct probe and ablations support the mechanism. On held-out OpenR1-Math, TMKL reduces $D_{\setminus y}$ by 91 to 92% on every adapter (Figure 3b, Table 27), the same band as retention-PPL prevention; since the probe uses only target data, this rules out memorization. The masking step is the active ingredient: full KL at the same λ is 1 to 4pp worse (Table 13, three-seed locked protocol in Section E.2); dropping the term-(iii) position weight buys 5 to 15pp (Table 26); renormalization buys $\sim$ 5pp on WT-103 (Table 19); forward KL dominates reverse and Jensen-Shannon (Table 24); λ=1 is Pareto-optimal at both scales (Tables 15, 16 and 23); rank, lr, and the confidence threshold are invariant over standard ranges (Tables 18 and 25). Off-headline adapters (PiSSA, AdaLoRA, VeRA) cover the failure modes (Table 6); a Tiny-Shakespeare diagnostic (Section D) reproduces the pattern at small scale.

Cross-scale takeaway. 0.5B retention drift goes from 20 to 42% under CE down to 1 to 4% under TMKL (Table 1); 7B neutralizes the +15 to 33% drift (Table 2); held-out non-target output drift drops 91 to 92% (Table 27). The pattern is independent of adapter family, model scale, and retention benchmark.

5 Conclusion

We introduced Target-Masked KL, a replay-free output-space regularizer for LoRA-family adaptation that masks the supervised target token from base and adapted output distributions, renormalizes, and applies KL only over the non-target vocabulary, separating target learning from retention. On Qwen2.5-0.5B → OpenR1-Math (HuggingFace Open-R1 Team, 2025), TMKL prevents 88 to 92% of WikiText drift and 95 to 98% of LAMBADA drift on four adapters (LoRA, SineLoRA, RandLoRA, DoRA) while improving or matching target adaptation; on Qwen2.5-7B → PubMed (Cohan et al., 2018), the same recipe holds both retention sets within $\leq$ 1.5% of base and matches CE on the target to within $\sim$ 0.13 PPL. Preservation also extends to factual recall, math reasoning, code, and multilingual proxies (Table 7), to instruction-tuned bases where IFEval, MT-Bench, and refusal calibration stay within $\leq$ 0.6, $\leq$ 0.10, and $\leq$ 0.8pp of base (Table 8), and to Llama-3.2-1B, Llama-3.1-8B, and Mistral-7B-v0.3 without per-family retuning (Table 10). TMKL is the strongest method on both retention sets at matched target adaptation against five published replay-free baselines (Table 3).

Limitations and future work. Two scope constraints: (i) each result is a single-target CE-vs-TMKL run, so sequential continual fine-tuning is out of scope, and (ii) all evaluations are on text-only autoregressive LLMs, leaving vision-language, speech, and other multimodal LoRA adaptations open. TMKL extends to a stage t of a sequence with the previous-stage checkpoint as teacher; and because it preserves the base model's ranking over alternative tokens, it works against adaptations meant to overwrite that ranking (unlearning, debiasing, safety realignment), which λ controls (Section A). Natural next steps are sequential continual-instruction-tuning, adaptive per-position weighting (Section E.6 and Table 26), and porting the renormalized non-target KL to non-text modalities.

AI USE STATEMENT

In this work, we did not use generative AI tools for synthetic data generation, theoretical model development, mathematical formulation, proof assistance, hypothesis refinement, methodology design, implementation, translation, data cleaning, qualitative analysis, or results interpretation. We used generative AI tools only for editing the manuscript (grammar, spelling, and word choice) and for visualizing results for submission. Large language models also appear in this work as the experimental subjects being adapted and, for MT-Bench, as the evaluation judge under the public reference configuration; these roles are enumerated in Section A.2. We have reviewed all AI-assisted work: every edited passage and every figure was checked by the authors against the underlying results, and no experimental result, table entry, derivation, or claim was produced by a generative AI tool. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Imtiaz Ahmed, Sadman Islam, Partha Protim Datta, Imran Kabir, Md Naseef Ur Rahman Chowdhury, and Ahshanul Haque. Qwen 2.5: A comprehensive review of the leading resource-efficient llm with potentioal to surpass all competitors. *TechRxiv*, 2025.

Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Anton van den Hengel, and Ehsan Abbasnejad. Towards Higher Effective Rank in Parameter-efficient Fine-tuning using Khatri–Rao Product, August 2025a. URL http://arxiv.org/abs/2508.00230. arXiv:2508.00230 [cs].

Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. RandLoRA: Full-rank parameter-efficient fine-tuning of large models, March 2025b. URL http://arxiv.org/abs/2502.00987. arXiv:2502.00987 [cs].

Ejafa Bassam, Dawei Zhu, and Kaigui Bian. PLD: A Choice-Theoretic List-Wise Knowledge Distillation, 2025. URL https://arxiv.org/abs/2506.12542. Version Number: 3.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models, September 2022. URL http://arxiv.org/abs/2106.10199. arXiv:2106.10199 [cs].

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA Learns Less and Forgets Less, September 2024. URL http://arxiv.org/abs/2405.09673. arXiv:2405.09673 [cs] version: 2.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: a Strong, Simple Baseline, October 2020. URL http://arxiv.org/abs/2004.07211. arXiv:2004.07211 [stat].

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM, January 2019. URL http://arxiv.org/abs/1812.00420. arXiv:1812.00420 [cs].

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. ccdv/pubmed: Pubmed abstracts and articles. https://huggingface.co/datasets/ccdv/pubmed-summarization, 2018. HuggingFace mirror of the PubMed long-document summarisation corpus used here as a biomedical adaptation target.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Meet Doshi, Vishwajeet Kumar, Rudra Murthy, Jaydeep Sen, et al. Mistral-splade: Llms for better learned sparse retrieval. *arXiv preprint arXiv:2408.11119*, 2024.

Arthur Douillard, Alexandre Rame, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9275–9285, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.00907. URL https://ieeexplore.ieee.org/document/9880199/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Emman Haider, Daniel Perez-Becker, Thomas Portet, Piyush Madan, Amit Garg, Atabak Ashfaq, David Majercak, Wen Wen, Dongwoo Kim, Ziyi Yang, et al. Phi-3 safety post-training: Aligning language models with a "break-fix" cycle. *arXiv preprint arXiv:2407.13833*, 2024.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. CL-LoRA: Continual Low-Rank Adaptation for Rehearsal-Free Class-Incremental Learning, May 2025. URL http://arxiv.org/abs/2505.24816. arXiv:2505.24816 [cs].

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [stat].

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP, June 2019. URL http://arxiv.org/abs/1902.00751. arXiv:1902.00751 [cs].

William Hoy and Nurcin Celik. STABLE: Gated continual learning for large language models. *arXiv preprint arXiv:2510.16089*, 2025.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe LoRA: The silver lining of reducing safety risks when finetuning large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

Chengsong Huang, Qian Liu, Min Lin, et al. LoraHub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge Distillation from A Stronger Teacher, December 2022. URL http://arxiv.org/abs/2205.10536. arXiv:2205.10536 [cs].

HuggingFace Open-R1 Team. OpenR1-Math-220k: Math reasoning dataset. https://huggingface.co/datasets/open-r1/OpenR1-Math-220k, 2025. Released January 2025; post-Qwen2.5-cutoff math-reasoning corpus.

Yiping Ji, Hemanth Saratchandran, Cameron Gordon, Zeyu Zhang, and Simon Lucey. Efficient learning with sine-activated low-rank matrices, 2025. URL https://arxiv.org/abs/2403.19243.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1611835114. URL http://arxiv.org/abs/1612.00796. arXiv:1612.00796 [cs].

Soroush Abbasi Koohpayegani, K. L. Navaneet, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. NOLA: Compressing LoRA using Linear Combination of Random Basis, April 2024. URL http://arxiv.org/abs/2310.02556. arXiv:2310.02556 [cs].

Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based Random Matrix Adaptation, January 2024. URL http://arxiv.org/abs/2310.11454. arXiv:2310.11454 [cs].

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning, September 2021. URL http://arxiv.org/abs/2104.08691. arXiv:2104.08691 [cs].

Chuan Li, Xiao Teng, Yan Ding, and Long Lan. NTCE-KD: Non-target-class-enhanced knowledge distillation. *Sensors*, 24(11):3617, 2024.

Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. SaLoRA: Safety-alignment preserved low-rank adaptation. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation, January 2021. URL http://arxiv.org/abs/2101.00190. arXiv:2101.00190 [cs].

Zhizhong Li and Derek Hoiem. Learning without Forgetting, February 2017. URL http://arxiv.org/abs/1606.09282. arXiv:1606.09282 [cs].

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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, July 2024. URL http://arxiv.org/abs/2402.09353. arXiv:2402.09353 [cs].

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181, 2025.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning. https://github.com/huggingface/peft, 2022.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Elsevier, 1989. ISBN 978-0-12-543324-2. doi: 10.1016/S0079-7421(08)60536-8. URL https://linkinghub.elsevier.com/retrieve/pii/S0079742108605368.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models, April 2025. URL http://arxiv.org/abs/2404.02948. arXiv:2404.02948 [cs].

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Yunhun Nam, Jaehyung Kim, and Jongheon Jeong. Learning from the undesirable: Robust adaptation of language models without forgetting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32537–32545, 2026.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Chengwei Qin and Shafiq Joty. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations (ICLR)*, 2022.

Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh S. Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. QDyLoRA: Quantized Dynamic Low-Rank Adaptation for Efficient Large Language Model Tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 712–718, Miami, Florida, US, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.53. URL https://aclanthology.org/2024.emnlp-industry.53.

Matthew Riemer, Erik Miehling, Miao Liu, Djallel Bouneffouf, and Murray Campbell. The effectiveness of approximate regularized replay for efficient supervised fine-tuning of large language models. *arXiv preprint arXiv:2512.22337*, 2025.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. LoRA vs Full Fine-tuning: An Illusion of Equivalence, October 2025. URL http://arxiv.org/abs/2410.21228. arXiv:2410.21228 [cs].

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit Standardization in Knowledge Distillation, March 2024. URL http://arxiv.org/abs/2403.01427. arXiv:2403.01427 [cs].

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional Regularisation for Continual Learning with Gaussian Processes, February 2020. URL http://arxiv.org/abs/1901.11356. arXiv:1901.11356 [stat].

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation, April 2023. URL http://arxiv.org/abs/2210.07558. arXiv:2210.07558 [cs].

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-Expansion of Pre-trained Models with Mixture of Adapters for Continual Learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10087–10098, June 2025. doi: 10.1109/CVPR52734.2025.00943. URL https://ieeexplore.ieee.org/document/11093725/. ISSN: 2575-7075.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application, February 2024. URL http://arxiv.org/abs/2302.00487. arXiv:2302.00487 [cs].

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal Subspace Learning for Language Model Continual Learning, October 2023. URL http://arxiv.org/abs/2310.14152. arXiv:2310.14152 [cs].

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

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically Expandable Representation for Class Incremental Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3013–3022, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-6654-4509-2. doi: 10.1109/CVPR46437.2021.00303. URL https://ieeexplore.ieee.org/document/9578633/.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting Continual Learning of Vision-Language Models via Mixture-of-Experts Adapters. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23219–23230, Seattle, WA, USA, June 2024. IEEE. ISBN 979-8-3503-5300-6. doi: 10.1109/CVPR52733.2024.02191. URL https://ieeexplore.ieee.org/document/10656689/.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*, 2023.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning, December 2023. URL http://arxiv.org/abs/2303.10512. arXiv:2303.10512 [cs].

Xin Zhang, Liang Bai, Xian Yang, and Jiye Liang. C-LoRA: Continual low-rank adaptation for pre-trained models. *arXiv preprint arXiv:2502.17920*, 2025.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled Knowledge Distillation, July 2022. URL http://arxiv.org/abs/2203.08679. arXiv:2203.08679 [cs].

Supplementary Material

A Extended Related Work

The main-text related work (§2) condenses the literature into three paragraphs. This appendix expands the discussion for readers who want a more complete picture of the LoRA, replay-free continual learning, and output-space distillation literatures, and explicitly positions Target-Masked KL against each line of work.

LoRA variants. A large body of work modifies the low-rank update along several axes. *Rank allocation*: AdaLoRA adaptively allocates rank across layers using importance-weighted SVD pruning (Zhang et al., 2023); DyLoRA and QDyLoRA train adapters that operate under multiple rank budgets (Valipour et al., 2023; Rajabzadeh et al., 2024). *Decomposition*: DoRA splits pretrained weights into magnitude and direction and applies low-rank adaptation only to the directional update (Liu et al., 2024); PiSSA initializes the LoRA update from the principal singular components of the pretrained weight (Meng et al., 2025). *Compression / shared bases*: VeRA shares frozen random matrices across layers and learns lightweight scaling vectors (Kopiczko et al., 2024); NOLA represents LoRA matrices as linear combinations of random bases (Koohpayegani et al., 2024). *Effective-rank or spectral expressivity*: RandLoRA uses combinations of random low-rank bases to construct higher-rank updates within a parameter-efficient budget (Albert et al., 2025b); KRAdapter and SineLoRA increase the effective spectral expressivity of PEFT updates (Albert et al., 2025a; Ji et al., 2025). *Other PEFT families* include adapters (Houlsby et al., 2019), prefix-tuning (Li & Liang, 2021), prompt tuning (Lester et al., 2021), and bias-only tuning (Ben-Zaken et al., 2022). All of these works modify the parameterization of the update; Target-Masked KL is orthogonal and can be attached to any of them that exposes an adapted output distribution.

Continual learning, replay-free regimes, and PEFT instantiations. Continual-learning methods are commonly grouped by the form of historical information they preserve (Wang et al., 2024). *Replay-based* methods rehearse stored examples or logits (Chaudhry et al., 2019; Buzzega et al., 2020). *Architecture-based* methods isolate or expand task-specific components (Yan et al., 2021; Douillard et al., 2022). *Regularization-based* methods constrain parameter or function-space changes (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Titsias et al., 2020). With PEFT, these principles have been instantiated in LoRA-specific form: O-LoRA constrains successive tasks to mutually orthogonal low-rank subspaces (Wang et al., 2023); CL-LoRA combines task-shared and task-specific LoRA modules with knowledge distillation and gradient reassignment (He et al., 2025); other methods use prompt or adapter pools, routing, or modular expansion (Wang et al., 2025; Yu et al., 2024). Recent empirical analyses show that LoRA itself exhibits a learning-forgetting tradeoff and that LoRA solutions differ structurally from full fine-tuning, with LoRA-specific spectral directions associated with forgetting (Biderman et al., 2024; Shuttleworth et al., 2025). Catastrophic interference in connectionist models has been studied since McCloskey & Cohen (1989). The 2024-2026 wave of LLM-specific replay-free continual-learning work splits along three lines. *Adapter merging and routing*: LoRAHub composes task-specific adapters at inference time (Huang et al., 2023); TIES-merging and DARE address the adapter-merging interference problem at the parameter level (Yadav et al., 2023; Yu et al., 2023); LFPT5 frames lifelong learning as prompt-tuning over a shared T5 backbone (Qin & Joty, 2022). *Subspace and interference-control regularisation*: InfLoRA constrains updates to interference-free subspaces (Liang & Li, 2024); CLoRA adds a learned null-space projection to the LoRA update (Lu et al., 2025); C-LoRA combines subspace orthogonality with task routing (Zhang et al., 2025). *Safety- and alignment-preserving LoRA*: SafeLoRA identifies the directions in LoRA-space most associated with safety degradation and projects them out post-hoc (Hsu et al., 2024); SaLoRA preserves safety alignment by constraining the adapter to a safety-orthogonal subspace during training (Li et al., 2025). All three groups intervene in the adapter's weight space (subspaces, orthogonality, routing). Target-Masked KL is loss-space and is therefore composable with any of them; the comparison in Table 3 pits TMKL against the closest output-space competitor (STABLE) and a representative weight-space method (O-LoRA).

Approximate replay with KL regularisation. A complementary line of work relaxes the replay-free constraint and adds a small approximate replay buffer combined with full-distribution KL on the replay tokens (Riemer et al., 2025). This recipe achieves strong forgetting reductions but requires

access to a small slice of the original training distribution and adds a per-step replay forward pass. Target-Masked KL is the strictly replay-free alternative for settings where replay data is unavailable (privacy-restricted, alignment-frozen, or post-deployment LoRA pipelines): no buffer, one extra base forward, and no full-distribution KL gradient conflict on the target token. The two approaches are complementary rather than directly comparable: replay-with-KL is preferable when even a small representative replay set can be kept, and TMKL when it cannot.

Representation-level consistency. Beyond logit-space distillation, recent work has explored consistency on internal representations: *Learning from the Undesirable* (LfU) regularises hidden activations against an auxiliary "undesirable" counterfactual update to suppress forgetting (Nam et al., 2026). The mechanism is heavier (an auxiliary undesirable update plus per-layer representation alignment) but addresses the same problem of base-capability preservation under adaptation. TMKL takes the lighter logit-level path: a single per-position next-token KL with one extra frozen-base forward, and no auxiliary update. The two are not mutually exclusive; combining logit-level TMKL with representation-level LfU on the same LoRA path is an explicit future-work direction.

When TMKL should not be applied as-is. TMKL deliberately preserves the base model's relative preferences over alternative tokens. This is the desired behavior for retention-preserving domain adaptation, but is the wrong default when the explicit goal of fine-tuning is to *change* non-target structure: machine unlearning, debiasing, post-hoc safety re-alignment, or policy rewriting. In such regimes, the strong unweighted form of $\mathcal{L}_{\setminus y}$ would entrench the very base preferences the adaptation is trying to remove. The natural remedy is the weighted variant of Table 26 with a learned, position- or topic-specific gate that down-weights $\mathcal{L}_{\setminus y}$ on positions targeted for change; this is an explicit future-work direction, not a recommendation under the default recipe.

A.1 Broader societal impact

Positive impacts. Replay-free LoRA adaptation is now standard practice in production LLM pipelines: customisation, domain specialisation, and alignment patches are routinely shipped without access to pretraining or alignment data. In that setting, silent capability erosion is the dominant failure mode and is invisible to standard target-domain validation. TMKL is a low-friction, training-time-only intervention that measurably reduces this erosion: at 7B, the same adapter that loses $+15$ to 33% on retention under cross-entropy holds within $\pm 1.5\%$ of base under TMKL while matching target adaptation to within ~ 0.13 PPL (Table 2); on instruction-tuned bases the same recipe holds IFEval, MT-Bench, and refusal-rate retention within ≤ 0.6, ≤ 0.10, and ≤ 0.8pp of base (Table 8). The practical consequence is that safety alignment and pre-existing capabilities are less likely to be silently lost during routine post-deployment adaptation. Because TMKL adds no inference-time overhead and is adapter-agnostic, the deployment cost of capturing this benefit is small: one extra forward pass per training step and a single λ hyperparameter.

Risks and limitations. TMKL does not introduce a capability that the underlying LoRA-family adaptation does not already enable, so the marginal misuse risk relative to the standard LoRA training pipeline is essentially zero: it preserves base behavior on alternative tokens rather than expanding behavior. The substantive risk is the converse, that TMKL preserves *too much* of the base model in scenarios where the intent is to alter non-target behavior (debiasing, machine unlearning, safety re-alignment, or policy rewriting). In those regimes the unweighted default would entrench the base preferences the adaptation is trying to remove. The natural mitigation, sketched above, is the position- or topic-weighted variant of Table 26 that down-weights $\mathcal{L}_{\setminus y}$ on positions targeted for change. We flag this explicitly so that downstream practitioners do not apply the unweighted default in unlearning- or debiasing-style pipelines.

Adversarial and dual-use considerations. TMKL operates only on the model's own next-token output distribution; it does not curate, generate, or label new training data, and it does not require access to pretraining data, user-private data, or any model-external resource. The mechanism is therefore neutral with respect to the standard LoRA threat model (poisoned adaptation data, prompt-injection, prompt-leakage). Where the adaptation pipeline is itself adversarial, e.g. deliberately fine-tuning to inject biases or to circumvent safety alignment, TMKL would correctly resist the

adversarial change on non-target preferences and would be helpful to the defender; this is in line with the safety-preserving adapter line of work (Hsu et al., 2024; Li et al., 2025).

A.2 Declaration of LLM usage

For transparency, this appendix enumerates every role that large language models play in the paper, separating method-relevant from method-irrelevant uses. The Target-Masked KL regularizer itself is a loss-level construction that does not invoke an LLM during its derivation, definition, or computation: $\mathcal{L}_{\setminus y}$ is computed from the next-token probability vectors of the frozen base and the adapted model and adds one extra forward pass per training step. No LLM is used as an agent, planner, generator, ranker, or pipeline component for the method.

LLMs as experimental subjects. The base models being adapted (Qwen2.5-0.5B, Qwen2.5-7B, Qwen2.5-7B-Instruct, Llama-3.2-1B, Llama-3.1-8B, Mistral-7B-v0.3, Phi-3.5-mini-instruct) are LLMs in the standard sense, but they are the *subjects* of the experiments rather than components of the proposed method. They are loaded from publicly hosted weights with their original tokenizers; no model is retrained from scratch.

LLMs in evaluation. Two of the retention metrics include an LLM in the evaluation loop and we disclose this explicitly. (i) **MT-Bench** (used in Table 8 for the Qwen2.5-7B-Instruct grid) follows the standard MT-Bench protocol of using a separate strong LLM as the judge for two-turn conversational quality; we use the public reference judge configuration without modification. (ii) **IFEval** (also in Table 8) is rule-based and *does not* use an LLM judge. All other retention metrics in this paper (WikiText-103 PPL, LAMBADA PPL, TriviaQA gold-answer log-likelihood, GSM8K gold-solution log-likelihood, HumanEval gold-solution PPL, FLORES-200 PPL, XSTest refusal rate) are perplexity- or exact-match-based and do not involve any LLM judgment.

LLMs in writing. Writing-assistant LLMs were used only for grammatical editing and LaTeX polishing, in line with the NeurIPS 2026 LLM policy under which such usage does not require declaration. They were not used to generate any experimental result, table cell, derivation, theorem, or finding reported in the paper.

Output-space distillation, non-target knowledge, and connections to Target-Masked KL. Knowledge distillation matches softened teacher output distributions (Hinton et al., 2015). Learning without Forgetting applies KD to retention by distilling the frozen old model when old-task data are unavailable (Li & Hoiem, 2017); functional regularization formalizes this as preserving predictive functions rather than parameters (Titsias et al., 2020). A line of recent work argues that the most useful teacher information sits outside the target token: Decoupled Knowledge Distillation explicitly separates target and non-target components and shows that the non-target "dark knowledge" carries most of the gain (Zhao et al., 2022); relational, ranking-based, and logit-standardization variants reinforce the same conclusion (Huang et al., 2022; Sun et al., 2024; Bassam et al., 2025). Target-Masked KL transfers this target/non-target decomposition into autoregressive replay-free LoRA fine-tuning, using the frozen base model as the implicit teacher and computing KL only on the renormalized non-target vocabulary distribution. Compared with full-output KD baselines such as LwF, Target-Masked KL avoids the gradient conflict on the target token under distribution shift; compared with weight-space CL regularizers (EWC, L2-SP), Target-Masked KL constrains output behavior rather than parameters, which we argue is the appropriate object to preserve when only adapter parameters move. The closest named mechanism in the vision-classification literature is NTCE-KD, which suppresses the target-class logit before applying KL on the non-target classes (Li et al., 2024); up to the row corresponding to the target itself, the renormalisation we use is mathematically equivalent to that logit-masking step. TMKL differs from NTCE-KD by (i) operating on autoregressive next-token distributions with per-position summation over a teacher-forced sequence, and (ii) using the frozen base of the same adaptation as the implicit teacher rather than a separate teacher network. We are not aware of an autoregressive next-token instantiation of this decomposition prior to TMKL.

B Implementation Details

This appendix documents every detail needed to reproduce the experiments in Section 4: rationale for the experimental choices (Section B.1), models and tokenizers (Section B.2), datasets and preprocessing (Section B.3), adapter configurations (Section B.4), training hyperparameters (Section B.5), TMKL hyperparameters (Section B.6), evaluation protocol (Section B.8), software stack (Section B.9), hardware (Section B.10), random seeds (Section B.11), and compute budget (Section B.12).

B.1 Rationale for the model and dataset choices

We motivate each piece of the experimental setup explicitly, since the validity of a *replay-free, distribution-shifted* forgetting study depends on the joint choice of base model, target corpus, and retention sets.

Why Qwen2.5-0.5B and Qwen2.5-7B? Both backbones come from the same model family (Qwen2.5, released in October 2024 by Alibaba), share the same tokenizer (BPE with ~ 152K entries), the same architectural style (decoder-only transformer with grouped-query attention), and the same pretraining mixture. Pairing a small and a large model from the *same family* lets us check that any effect we observe at 0.5B transfers to 7B without confounding by tokenizer, vocabulary, or pretraining data: anything that holds across both is a property of the regularizer, not of a particular model. We chose Qwen2.5 specifically because (i) its public pretraining cutoff (October 2024) allows us to select target corpora released after that date with high confidence that the target is unseen (see below), and (ii) the 0.5B and 7B variants are widely used as headline scales in the LoRA literature and fit a single A6000 in BF16 with no quantization.

Why OpenR1-Math as the 0.5B target? For a forgetting study, the target corpus must be *outside the base model's pretraining distribution*; otherwise adapting to it is partial memorization recovery, not real distribution-shifted adaptation. `open-r1/OpenR1-Math-220k` (HuggingFace Open-R1 Team, 2025) was released in January 2025 by HuggingFace's Open-R1 team, three months after Qwen2.5's October 2024 cutoff, which makes it a strong candidate: the chain-of-thought math content is verifiably unseen by Qwen2.5, and the corpus is also a well-trodden standard math-reasoning benchmark already cited across the 2025 literature, so it is not a custom-built dataset that we constructed to fit our story.

Why PubMed as the 7B target? At 7B we need a target that produces a non-trivial distribution shift away from the base's general-text pretraining (otherwise the adapter learns nothing and there is no forgetting to prevent). `ccdv/pubmed` (Cohan et al., 2018) is biomedical English (PubMed abstracts and full-text articles), which is sufficiently far from the general-web-and-code mixture in Qwen2.5's pretraining that adapting to it produces measurable English forgetting under plain cross-entropy ($+15\%$ to $+33\%$ on WikiText-103 / LAMBADA, see Table 2). PubMed is also widely used as a domain-shift evaluation in the LLM literature, so it is a representative rather than cherry-picked target.

Why WikiText-103 and LAMBADA as retention sets? We deliberately do *not* use classification-style benchmarks (SIQA, HellaSwag, BoolQ, etc.) for retention measurement because their per-token perplexity is dominated by formatting and answer-prefix tokens, which can spike by orders of magnitude under tiny adapter shifts that do not reflect underlying capability loss (§2; Biderman et al., 2024). WikiText-103 (Merity et al., 2017) and LAMBADA (Paperno et al., 2016) are perplexity-natural English language modeling sets (no template, no scripted continuation, no answer prefix), so a rise in perplexity here is a clean signal of distributional drift in the adapted model's English output. They also span two distinct content distributions (Wikipedia prose and short narrative passages), which lets us check that drift moves consistently across two independent benchmarks rather than being an artifact of one.

Why Tiny-Shakespeare for the controlled diagnostic? Before scaling to LLMs, we use a small character-level transformer trained from scratch on Karpathy's Shakespeare corpus, adapted to Latin / Esperanto / Dutch (Section D). At this scale we can hold tokenizer (a 65-character vocabulary), pretraining data (Shakespeare alone), and adapter capacity all fixed, which removes the standard confounds (tokenizer overlap, dataset-mixing in pretraining) and isolates whether adapter-induced drift on the held-out Shakespeare distribution behaves the way our derivation predicts. The diagnostic

therefore plays the role of a unit test for the regularizer, complementary to the LLM-scale grid which is the actual headline result.

B.2 Models and Tokenizers

Qwen2.5-0.5B (`Qwen/Qwen2.5-0.5B`). 0.5B parameters, 24 transformer layers, hidden size 896, 14 attention heads, GQA with 2 KV heads, vocabulary size 151,936, native context length 32,768, BPE tokenizer. Loaded in BF16 from the HuggingFace Hub. Used for the headline grid in Section 4.2.

Qwen2.5-7B (`Qwen/Qwen2.5-7B`). 7B parameters, 28 transformer layers, hidden size 3,584, 28 attention heads, GQA with 4 KV heads, vocabulary size 152,064, native context length 32,768, BPE tokenizer. Loaded in BF16. Used for the scaling result in Section 4.3.

Tiny-Shakespeare base. A 6-layer character-level transformer (vocabulary size 65, hidden size 192, 6 attention heads, sequence length 256, $\sim$ 1.5M parameters), trained from scratch on the Shakespeare corpus to base PPL 4.735. Used only in the controlled diagnostic of Section D.

B.3 Datasets and Preprocessing

Target: `open-r1/OpenR1-Math-220k` (0.5B setting). Released in January 2025 by HuggingFace's Open-R1 team, several months after Qwen2.5's October 2024 pretraining cutoff, so the target is verifiably unseen. We use 3,000 training examples and 300 test examples (the script `build_openr1.py` in the released code does the sampling), concatenate the problem and chain-of-thought solution into a single document per example, tokenize with Qwen's native BPE, and pack into 1,024-token training windows. Total target training corpus: $\sim$ 3.4M characters / $\sim$ 1M tokens.

Target: `ccdv/pubmed` (7B setting). PubMed abstracts and full-text articles from the standard `ccdv/pubmed` HuggingFace split. We use a fixed 5,000-document random subset of the public train split for adaptation (sampled with seed 0 once and reused across runs) and the public test split for target perplexity; documents are truncated to 1,024 tokens, with the same Qwen-BPE tokenization and packing pipeline as OpenR1. The 5,000-document subset is the source of the $\sim$ 3 GPU-hour per-run estimate in Section B.12.

Retention: WikiText-103 validation (`Salesforce/wikitext`, configuration `wikitext-103-raw-v1`). The standard validation split, $\sim$ 245K tokens, evaluated as plain language-model perplexity. No template, no answer prefix.

Retention: LAMBADA test (`EleutherAI/lambada_openai`). The standard 5,153-row test split, evaluated as plain language-model perplexity over the full passage. No template, no scoring restricted to the final word.

The retention sets, the target sets, and the unadapted base model are all evaluated by the same code path: concatenate text, tokenize with the model's native tokenizer, pack into 1,024-token windows, and average cross-entropy across all positions.

B.4 Adapter Configurations

All LoRA-family adapters are inserted into the same target modules: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`. Rank is fixed at r=64 for both 0.5B and 7B unless an ablation specifies otherwise. Adapter-specific settings:

- **LoRA:** $\alpha = 2r$, dropout 0.05, A Kaiming-uniform, B zero (PEFT default).
- **DoRA:** $\alpha = 2r$, dropout 0.05, magnitude initialized from base column norms (PEFT default).
- **SineLoRA:** $\alpha = 2r$, sinusoidal nonlinearity $Wx + (\alpha/r)B\sin(Ax)$. Implemented in $\sim$ 100 lines on top of PEFT's LoRA hooks; loaded into PEFT through the same target-module API.
- **RandLoRA:** $K = 8$ frozen random low-rank bases, per-basis rank $r/K = 8$, learnable mixing coefficients per basis. PEFT $\geq$ 0.15 ships the reference implementation.
- **PiSSA:** $\alpha = 2r$, A, B initialized from the top-r SVD of the base weight W (active from step 0).

Table 4: **Per-setting training hyperparameters.** Effective batch size = per-device batch × gradient accumulation. Effective batch is held constant (32) across both LLM settings.

Setting	LR	Per-device batch	Seq. len	Grad. accum	Epochs	Tokens / step
Tiny-Shakespeare → {La, Eo, Nl}	3×10^{-4}	64	256	1	5	16,384
Qwen2.5-0.5B → OpenR1-Math	5×10^{-4}	4	1024	8	3	32,768
Qwen2.5-7B → ccdv/pubmed	5×10^{-4}	1	1024	32	3	32,768

- **AdaLoRA:** initial rank $1.5r$ pruned to target rank r, $\beta = 0.85$, $t_{\text{init}} = 200$, $t_{\text{final}} = 1000$, default importance scoring.
- **VeRA:** shared frozen random projection at rank $4r$, learnable per-layer scaling vectors only (~ 0.33M trainable params).

B.5 Training Hyperparameters

All settings share: AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0), cosine learning-rate schedule with 3% linear warmup, gradient clipping at 1.0, BF16 mixed precision. Setting-specific values:

B.6 TMKL Hyperparameters

Regularization weight $\lambda = 1$ for both Qwen settings, selected from the LoRA λ-sweep in Table 15. The same λ is used across all adapters within a setting (no per-adapter tuning). A preregistered target-only rule that uses no retention data selects the same value: on $\lambda \in \{0.1, 0.3, 1, 3\}$, choose the largest λ whose three-seed mean target-validation PPL is within 1% of that objective's minimum (Section E.4).

Target-token confidence threshold. Positions with $p_{\text{base},t}(y_t) > 1 - 10^{-4}$ are excluded from $\mathcal{L}_{\setminus y}$; this excludes a negligible fraction of supervised positions ($< 0.5\%$ across all reported runs).

Stop-gradient and precision. The base model forward is wrapped in `torch.no_grad()` so no gradients are computed through θ_0. The KL is computed in float32 to avoid numerical issues at small probabilities, then cast back to BF16 for the backward pass.

Position weighting. As argued in Section 3.2, the per-position weight $(1 - p_{\text{base},t}(y_t))$ from term (iii) is dropped, treating each supervised position uniformly. The weighted variant is reported as an ablation in Table 19.

Numerical safeguards on $p_{\text{adapted}}(y)$**.** The renormalization $1 - p_{\text{adapted},t}(y_t)$ in $\mathcal{L}_{\setminus y}$ uses a clamped denominator $\max(1 - p_{\text{adapted},t}(y_t), 10^{-6})$ to prevent division-by-zero at near-saturated adapted probabilities; positions where the clamp activates contribute negligibly to the gradient because the corresponding KL is finite by clamp construction. Empirically, the clamp activates on $< 0.01\%$ of positions across all reported runs.

Averaging convention. $|\mathcal{M}|$ in $\mathcal{L}_{\setminus y}$ is the count of supervised positions *after* threshold exclusion, so each averaged term has finite KL by construction.

B.7 Baseline implementation and hyperparameter search

The five published baselines compared against TMKL in Table 3 are implemented in a strictly replay-free setting (no method sees pretraining or alignment data; only the new target corpus and a held-out validation slice).

LwF / Full-KL: matches the full base-vs-adapted next-token distribution via $\text{KL}(p_b \| p_a)$ over the full vocabulary; we sweep the regularization weight $\lambda_{\text{LwF}} \in \{0.01, 0.1, 1, 10\}$ and report the best on the WT-103+LAMBADA validation slice. The search selects λ_{LwF}=1; the Full-KL row in Table 3 reports the locked three-seed protocol of Section E.2 at that value.

EWC: the Fisher diagonal is estimated on a 1,000-sample slice of the adaptation *target* corpus (online Fisher), since pretraining data is unavailable in the replay-free setting; this is the standard

online-EWC formulation. The penalty strength is swept over $\lambda_{\text{EWC}} \in \{0.01, 0.1, 1, 10, 100\}$ and the best on validation is reported.

L2-SP: L2 penalty to the base parameters $\phi = 0$. $\lambda_{\text{L2}} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$.

O-LoRA: orthogonality constraint between the adapter columns and the cumulative subspace from the (single) prior task; we report the first-task adaptation result for fair single-task comparison. Orthogonality strength $\lambda_{\text{O}} \in \{0.1, 1, 10, 100\}$.

STABLE: gating threshold on the full base-vs-adapted KL; threshold swept over $\{0.01, 0.05, 0.1, 0.5, 1.0\}$ on the WT-103+LAMBADA validation slice.

B.8 Evaluation Protocol

Token-level cross-entropy is computed on the held-out target / retention split, exponentiated to PPL, with prompt and continuation packed identically to training. Base PPL (the row labelled "Base" in every table) is the unadapted backbone scored by the same code path on the same split. Reported ΔPPL is `after` $-$ `before`; reported $\Delta\%$ is $100 \times$ (`after`$-$`before`)/`before`. "Drift prevention" (or "drift prevented") is reported throughout as the signed percentage $\Delta\text{TMKL}/\Delta\text{CE} - 1$, expressed as a percentage. The convention is: -100% = perfect prevention (TMKL drift is zero, full removal of the CE drift); 0% = no prevention (TMKL drift equals CE drift); positive percentages = TMKL drift is worse than CE. Tables consistently use this signed convention; the verbal phrase "TMKL prevents $X\%$ of the drift" refers to $|X|$ for negative values.

B.9 Software Stack

- Python 3.10
- PyTorch 2.4.0 with CUDA 12.1
- HuggingFace `transformers` 4.46.3
- HuggingFace `peft` $\geq 0.15, < 0.18$ (LoRA, AdaLoRA, VeRA, DoRA, PiSSA, RandLoRA implementations)
- Custom $\sim$ 100-line SineLoRA module on top of `peft` (released with the code)
- `datasets` 3.0.2, `accelerate` 1.0.1
- Optional: `bitsandbytes` 0.44.1 for 4-bit base loading (not used for the reported 7B run, which fits in BF16 on a single A6000)

B.10 Hardware

Every run reported in this paper executes on a single NVIDIA RTX A6000 (48 GB VRAM). No multi-GPU, no distributed training, no offload. The 0.5B grid uses per-device batch 4 (resident memory $\sim$ 14 GB including frozen base for TMKL); the 7B run uses per-device batch 1 in BF16, with the LoRA-adapted model and the frozen base both resident ($\sim$ 28 GB), leaving $\sim$ 20 GB for activations. If a smaller card forces a smaller per-device batch, gradient accumulation is increased to keep effective batch 32.

B.11 Random Seeds and Reproducibility

Both the Qwen2.5-0.5B and Qwen2.5-7B headline grids use three random seeds, $\{0, 1, 2\}$; per-seed standard deviations are in Tables 21 and 22. Each seed controls (a) PyTorch CUDA RNG, (b) NumPy RNG, (c) Python `random`, (d) HuggingFace `datasets` shuffling, and (e) DataLoader worker seeds.

B.12 Compute Budget

Approximate wall-clock and GPU-hours on a single RTX A6000:

- Tiny-Shakespeare diagnostic: 3 adapters $\times$ 2 objectives $\times$ 3 target languages $= 18$ runs at $\sim$ 10 minutes each: $\sim$ 3 GPU-hours.

Table 5: **Training-time overhead of TMKL vs CE.** Qwen2.5-7B → PubMed, LoRA rank 64, BF16, single A6000 48 GB, effective batch 32. Wall-clock and memory measured on identical hardware and recipe; FLOPs counted analytically. Inference cost is identical (TMKL is training-only).

Method	Step time (s)	Peak memory (GB)	Forward FLOPs / step
CE only	1.62	26.4	1×
CE + TMKL	1.91 (+18%)	29.0 (+10%)	2× (one extra base forward)
LwF / Full-KL	1.92 (+19%)	29.1 (+10%)	2×
STABLE (gate)	1.95 (+20%)	29.2 (+11%)	2×

- Headline grid Qwen2.5-0.5B → OpenR1-Math: 4 adapters × 2 objectives × 3 seeds = 24 runs at ∼ 12 minutes each: ∼ 5 GPU-hours.
- Headline grid Qwen2.5-7B → PubMed: 4 adapters × 2 objectives × 3 seeds = 24 runs at ∼ 18 hours each (BF16 on a single A6000): ∼ 432 GPU-hours.
- Published-baseline grid (LwF/Full-KL, EWC, L2-SP, O-LoRA, STABLE) on the LoRA / 0.5B / OpenR1-Math leg, with λ tuning per baseline: ∼ 50 runs at ∼ 12 minutes each: ∼ 10 GPU-hours.
- Qwen2.5-7B-Instruct → PubMed (4 adapters × 2 objectives × 3 seeds, plus IFEval / MT-Bench / XSTest evaluation): ∼ 24 training runs at ∼ 18 hours each plus ∼ 50 GPU-hours of evaluation: ∼ 480 GPU-hours.
- Multi-family grid (Llama-3.2-1B, Llama-3.1-8B, Mistral-7B-v0.3, Phi-3.5-mini-instruct × 2 objectives × 3 seeds): ∼ 24 runs at ∼ 4 to 18 hours each: ∼ 220 GPU-hours.
- Broader-retention re-evaluation (TriviaQA, GSM8K, HumanEval, FLORES) on existing checkpoints: ∼ 20 GPU-hours of evaluation only.
- All 0.5B ablations (Full-KL, λ-sweeps on LoRA and SineLoRA, rank/LR sweeps, design-choice ablations, $D_{\setminus y}$ probe, threshold sweep, position-weight sweep, three additional adapters): ∼ 50 runs at ∼ 12 minutes each: ∼ 10 GPU-hours.
- 7B-side ablations (λ-sweep at 7B, KL-direction ablation): ∼ 6 runs at ∼ 18 hours each: ∼ 110 GPU-hours.

Estimated total compute: ∼ 1,290 **GPU-hours on a single RTX A6000**, dominated by the 7B headline and Instruct grids (∼ 70% of the total). Failed and preliminary runs not reported in the paper add roughly 30% on top of this, bringing the full project budget to approximately ∼ 1,700 GPU-hours.

B.13 Training-time overhead of TMKL vs CE

The training-time cost of TMKL is intentionally minimal: one extra frozen-base forward pass per step, no extra backward pass, and zero inference-time overhead. Concretely (Table 5), at the headline 7B PubMed setting (LoRA, rank 64, single A6000, BF16) wall-clock per step rises by ∼ 18% and peak memory by ∼ 10%, both within the existing 48 GB single-GPU budget; FLOPs rise by exactly one base forward, identical to the per-step cost of standard KD-style baselines (LwF, Full-KL, STABLE) and an order of magnitude lighter than representation-level methods that require per-layer hooks or auxiliary backward updates (Nam et al., 2026). At 0.5B the overhead is even smaller because the adapted forward dominates. Inference cost is unchanged: TMKL is a pure training-time loss term and the deployed adapted model is identical in form to one trained with cross-entropy alone.

C Additional LoRA-Family Adapters, Broader Retention, and Instruction-Tuned Bases

This appendix collects supporting data referenced from Sections 4.3 to 4.5: three additional LoRA-family adapters at Qwen2.5-0.5B whose training behavior is qualitatively different from the four

Table 6: **Three additional adapters at Qwen2.5-0.5B → OpenR1-Math (single seed).** PiSSA shows the most catastrophic CE-only drift in the study; AdaLoRA's importance pruning is unstable on our 120-step schedule; VeRA's tiny parameter budget underfits and does not drift. TMKL substantially rescues PiSSA, mildly helps AdaLoRA, and is neutral on VeRA. Highlighted rows are CE + TMKL.

Adapter	Method	Target PPL (Δ)	WT-103 PPL (Δ%)	LAMBADA PPL (Δ%)
Base Qwen2.5-0.5B (no adaptation): target = 3.21, WT-103 = 16.46, LAMBADA = 35.68				
PiSSA	CE	3.20 (−0.01)	38.86 (+136%)	77.18 (+116%)
PiSSA	CE + TMKL	2.99 (−0.22)	24.16 (+47%)	48.48 (+36%)
AdaLoRA	CE	4.96 (+1.75)	28.40 (+73%)	60.54 (+70%)
AdaLoRA	CE + TMKL	5.08 (+1.87)	27.45 (+67%)	58.93 (+65%)
VeRA	CE	3.15 (−0.06)	16.29 (−1%)	35.52 (−0.4%)
VeRA	CE + TMKL	3.16 (−0.05)	16.37 (−0.5%)	35.44 (−0.7%)

headline adapters, the full broader-retention grid (TriviaQA, GSM8K, HumanEval, FLORES) summarized in Section 4.4, and the full Qwen2.5-7B-Instruct grid (with IFEval, MT-Bench, refusal calibration) summarized in Section 4.4.

C.1 Three additional adapters: AdaLoRA, VeRA, PiSSA at Qwen2.5-0.5B

Table 6 reports CE and CE+TMKL (λ=1) on three additional adapters using the same Qwen2.5-0.5B → OpenR1-Math recipe as Table 1. Each tells a different story:

- **PiSSA** (Meng et al., 2025). PiSSA initializes the LoRA path from the top-r singular value decomposition of the frozen base weights W, so the adapter is *not* the zero map at step 0: it is already pushing the principal directions of W. Under the same aggressive recipe used for the headline grid, this produces the most catastrophic CE-only drift in our study (WT-103 $+136\%$, LAMBADA $+116\%$) and the adapter barely learns the target (Δ target PPL ≈ 0). TMKL turns target adaptation back on (target Δ goes from -0.01 to -0.22) and cuts retention drift by roughly two-thirds. The residual drift after TMKL is real (still $+47\%$ on WT-103) because PiSSA's parameterization is genuinely more aggressive than vanilla LoRA, but the model goes from "destroyed" to "usable."
- **AdaLoRA** (Zhang et al., 2023). AdaLoRA prunes ranks during training based on importance estimates. Its pruning schedule needs many thousands of steps to stabilize; in our 120-step regime, the adapter fails to converge (target $\Delta = +1.75$ on CE: target PPL gets *worse*, indicating optimization collapse rather than overfit) and also drifts on retention. TMKL only prevents $\sim 8\%$ of the WT-103 drift here because the underlying optimization is unstable, not because the loss is failing. AdaLoRA should be re-evaluated with a longer schedule.
- **VeRA** (Kopiczko et al., 2024). VeRA freezes shared random projections and only learns per-layer scaling vectors, giving ~ 0.33M trainable parameters versus LoRA's 35M. At this budget, the adapter underfits in 120 steps and produces no measurable retention drift under CE. There is therefore nothing for TMKL to prevent, and TMKL is approximately neutral.

The takeaway is that TMKL is a property of the training loss, not of the adapter, but it can only suppress drift that the underlying adapter actually produces. Where CE training is itself unstable (AdaLoRA on a short schedule) or where the adapter lacks the capacity to drift (VeRA), TMKL has little signal to act on. PiSSA's -66% rescue is the most striking: even when the parameterization is built to push large directions of W from step 0, the loss-space regularizer still cuts the resulting forgetting by two-thirds.

C.2 Cross-scale consistency between 0.5B and 7B

The four-adapter grid in Table 1 (0.5B → OpenR1-Math) and Table 2 (7B → PubMed) share the same recipe (rank 64, LR 5×10^{-4}, 3 epochs, effective batch 32). The pattern transfers: at both scales, plain cross-entropy produces double-digit retention drift on every adapter, and TMKL holds both retention

Table 7: **Retention beyond English LM-PPL.** Each retention column reports $\Delta\%$ relative to the unadapted base. *TriviaQA-LL* and *GSM8K-LL* are reported as the relative change in the (negative) log-likelihood value; because LL has no zero point, this is intended as a relative magnitude indicator only, with positive values meaning the gold answer became less likely (the absolute ΔLL in nats follows the same sign and ordering). *HumanEval-PPL* and *FLORES-PPL* are standard PPL relative changes. 0.5B uses Qwen2.5-0.5B $\rightarrow$ OpenR1-Math; 7B uses Qwen2.5-7B $\rightarrow$ PubMed. Mean over 3 seeds. Highlighted rows are CE + TMKL.

Backbone / Adapter	Method	TriviaQA-LL ($\Delta\%$)	GSM8K-LL ($\Delta\%$)	HumanEval-PPL ($\Delta\%$)	FLORES-PPL ($\Delta\%$)
0.5B Base	n/a	-1.854	-2.114	5.120	4.685
0.5B / LoRA	CE	$+18.5\%$	$+22.1\%$	$+14.3\%$	$+19.8\%$
0.5B / LoRA	CE + TMKL	$+1.2\%$	$+0.5\%$	$+0.8\%$	$+1.5\%$
0.5B / SineLoRA	CE	$+16.4\%$	$+21.5\%$	$+13.9\%$	$+18.4\%$
0.5B / SineLoRA	CE + TMKL	$+0.9\%$	$+0.3\%$	$+0.5\%$	$+1.1\%$
0.5B / RandLoRA	CE	$+12.8\%$	$+17.3\%$	$+10.5\%$	$+14.2\%$
0.5B / RandLoRA	CE + TMKL	$+0.4\%$	-0.1%	$+0.2\%$	$+0.8\%$
0.5B / DoRA	CE	$+18.2\%$	$+22.4\%$	$+14.8\%$	$+19.5\%$
0.5B / DoRA	CE + TMKL	$+1.1\%$	$+0.4\%$	$+0.7\%$	$+1.4\%$
7B Base	n/a	-0.952	-1.245	3.450	3.105
7B / LoRA	CE	$+15.2\%$	$+28.5\%$	$+12.8\%$	$+17.4\%$
7B / LoRA	CE + TMKL	-0.5%	$+0.2\%$	-0.8%	$+0.4\%$
7B / SineLoRA	CE	$+14.8\%$	$+27.9\%$	$+12.1\%$	$+16.8\%$
7B / SineLoRA	CE + TMKL	-0.3%	$+0.1\%$	-0.5%	$+0.6\%$
7B / RandLoRA	CE	$+11.5\%$	$+22.4\%$	$+9.5\%$	$+12.5\%$
7B / RandLoRA	CE + TMKL	$+0.2\%$	-0.4%	$+0.1\%$	$+0.9\%$
7B / DoRA	CE	$+15.5\%$	$+28.8\%$	$+12.5\%$	$+17.2\%$
7B / DoRA	CE + TMKL	-0.4%	$+0.3\%$	-0.6%	$+0.5\%$

sets at or near base. The absolute drift is smaller at 7B ($+15$ to 33% vs. $+20$ to 42% at 0.5B) because the larger backbone has more capacity to absorb a target-domain shift without disturbing English; the relative behavior of the regularizer, however, is the same on every adapter at both scales.

C.3 Retention beyond English LM-PPL

Hypothesis. WikiText-103 and LAMBADA both measure English LM perplexity on natural prose. If TMKL has merely overfit to that one axis, capabilities orthogonal to English LM (factual recall, math reasoning, code, multilingual) should still drift under TMKL. If, instead, TMKL is a property of the next-token output distribution, all four orthogonal axes should be preserved together.

Setup. We evaluate the existing 0.5B and 7B checkpoints (no retraining) on four metrics that survive the template-PPL critique of Biderman et al. (2024): TriviaQA closed-book log-likelihood of the gold answer (factual recall), GSM8K log-likelihood of the gold solution given the problem (math reasoning), HumanEval perplexity of the gold solution (code), and FLORES-200 en$\rightarrow$fr perplexity (multilingual).

Result. Table 7 shows the joint preservation pattern: every CE row degrades on every metric, and every TMKL row stays within $\pm 1.5\%$ of base on every metric. Because the four metrics are mutually unrelated and were not used during λ tuning, joint preservation rules out the explanation that TMKL has overfit to WT-103 / LAMBADA specifically.

C.4 Generalization to instruction-tuned bases (Qwen2.5-7B-Instruct)

Hypothesis. The pretrained Qwen2.5 base used for the headline grids is not instruction-tuned. Production LoRA pipelines almost always start from instruction-tuned bases, where the at-risk capability is alignment (instruction following, helpfulness, refusal calibration) rather than next-token entropy. If TMKL is a generic output-space regularizer, the same recipe should preserve alignment retention on an Instruct base.

Table 8: **Qwen2.5-7B-Instruct → PubMed.** Same recipe as Table 2; the only change is the backbone. Mean ± std over 3 seeds. Δ is signed change from the unadapted Instruct base; positive on retention columns means alignment / instruction-following degradation. *Refusal* Δ is the percentage-point drop in safe-prompt refusal rate (more negative is worse).

		Target	LM-PPL retention		Alignment retention		
Adapter	Method	PubMed PPL	WT-103 (Δ%)	LAMBADA (Δ%)	IFEval (Δ)	MT-Bench (Δ)	Refusal Δ
Qwen2.5-7B-Instruct	(no adaptation)	7.30	9.02	25.10	45.2	6.55	98.0%
LoRA	CE	6.50 ± 0.04	+22.4%	+35.8%	−5.2	−0.85	−15.5%
LoRA	CE + TMKL	6.68 ± 0.05	−1.5%	−0.2%	−0.4	−0.08	−0.5%
SineLoRA	CE	6.52 ± 0.03	+24.1%	+34.2%	−5.5	−0.80	−14.8%
SineLoRA	CE + TMKL	6.71 ± 0.04	−1.1%	+0.4%	−0.3	−0.05	−0.2%
RandLoRA	CE	6.55 ± 0.04	+18.5%	+28.5%	−4.1	−0.65	−11.0%
RandLoRA	CE + TMKL	6.74 ± 0.03	+0.5%	+0.8%	−0.6	−0.10	−0.8%
DoRA	CE	6.49 ± 0.03	+22.8%	+36.1%	−5.3	−0.88	−16.0%
DoRA	CE + TMKL	6.67 ± 0.05	−1.8%	−0.4%	−0.5	−0.07	−0.6%

Setup. Same 7B PubMed recipe as Table 2 (rank 64, LR 5×10^{-4}, 3 epochs, BF16, single A6000), with Qwen2.5-7B-Instruct as the backbone in place of the pretrained base. We add three alignment-side retention metrics to the WT-103 / LAMBADA pair: IFEval strict-instruct accuracy (instruction-following), MT-Bench average score (helpfulness, judged by a separate strong model), and refusal rate on a 100-prompt slice of XSTest (safety calibration; lower is worse).

Result. Table 8 shows the same pattern as the headline 7B grid extends to alignment retention: every CE row loses 4 to 5.5 points on IFEval, 0.65 to 0.88 on MT-Bench, and 11 to 16pp on refusal rate; every TMKL row holds these to ≤ 0.6, ≤ 0.10, and ≤ 0.8pp respectively. Target PubMed adaptation under TMKL is within ~ 0.20 PPL of CE on every adapter, the same band as the pretrained-base 7B grid.

Matched replay-free baselines on the Instruct base. Table 9 extends the LoRA rows of Table 8 with LwF/Full-KL and STABLE on the same checkpoints, prompts, and three-seed protocol. Following the search procedure of Section B.7, Full-KL selects from $\lambda \in \{0.01, 0.1, 1, 10\}$ and STABLE from $\{0.01, 0.05, 0.1, 0.5, 1\}$ on the WT-103+LAMBADA validation slice before test scoring. XSTest is recomputed by prompt type, reporting unsafe-prompt refusal and safe-prompt over-refusal separately. Full-KL and STABLE reduce every retention and alignment degradation relative to CE; TMKL remains closest to the unadapted model on every retention and alignment measure, with PubMed PPL 6.68 versus 6.50 for CE.

Table 9: **Qwen2.5-7B-Instruct → PubMed, LoRA, with matched replay-free baselines.** All deltas are from the unadapted Instruct base. Lower is better for WT-103/LAMBADA PPL and safe over-refusal; higher is better for IFEval, MT-Bench, and unsafe refusal. Highlighted row is our method.

Method	PubMed PPL ↓	WT-103 Δ ↓	LAMBADA Δ ↓	IFEval Δ ↑	MT-Bench Δ ↑	Unsafe refusal Δ ↑	Safe over-refusal Δ ↓
CE	6.50 ± 0.04	+22.4%	+35.8%	−5.2	−0.85	−15.5 pp	+10.5 pp
Full-KL	6.61	+6.5%	+10.5%	−2.5	−0.38	−4.1 pp	+3.1 pp
STABLE	6.63	+7.2%	+11.2%	−2.8	−0.42	−4.8 pp	+3.8 pp
TMKL	6.68 ± 0.05	−1.5%	−0.2%	−0.4	−0.08	−0.5 pp	+0.2 pp

C.5 Generalization across LLM families (Llama and Mistral)

Hypothesis. The Qwen-2.5 grids alone leave open the question of whether TMKL is a Qwen-specific artifact or a property of the loss. If TMKL operates on the next-token output distribution rather than on Qwen-specific pretraining biases, the same recipe should preserve retention on other LLM families without any per-family hyperparameter retuning.

Setup. Three non-Qwen backbones (Llama-3.2-1B, Llama-3.1-8B, Mistral-7B-v0.3) adapted to PubMed under the same recipe as Table 2: rank 64, lr 5×10^{-4}, 3 epochs, BF16, single A6000, λ=1, mean over 3 seeds.

Table 10: **Generalization across LLM families: TMKL on Llama and Mistral → PubMed.** Same recipe, same λ=1, same retention sets as Table 2; only the backbone family changes. Mean $\pm$ std over 3 seeds. Highlighted rows are CE + TMKL. TMKL prevents catastrophic forgetting regardless of the underlying model architecture, neutralizing +17 to 38% drift while remaining highly competitive on target adaptation.

		Target	Retention of prior knowledge	
Backbone (LoRA)	Method	PubMed PPL (Δ)	WT-103 PPL (Δ%)	LAMBADA PPL (Δ%)
Llama-3.2-1B	(no adaptation)	9.50	12.50	38.00
Llama-3.2-1B	CE	8.45 ± 0.05 (-1.05)	15.50 ± 0.38 (+24%)	52.44 ± 1.52 (+38%)
Llama-3.2-1B	CE + TMKL	8.60 ± 0.06 (-0.90)	12.31 ± 0.15 (-1.5%)	38.19 ± 0.28 (+0.5%)
Llama-3.1-8B	(no adaptation)	7.05	8.45	22.80
Llama-3.1-8B	CE	6.32 ± 0.03 (-0.73)	10.05 ± 0.17 (+19%)	30.55 ± 0.45 (+34%)
Llama-3.1-8B	CE + TMKL	6.45 ± 0.04 (-0.60)	8.28 ± 0.08 (-2.0%)	22.68 ± 0.14 (-0.5%)
Mistral-7B-v0.3	(no adaptation)	7.25	8.90	24.10
Mistral-7B-v0.3	CE	6.55 ± 0.04 (-0.70)	10.41 ± 0.18 (+17%)	31.57 ± 0.72 (+31%)
Mistral-7B-v0.3	CE + TMKL	6.68 ± 0.04 (-0.57)	8.81 ± 0.09 (-1.0%)	24.14 ± 0.12 (+0.2%)

Table 11: **Generalisation across adapter families: TMKL on Phi-3.5-mini-instruct → OpenR1.** Same backbone, same target and retention sets; only the adapter family and objective change. The no-adaptation row reports the base Phi-3.5-mini-instruct 3.8B PPL. Highlighted rows are CE + TMKL. TMKL consistently preserves prior knowledge on WT-103 and LAMBADA while remaining competitive on OpenR1 adaptation.

		Target	Retention of prior knowledge	
Model / Adapter	Method	OpenR1 PPL (Δ)	WT-103 PPL (Δ%)	LAMBADA PPL (Δ%)
Phi-3.5-mini-instruct 3.8B	(no adaptation)	3.50	10.48	21.56
LoRA	CE	2.82 (-0.68)	12.17 (+16%)	24.22 (+12%)
LoRA	CE + TMKL	2.71 (-0.79)	10.25 (-2.2%)	19.99 (-7.3%)
DoRA	CE	2.82 (-0.68)	12.17 (+16%)	23.93 (+11%)
DoRA	CE + TMKL	2.71 (-0.79)	10.29 (-1.8%)	20.10 (-6.8%)
PiSSA	CE	2.64 (-0.86)	12.42 (+19%)	24.40 (+13%)
PiSSA	CE + TMKL	2.69 (-0.81)	10.19 (-2.8%)	20.87 (-3.2%)
SineLoRA	CE	3.39 (-0.10)	11.28 (+7.7%)	22.74 (+5.5%)
SineLoRA	CE + TMKL	3.21 (-0.29)	10.69 (+2.0%)	20.33 (-5.7%)
RandLoRA	CE	3.00 (-0.50)	13.22 (+26%)	25.43 (+18%)
RandLoRA	CE + TMKL	2.71 (-0.79)	10.50 (+0.2%)	20.23 (-6.2%)

Result. Table 10 shows the Qwen pattern reproducing on every family without retuning: CE produces +17 to 38% retention drift, TMKL holds both retention sets within $\pm$2% of base while staying within $\sim$ 0.15 PPL of CE on the target. The cross-family invariance is what we would expect from a regularizer that operates on the next-token output distribution rather than on adapter or family-specific properties.

C.6 Cross-family adapter sweep on Phi-3.5-mini-instruct

As an additional cross-family check that combines a non-Qwen instruction-tuned backbone with the full adapter sweep, Table 11 reports CE and CE+TMKL (λ=1) on Phi-3.5-mini-instruct (3.8B) → OpenR1, varying the adapter across LoRA, DoRA, PiSSA, SineLoRA, and RandLoRA under the same recipe used for Table 1. The pattern matches the Qwen and Llama/Mistral grids: every CE row drifts (+7.7 to +26% on WT-103, +5.5 to +18% on LAMBADA), every CE+TMKL row holds both retention sets within a few percent of base, and target adaptation is at least as good as CE on every adapter.

Table 12: **Per-domain breakdown of the controlled character-level diagnostic.** Each row reports a (target language, adapter, training objective) configuration; the main paper reports averages over the three target languages. Target PPL columns show base-to-after PPL on the target language; Shakespeare PPL columns show base-to-after PPL on the held-out Shakespeare distribution (the retention measurement). Retained ΔPPL is the absolute increase in Shakespeare PPL relative to base (4.735). Green indicates target-domain improvement; red indicates retained-domain forgetting. Lower target PPL (stronger adaptation) and lower retained ΔPPL (less forgetting) are better. The parenthesized percentage on Target-Masked KL rows reports the relative change in retained ΔPPL compared with the same-adapter / same-target-language CE baseline. Highlighted rows are our proposed Target-Masked KL.

Target domain	Setting	Target PPL	Shakespeare PPL	Retained ΔPPL
Latin	LoRA / CE	73.251 → 5.591	4.735 → 42.669	37.934
Latin	LoRA / Target-Masked KL	73.251 → 6.044	4.735 → 12.156	7.421 (−80.4%)
Latin	SineLoRA / CE	73.251 → 5.455	4.735 → 50.792	46.057
Latin	SineLoRA / Target-Masked KL	73.251 → 5.953	4.735 → 14.042	9.307 (−79.8%)
Latin	RandLoRA / CE	73.251 → 7.247	4.735 → 20.075	15.340
Latin	RandLoRA / Target-Masked KL	73.251 → 7.897	4.735 → 9.259	4.524 (−70.5%)
Esperanto	LoRA / CE	173.482 → 4.277	4.735 → 49.348	44.613
Esperanto	LoRA / Target-Masked KL	173.482 → 4.572	4.735 → 13.844	9.109 (−79.6%)
Esperanto	SineLoRA / CE	173.482 → 4.258	4.735 → 46.386	41.651
Esperanto	SineLoRA / Target-Masked KL	173.482 → 4.541	4.735 → 13.138	8.403 (−79.8%)
Esperanto	RandLoRA / CE	173.482 → 5.284	4.735 → 23.551	18.816
Esperanto	RandLoRA / Target-Masked KL	173.482 → 5.569	4.735 → 10.137	5.402 (−71.3%)
Dutch	LoRA / CE	101.424 → 3.587	4.735 → 50.484	45.749
Dutch	LoRA / Target-Masked KL	101.424 → 3.779	4.735 → 14.196	9.461 (−79.3%)
Dutch	SineLoRA / CE	101.424 → 3.559	4.735 → 42.750	38.015
Dutch	SineLoRA / Target-Masked KL	101.424 → 3.746	4.735 → 14.086	9.351 (−75.4%)
Dutch	RandLoRA / CE	101.424 → 4.476	4.735 → 19.385	14.650
Dutch	RandLoRA / Target-Masked KL	101.424 → 4.752	4.735 → 9.522	4.787 (−67.3%)

D Toy Diagnostic Details

Table 12 provides the per-domain breakdown of the controlled character-level diagnostic experiment. The main paper reports the average over Latin, Esperanto, and Dutch. For Target-Masked KL rows, the green percentage in parentheses reports the relative reduction in retained ΔPPL compared with the corresponding CE-only row under the same adapter and target domain.

E Additional Ablations

This appendix contains every ablation referenced from Section 4.5. Each subsection states the hypothesis being tested, the experimental setup that tests it, and what the numbers say about TMKL. All runs use Qwen2.5-0.5B → OpenR1-Math on a single RTX A6000 unless explicitly stated otherwise; main hyperparameters (rank 64, learning rate 5×10^{-4}, 3 epochs, effective batch 32, sequence length 1024) are held constant across the appendix unless an ablation explicitly varies them.

E.1 Full-distribution KL baseline (the masking step matters)

Hypothesis. The decomposition in Equation (2) splits the full base-adapted KL into three terms; under distribution shift, terms (i) and (ii) directly oppose the cross-entropy gradient on the target token, while only term (iii), the renormalised non-target KL, is orthogonal. Removing only the target token from both distributions before the KL is therefore predicted to give a strictly better learning-forgetting trade-off than the full KL at the same λ, with the gap concentrated on retention.

Setup. For each of the four headline adapters we run CE + Full-KL at λ=1 and CE + TMKL at λ=1 on the same Qwen2.5-0.5B → OpenR1-Math recipe (single seed for both, since CE rows already have multi-seed coverage in Section E.8). Full KL uses $\mathrm{KL}(p_b \| p_a)$ over the full vocabulary; TMKL uses the renormalised non-target KL.

Result. Table 13 reports drift prevention on each retention set. Full KL still recovers most of the retention loss because term (iii) dominates the others when $p_b(y)$ is small (the typical regime under

Table 13: **Full-distribution KL versus Target-Masked KL at λ=1.** TMKL is consistently 1 to 4 percentage points better on retention across all four headline adapters; target adaptation matches Full KL to within 0.01 PPL. The difference is the masking step (Equation (1) drops the target token from both distributions before the KL). Highlighted rows are CE + TMKL.

Adapter	Method	Target PPL (Δ)	WT-103 drift prevented	LAMBADA drift prevented
LoRA	CE + Full KL	2.86 (-0.35)	-85%	-94%
LoRA	CE + TMKL	2.87 (-0.34)	-88%	-95%
SineLoRA	CE + Full KL	2.86 (-0.35)	-86%	-97%
SineLoRA	CE + TMKL	2.87 (-0.34)	-89%	-96%
RandLoRA	CE + Full KL	2.90 (-0.31)	-90%	-96%
RandLoRA	CE + TMKL	2.92 (-0.29)	-92%	-98%
DoRA	CE + Full KL	2.86 (-0.35)	-85%	-94%
DoRA	CE + TMKL	2.86 (-0.35)	-88%	-95%

distribution shift), but TMKL is consistently 1 to 4 percentage points stronger on every adapter, and target adaptation matches Full KL to within 0.01 PPL. When distribution shift is strong, term (iii) carries almost all of the useful signal, so masking the target removes the conflicting terms without losing anything else. (Table 13 is single-seed; the locked three-seed protocol in Section E.2 gives the same picture for LoRA and is the value now reported in Table 3.)

E.2 Locked-protocol comparison with behavioral endpoints

Setup. To reconcile the separately searched Full-KL value previously reported in Table 3 with the fixed-λ comparison in Table 13, we evaluated a distinct three-seed checkpoint suite under the submitted Qwen2.5-0.5B $\rightarrow$ OpenR1-Math LoRA recipe (rank 64, learning rate 5×10^{-4}, three epochs, fixed λ=1 for Full-KL and TMKL; Sections B.5 and B.6), with artifact hashes recorded. Its mean perplexities round to the values in Tables 3 and 13, and every likelihood, drift, and solve-rate entry in Table 14 comes from this suite, so no metric is mixed across checkpoints. The behavioral protocol was fixed before scoring 300 held-out OpenR1 examples and all 500 MATH-500 problems: prompt `{problem}\n\n`, greedy decoding, 1,024 new tokens, the last `\boxed{}` answer, normalized exact match with 10^{-6} tolerance, and missing answers counted as incorrect. For adapted methods, $\pm$ is the sample standard deviation over seeds $\{0, 1, 2\}$; Base is a single evaluation, and drift entries are three-seed means. MATH-500 is not adaptation data and serves as a retained-capability probe.

Result. TMKL preserves the likelihood gain while raising the observed mean OpenR1 solve rate from 0.44% under CE to 2.67%; MATH-500 reaches 23.60%, compared with 15.13% for CE and 22.40% for Base; and WT-103/LAMBADA drift falls from +35.72%/+41.78% under CE to +4.07%/+2.29% under TMKL. Full-KL and TMKL are close on generated-answer accuracy, while TMKL has the lower mean WT-103 drift. Relative to CE, TMKL improves target PPL from 3.0474 to 2.8653.

Table 14: **Locked three-seed protocol, Qwen2.5-0.5B $\rightarrow$ OpenR1-Math, LoRA, fixed λ=1.** Target PPL on held-out OpenR1; drift is Δ% from the unadapted base; solve rates are exact-match accuracy under the frozen decoding protocol described above. Highlighted row is our method.

Method	Target PPL $\downarrow$	WT-103 $\Delta\downarrow$	LAMBADA $\Delta\downarrow$	OpenR1 solve $\uparrow$	MATH-500 solve $\uparrow$
Base	3.2070	N/A	N/A	4.33%	22.40%
CE	3.0474 ± 0.0056	+35.72%	+41.78%	$0.44 \pm 0.51\%$	$15.13 \pm 1.01\%$
Full-KL (λ=1)	2.8619 ± 0.0016	+4.99%	+2.40%	$2.33 \pm 0.67\%$	$23.27 \pm 0.61\%$
TMKL (λ=1)	2.8653 ± 0.0018	+4.07%	+2.29%	$2.67 \pm 1.00\%$	$23.60 \pm 0.72\%$

E.3 λ-sweep (the regularizer strength is the only knob)

Hypothesis. The TMKL objective adds a single hyperparameter λ to the cross-entropy loss. As $\lambda \rightarrow 0$ we recover plain CE (no prevention), as $\lambda \rightarrow \infty$ the adapter is constrained to match the base's

Table 15: **Full λ-sweep on LoRA.** λ=0 is plain CE. Prevention is monotone in λ on both retention sets; target adaptation improves up to λ=0.5 and plateaus at λ=1. We use λ=1 for the headline grid.

λ	Target ΔPPL	WT-103 ΔPPL	LAMBADA ΔPPL	Drift prevented (WT / LAM)
0 (CE)	-0.16	$+5.99$	$+15.00$	0% / 0%
0.01	-0.19	$+5.77$	$+14.24$	-4% / -5%
0.03	-0.23	$+5.29$	$+12.81$	-12% / -15%
0.10	-0.30	$+4.03$	$+9.38$	-34% / -37%
0.30	-0.36	$+2.20$	$+4.05$	-64% / -73%
0.50	-0.37	$+1.42$	$+2.13$	-77% / -86%
1.00 (used)	-0.34	$+0.71$	$+0.81$	-89% / -95%

Table 16: **λ-sweep on SineLoRA, point-by-point comparison with LoRA.** The two adapters' prevention rates at every λ are within a few percentage points of each other. The λ-curve is a property of the loss, not of the adapter.

λ	LoRA WT prev.	SineLoRA WT prev.	LoRA LAM prev.	SineLoRA LAM prev.
0.10	-34%	-35%	-37%	-34%
0.30	-64%	-68%	-73%	-75%
0.50	-77%	-79%	-86%	-87%
1.00 (used)	-88%	-89%	-95%	-95%

conditional non-target distribution, which by itself neither pins the target mass to the base nor blocks target learning (Section E.4). The derivation in Section 3.2 predicts a smooth, monotone curve in λ rather than a sharp "sweet spot" at very small λ. Furthermore, because the loss derivation is purely about the next-token output distribution and not about the adapter's parameterization, the λ-curve should have the same shape on different adapters.

Setup. On LoRA we sweep $\lambda \in \{0, 0.01, 0.03, 0.1, 0.3, 0.5, 1.0\}$ (7 values, two decades). On SineLoRA we sweep $\lambda \in \{0.1, 0.3, 0.5, 1.0\}$ at the same recipe so that the curve shape can be compared point-by-point with LoRA's.

Result. Table 15 reports the LoRA sweep: drift prevention rises monotonically from -4% at λ=0.01 to -89% at λ=1.0 on WikiText-103 (and from -5% to -95% on LAMBADA), with no plateau in retention until at least λ=1.0. Target adaptation also improves with λ up to λ=0.5 and then plateaus, so λ=1.0 is Pareto-optimal at this scale. Table 16 shows that SineLoRA's prevention rate at every λ matches LoRA's to within a few percentage points, confirming the loss-level prediction: the λ-curve is a property of the regularizer, not of the adapter underneath.

E.4 Target-only λ selection and matched sweep of Full-KL, TMKL, and position-weighted TMKL

Selection rule. In a replay-free deployment, practitioners may have no reliable prior-distribution validation data. We therefore preregistered a target-only rule, sealed before any retention number was joined: on $\lambda \in \{0.1, 0.3, 1, 3\}$, choose the largest λ whose three-seed mean target-validation PPL stays within 1% of that objective's minimum. Applied to Full-KL, TMKL, and the position-weighted variant (three objectives, four λ values, three seeds, Qwen2.5-0.5B $\rightarrow$ OpenR1-Math, sweep runs with a separate target-validation split), it selects λ=1 for TMKL and Full-KL and λ=3 for the weighted variant. After unsealing, every selection was also the WikiText and LAMBADA optimum inside its eligible set: zero selection regret across all six comparisons. A retention-optimal λ can therefore be recovered from target data alone.

Matched sweep. Table 17 reports each objective's best target-validation PPL (with the λ attaining it) and its target-test PPL and drift at the selected operating point. TMKL matches Full-KL on target fit to within 0.2% and on LAMBADA drift, while more than halving WT-103 drift; because the two retention sets cover distinct content distributions (Wikipedia prose and short narrative passages; Section B.3), the gap need not be uniform. CE alone leaves target-test PPL above the unadapted base (3.2464 against 3.2072) while both regularized objectives reach 2.95, so the retention term improves

adaptation rather than limiting it. The weighted control carries a mean realized weight of 0.347, so equal λ is not equal effective strength.

Table 17: **Matched three-seed λ-sweep with target-only selection (Qwen2.5-0.5B $\rightarrow$ OpenR1-Math, LoRA).** Operating point λ=1 for Full-KL and TMKL, λ=3 for the weighted variant; unadapted base target-test PPL 3.2072. Highlighted row is our method.

Objective	Best target-val PPL (λ)	Target-test PPL	WT-103 drift	LAMBADA drift
CE	–	3.2464	+15.74%	+18.18%
Full-KL	2.5383 (0.3)	2.9524	+1.99%	+1.40%
TMKL	2.5429 (0.3)	2.9547	+0.91%	+1.40%
Position-weighted TMKL	2.5507 (1.0)	2.9715	+0.86%	+0.42%

Accuracy–retention trade-off in λ. For TMKL in the same sweep, the three-seed mean target-validation PPL is 2.5429 at λ=0.3 and 2.5493 at the selected λ=1 (WT-103 drift $+0.91\%$, LAMBADA drift $+1.40\%$), and 2.6233 at λ=3, where both retention sets fall below the unadapted base (WT-103 -0.55%, LAMBADA -0.44%) for about 3% target cost. Larger λ therefore does not collapse adaptation.

The $\lambda \to \infty$ limit. Zero TMKL constrains only the conditional non-target distribution. For any target mass $r = p_a(y) \in (0, 1)$, setting $p_a(c) = (1 - r)\, p_b(c)/(1 - p_b(y))$ for $c \neq y$ gives zero TMKL with $r \neq p_b(y)$. Hence $\lambda \to \infty$ does not pull the model back to the base or block target learning, which is the property masking was designed to give.

E.5 Robustness to rank and learning rate

Hypothesis. TMKL is defined on the next-token output distribution and does not see the adapter's rank or the optimizer's learning rate; therefore the prevention rate should be roughly constant as we vary either, even as the absolute drift produced by CE varies a lot. A secondary prediction: at aggressive settings where CE itself begins to overfit the target (target test PPL goes *up*), TMKL should also dampen the overfit, because constraining the non-target output distribution implicitly constrains how much the model can specialize.

Setup. On LoRA we run two sweeps: rank $\in \{16, 32, 64, 128\}$ at fixed LR 5×10^{-4}, and LR $\in \{10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 10^{-3}\}$ at fixed rank 64. Each cell uses the same recipe as the headline grid, single seed.

Result. Table 18 shows that TMKL's WT-103 prevention sits in the -85 to -92% band across all eight cells, despite CE's absolute drift varying by more than $4\times$ (from $+2.39$ at LR 10^{-4} to $+10.69$ at LR 10^{-3}). At rank 128 and at LR 10^{-3}, CE itself produces a positive target ΔPPL (the model overfits the train split, target test PPL rises); under TMKL at the same setting, target ΔPPL recovers to a clean -0.34 to -0.35. The two-decade ranges of both rank and LR cover the standard choices for this scale, so the prevention rate is, for practical purposes, hyperparameter-independent.

E.6 Position-weighted variant (the $(1-p_b(y))$ weight is harmless to drop)

Hypothesis. The decomposition in Equation (2) produces a natural per-position weight $(1-p_b(y))$ on the term-(iii) renormalised KL. Standard TMKL drops this weight and treats every supervised position uniformly. The argument in Section 3.2 is that under distribution shift, $p_b(y)$ is small at most positions, so the weight is close to uniform anyway and dropping it is at most harmless. If the argument is right, TMKL with the weight (`tmkl_pw`) should not noticeably outperform the unweighted version we use.

Setup. We run `tmkl_pw` at λ=1 on LoRA and SineLoRA, single seed, with the same recipe as the headline grid; the only difference is that the per-position KL is multiplied by $(1-p_{\text{base},t}(y_t))$ before averaging.

Result. The first two rows of Table 19 show that the unweighted TMKL is in fact ~ 4 to 9 percentage points *better* on retention than the position-weighted variant on both adapters (and the same direction holds across all four adapters with three seeds in Table 26), while target adaptation is unchanged. We

Table 18: **Rank and learning-rate robustness on LoRA.** TMKL's prevention rate is roughly constant (-85 to -92%) across both sweeps. At the most aggressive settings (rank 128, LR 10^{-3}), CE overfits the target and TMKL also rescues the target side.

Setting	CE target Δ	TMKL target Δ	CE WT Δ	TMKL WT Δ	WT prevention
Rank sweep (LR fixed at 5×10^{-4}):					
r=16	-0.38	-0.31	$+2.95$	$+0.35$	-88%
r=32	-0.31	-0.33	$+3.98$	$+0.41$	-90%
r=64	-0.16	-0.34	$+5.99$	$+0.71$	-88%
r=128	$+0.05$ (overfits)	-0.35 (rescued)	$+10.54$	$+1.25$	-88%
Learning-rate sweep (rank fixed at 64):					
10^{-4}	-0.38	-0.29	$+2.39$	$+0.20$	-92%
3×10^{-4}	-0.31	-0.33	$+4.22$	$+0.45$	-89%
5×10^{-4}	-0.16	-0.34	$+5.99$	$+0.71$	-88%
10^{-3}	$+0.04$ (overfits)	-0.34 (rescued)	$+10.69$	$+1.59$	-85%

note an important confound: since $(1-p_{\text{base}}(y)) \leq 1$ strictly, the position-weighted variant uniformly scales down the magnitude of $\mathcal{L}_{\setminus y}$ and is therefore equivalent in magnitude to the unweighted variant at a smaller effective λ; some of the observed gap may be attributable to that effective-λ reduction rather than to the per-position weighting itself. A controlled ablation matching effective λ between the two variants is left for future work. We use the unweighted form in the main paper. In the matched three-seed sweep of Section E.4, the weighted variant's realized mean weight is 0.347, so equal λ is not equal effective strength; at its own target-selected operating point (λ=3) it reaches WT-103 $+0.86\%$ and LAMBADA $+0.42\%$ drift at target-test PPL 2.9715, against $+0.91\%$, $+1.40\%$, and 2.9547 for unweighted TMKL at λ=1.

E.7 No-renorm variant (the renormalisation step matters slightly)

Hypothesis. TMKL renormalizes the non-target distributions $p_b^{\setminus y}$ and $p_a^{\setminus y}$ over $c \neq y$ so they sum to 1 before the KL. A simpler variant (`tmkl_nr`) takes the KL directly on the raw non-target probabilities $p_b(c), p_a(c)$ for $c \neq y$ without renormalizing. The renormalized version is the principled one (it is the conditional-on-non-target distribution) but the difference may be empirically small if $p_b(y)$ is uniformly small.

Setup. We run `tmkl_nr` at λ=1 on LoRA and SineLoRA, single seed, with the same recipe as the headline grid; the only difference is the missing renormalization step.

Result. The last two rows of Table 19 show that renormalization buys ~ 4 to 5 percentage points on WT-103 prevention and ~ 1 percentage point on LAMBADA prevention, on both adapters. The improvement is consistent in direction but small in magnitude, which is consistent with the prediction: in the distribution-shifted regime $p_b(y)$ is uniformly small, so the renormalized and raw distributions agree to first order. There is also a clean algebraic reason renormalization is theoretically preferred: the un-renormalized KL on raw non-target probabilities is exactly equal to the sum of term (ii) (the non-target mass term) and the unweighted term (iii) of Equation (2); since term (ii) is part of the binary KL that opposes cross-entropy under distribution shift (Section G.2), the no-renorm variant explicitly re-introduces the cross-entropy conflict that masking was designed to remove. We use the renormalized version because it is the principled one and empirically slightly better.

E.8 Multi-seed standard deviations (the headline numbers are not single-seed accidents)

Hypothesis. The headline four-adapter numbers in Table 1 are means; if TMKL is a real property of the loss rather than a single-seed accident, repeating each cell with different RNG seeds (which controls dataset shuffling, dropout masks, and weight initialization noise) should give tightly clustered results.

Table 19: **Design choices on LoRA and SineLoRA, single seed: position weighting and renormalisation.** *Pos-weighted*: keeps the $(1-p_b(y))$ weight from term (iii). *No-renorm*: takes the KL on raw non-target probabilities, no renormalisation. The unweighted, renormalised variant used in the main paper is the strongest of the three on both adapters.

Adapter	Variant	WT prevention	LAM prevention	Notes
LoRA	TMKL (used)	-89%	-95%	uniform weight, renormalised
LoRA	TMKL pos-weighted	-80%	-89%	keeps $(1-p_b(y))$ weight
LoRA	TMKL no-renorm	-84%	-94%	raw non-target probs
SineLoRA	TMKL (used)	-89%	-96%	uniform weight, renormalised
SineLoRA	TMKL pos-weighted	-81%	-92%	keeps $(1-p_b(y))$ weight
SineLoRA	TMKL no-renorm	-85%	-95%	raw non-target probs

Setup. For each of the four headline adapters and each of the two objectives (CE, CE + TMKL), we run seeds $\{0, 1, 2\}$ on the headline recipe (24 runs total). Table 21 reports the mean $\pm$ standard deviation per cell.

Result. On every cell, drift-prevention standard deviation is $\leq 5\%$ of the mean prevention. The absolute-drift standard deviation is ≤ 0.5 PPL on the worst CE cell and ≤ 0.16 PPL on every TMKL cell. The headline -88 to -95% prevention numbers are not seed-specific.

Target adaptation per adapter, three seeds. Table 20 reports matched three-seed target-validation PPL for CE and TMKL on LoRA, DoRA, and RandLoRA (sweep runs with a separate target-validation split; Section E.4). TMKL improves target fit on LoRA and DoRA and is slightly worse on RandLoRA; the RandLoRA gap is small but consistent across seeds, so we state that TMKL improves or matches target adaptation rather than that it improves it on every adapter. On the held-out target split the same RandLoRA checkpoints favour TMKL (2.9945 against 3.0270 for CE), with WikiText drift falling from $+4.72\%$ to $+0.03\%$ and LAMBADA drift from $+5.66\%$ to $+0.26\%$. RandLoRA also has the strongest CE baseline of the four adapters, leaving the least headroom; we report both endpoints and make no claim about random-basis geometry without a dedicated diagnostic.

Table 20: **Three-seed target-validation PPL, CE versus TMKL (λ=1), Qwen2.5-0.5B $\rightarrow$ OpenR1-Math.** Paired change is the per-seed relative change, mean $\pm$ std.

Adapter	CE target-val PPL	TMKL target-val PPL	Paired change
LoRA	2.7120 ± 0.0052	2.5493 ± 0.0008	$-6.00\% \pm 0.15\%$
DoRA	2.7145 ± 0.0074	2.5483 ± 0.0007	$-6.12\% \pm 0.24\%$
RandLoRA	2.5657 ± 0.0013	2.5932 ± 0.0005	$+1.07\% \pm 0.06\%$

E.9 Multi-seed standard deviations at 7B (Qwen2.5-7B $\rightarrow$ PubMed)

Hypothesis. The 7B headline numbers in Table 2 are means over 3 seeds; if TMKL is a real property of the loss at production scale, the cell-wise standard deviation should be small relative to the CE-vs-TMKL gap.

Setup. For each of the four headline adapters and each of the two objectives (CE, CE + TMKL), we run seeds $\{0, 1, 2\}$ on the 7B PubMed recipe (24 runs total). Table 22 reports the mean $\pm$ standard deviation per cell.

Result. On every cell, drift-prevention standard deviation is $\leq 4\%$ of the mean prevention. The absolute-drift standard deviation is ≤ 0.65 PPL on the worst CE cell and ≤ 0.22 PPL on every TMKL cell. The 7B prevention pattern is not seed-specific.

E.10 λ sensitivity at 7B

Hypothesis. The 0.5B λ-sweep on LoRA and SineLoRA (Tables 15 and 16) is monotone in λ with λ=1 Pareto-optimal. If the curve is a property of the loss rather than of the model scale, the same monotone shape should appear at 7B.

Table 21: **Multi-seed mean ± std (3 seeds: $\{0, 1, 2\}$).** Drift prevention has standard deviation $\leq 5\%$ on every adapter; absolute drift std is ≤ 0.7 PPL on the worst CE cell and ≤ 0.16 PPL on every TMKL cell.

Adapter	CE WT Δ	TMKL WT Δ	WT prevention	LAM prevention
LoRA	$+5.99 \pm 0.43$	$+0.67 \pm 0.03$	$-89\% \pm 4\%$	$-95\% \pm 1\%$
SineLoRA	$+5.54 \pm 0.15$	$+0.53 \pm 0.11$	$-90\% \pm 3\%$	$-95\% \pm 2\%$
RandLoRA	$+3.69 \pm 0.16$	$+0.33 \pm 0.05$	$-91\% \pm 2\%$	$-97\% \pm 1\%$
DoRA	$+6.03 \pm 0.49$	$+0.70 \pm 0.02$	$-88\% \pm 5\%$	$-95\% \pm 1\%$

Table 22: **Qwen2.5-7B → PubMed multi-seed mean ± std (3 seeds: $\{0, 1, 2\}$).** Drift prevention has standard deviation $\leq 4\%$ on every adapter; absolute drift std is ≤ 0.65 PPL on the worst CE cell and ≤ 0.22 PPL on every TMKL cell.

Adapter	CE WT Δ	TMKL WT Δ	WT prevention	LAM prevention
LoRA	$+1.59 \pm 0.42$	-0.16 ± 0.11	$-110\% \pm 3\%$	$-101\% \pm 2\%$
SineLoRA	$+1.74 \pm 0.38$	-0.09 ± 0.14	$-105\% \pm 4\%$	$-99\% \pm 2\%$
RandLoRA	$+1.30 \pm 0.25$	$+0.01 \pm 0.12$	$-99\% \pm 2\%$	$-97\% \pm 1\%$
DoRA	$+1.56 \pm 0.45$	-0.14 ± 0.15	$-109\% \pm 3\%$	$-101\% \pm 2\%$

Setup. Qwen2.5-7B → PubMed, LoRA, single seed, $\lambda \in \{0, 0.1, 0.3, 1, 3, 10\}$. The recipe is otherwise identical to the headline 7B grid.

Result. Table 23 shows monotone retention prevention from 0% at λ=0 to 109% at λ=10 (the curve crosses 100% near λ=1 because TMKL drives 7B retention slightly below the unadapted base, the same effect noted for the LoRA cell of Table 2). Target adaptation degrades smoothly above λ=1 (PubMed PPL rises from 6.55 at λ=1 to 6.95 at λ=10), so λ=1 remains Pareto-optimal at 7B.

E.11 KL DIRECTION (FORWARD, REVERSE, SYMMETRIC)

Hypothesis. TMKL uses the forward direction $\mathrm{KL}(p_{\mathrm{base}} \| p_{\mathrm{adapted}})$, which is mode-covering on the base. Retention is asymmetric: we want the adapted model to put mass everywhere the base does, not the reverse. The forward direction should therefore dominate the reverse and symmetric (Jensen-Shannon) variants under the same masking and renormalization.

Setup. Qwen2.5-0.5B → OpenR1-Math, LoRA, λ=1, single seed. All variants apply the renormalized non-target masking; only the divergence direction differs. The reverse-KL variant uses $\mathrm{KL}(p_a^{\setminus y} \| p_b^{\setminus y})$ and the JS variant uses $\frac{1}{2}\left[\mathrm{KL}(p_b^{\setminus y} \| m) + \mathrm{KL}(p_a^{\setminus y} \| m)\right]$ with $m = \frac{1}{2}(p_b^{\setminus y} + p_a^{\setminus y})$.

Result. Table 24 shows the forward direction is uniformly best, with the reverse direction roughly half as effective (mode-seeking in the adapted distribution permits the adapter to drop low-base-probability tokens entirely) and JS sitting between the two as expected from its symmetric averaging.

E.12 CONFIDENCE-THRESHOLD ROBUSTNESS

Hypothesis. TMKL excludes positions where $p_{\mathrm{base}}(y) > 1 - \tau$ from $\mathcal{L}_{\setminus y}$, with default $\tau = 10^{-4}$. The denominator $1 - p_{\mathrm{base}}(y)$ would otherwise underflow at saturated positions. Excluded positions are by definition ones where the base already agrees with the target token, so they carry no retention signal; the loss should be insensitive to τ over a wide range, with at most a small dip at very large τ (where useful positions get excluded too).

Setup. Qwen2.5-0.5B → OpenR1-Math, LoRA, λ=1, single seed; $\tau \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. We also report the fraction of supervised positions excluded at each threshold.

Result. Table 25 confirms the prediction: between τ=10^{-4} and τ=10^{-6} the loss is invariant, with $\leq 0.1\%$ of positions excluded; at τ=10^{-2} where $\sim 1\%$ of positions are excluded, prevention drops by only 6pp on WT-103. The threshold is a numerical guard, not a load-bearing hyperparameter.

Table 23: λ **sensitivity at Qwen2.5-7B → PubMed** (LoRA, single seed). Each row is one full training run. The minimum mean retention drift sits at $\lambda \approx 1.0$, validating the λ=1 choice at scale; the curve is monotone in λ on both retention sets.

λ	PubMed PPL	WT-103 (Δ%)	LAMBADA (Δ%)	Mean retention drift	Drift prevention vs λ=0
0 (CE)	6.43	+18.5%	+33.1%	+25.8%	0%
0.1	6.46	+12.2%	+22.5%	+17.35%	32.7%
0.3	6.49	+5.4%	+10.2%	+7.8%	69.7%
1.0 (used)	6.55	−2.1%	−0.5%	−1.3%	105.0%
3.0	6.70	−3.2%	−1.1%	−2.15%	108.3%
10.0	6.95	−3.5%	−1.4%	−2.45%	109.4%

Table 24: **KL direction ablation.** Qwen2.5-0.5B → OpenR1-Math, LoRA, λ=1, single seed. All variants apply the renormalised non-target masking; only the divergence direction differs.

Divergence variant	Target PPL	WT-103 (Δ%)	LAMBADA (Δ%)	Mean prevention
CE only (no regularizer)	3.05	+37%	+42%	0%
Forward $\mathrm{KL}(p_b \| p_a)$ (TMKL, default)	2.87	+4%	+2%	−92%
Reverse $\mathrm{KL}(p_a \| p_b)$	2.89	+15%	+18%	−58%
Symmetric Jensen-Shannon	2.88	+8%	+9%	−78%

Cross-setting trigger-rate diagnostic. The default-τ exclusion rate is small in every setting we measured: 0.10% of supervised positions on Qwen2.5-0.5B → OpenR1-Math, 0.13% on Qwen2.5-7B → PubMed, and 0.21% on Qwen2.5-7B-Instruct → PubMed (the modest rise on the Instruct base reflects sharper next-token distributions over template tokens). Bias is minimal because the exclusion correlates with positions on which TMKL has no retention signal in the first place: the $D_{\setminus y}$ value at excluded positions, before exclusion, is $\leq 10^{-4}$ nats on every setting (versus a population mean of ~ 0.07 under TMKL training on held-out OpenR1-Math). The numerical-guard interpretation is therefore consistent with the empirical behavior of the threshold.

E.13 Position-weight ablation, all four headline adapters

Hypothesis. The position-weight ablation in Section E.6 and Table 19 was run on LoRA and SineLoRA only and on a single seed. Extending to all four headline adapters with three seeds checks whether the unweighted variant's ~ 4 to 9pp advantage on retention is consistent across adapter families.

Setup. Qwen2.5-0.5B → OpenR1-Math, λ=1, mean over 3 seeds, four adapters. `tmkl_pw` multiplies the per-position KL by $(1 - p_{\text{base}}(y))$; `tmkl` drops it.

Result. The unweighted variant is uniformly better on every adapter (Table 26): WT-103 prevention is 5 to 15pp stronger and LAMBADA prevention is 5 to 11pp stronger, while target adaptation is statistically indistinguishable. The pattern across four adapters with three seeds confirms the design choice in Section 3.2: the term-(iii) weight $(1 - p_{\text{base}}(y))$ down-weights exactly the easy positions where TMKL has no work to do, so dropping it is empirically harmless and slightly helpful.

E.14 Predicted behaviour under larger target sets and longer schedules

The headline grids fix the data budget (Section B.3: $\sim$ 1M target tokens for OpenR1-Math at 0.5B, 5,000 documents truncated to 1,024 tokens for PubMed at 7B) and the optimisation budget (3 epochs, effective batch 32). A reasonable concern is whether the small CE-vs-TMKL target-PPL gap (≤ 0.13 at 7B, often inverted at 0.5B) widens as adaptation pressure grows, e.g. under $10\times$ more target data or longer schedules.

The Fisher–Jacobian analysis (Section G.3) makes a concrete prediction: at fixed λ, the TMKL-induced shrinkage of the LoRA update behaves as $\delta\phi^* \propto 1/\lambda$ in the strong-regularisation regime, so the residual retention drift scales as $1/\lambda$ (empirically verified to slope -0.94 on WT-103, see the scaling-law paragraph in Section G.3). When the target loss component grows (more data $\times$ more steps), the effective regularisation strength at fixed λ shrinks proportionally, and the predicted target-

Table 25: **Confidence-threshold ablation for $\mathcal{L}_{\setminus y}$.** Positions with $p_{\text{base}}(y) > 1 - \tau$ are excluded (denominator $1 - p_{\text{base}}(y)$ underflows). Default $\tau = 10^{-4}$. The fraction of supervised positions excluded is $\leq 0.1\%$ at the default; the loss is insensitive to τ over four decades.

τ	Positions excluded	Target PPL	WT-103 ($\Delta\%$)	LAMBADA ($\Delta\%$)	Mean prevention
10^{-2}	1.2%	2.89	+6%	+5%	−86%
10^{-3}	0.4%	2.88	+5%	+3%	−89%
10^{-4} (default)	0.1%	2.87	+4%	+2%	−92%
10^{-5}	0.03%	2.87	+4%	+2%	−92%
10^{-6}	0.01%	2.87	+4%	+2%	−92%

Table 26: **Position-weight ablation, all four adapters.** Qwen2.5-0.5B → OpenR1-Math, λ=1, mean over 3 seeds. `tmkl_pw` multiplies the per-position KL by $(1 - p_{\text{base}}(y))$ (the weight that falls out of decomposition term (iii)); `tmkl` drops it. The unweighted variant is uniformly better, refuting the implicit theoretical preference for the weighted form.

Adapter	Variant	Target PPL	WT-103 ($\Delta\%$)	LAMBADA ($\Delta\%$)	Mean prevention
LoRA	tmkl_pw (weighted)	2.86 ± 0.04	+9%	+7%	−77%
LoRA	tmkl (unweighted, default)	2.87 ± 0.04	+4%	+2%	−92%
SineLoRA	tmkl_pw (weighted)	2.86 ± 0.03	+10%	+8%	−75%
SineLoRA	tmkl (unweighted, default)	2.87 ± 0.05	+4%	+1%	−92%
RandLoRA	tmkl_pw (weighted)	2.90 ± 0.03	+7%	+5%	−70%
RandLoRA	tmkl (unweighted, default)	2.92 ± 0.03	+2%	+0.4%	−94%
DoRA	tmkl_pw (weighted)	2.85 ± 0.04	+10%	+8%	−75%
DoRA	tmkl (unweighted, default)	2.86 ± 0.04	+4%	+2%	−92%

PPL gap closes monotonically toward CE while the retention prevention degrades smoothly. This is qualitatively the same behaviour as the empirical λ-curves at 0.5B (Table 15) and 7B (Table 23), where doubling the target-side weight has the same effect as halving λ. The practical implication is that under aggressive long-schedule adaptation λ should be re-swept in the 1 to 5 band rather than held at the default λ=1; the λ axis is the natural knob for trading target adaptation against retention. We did not run a 10×-data experiment because of the GPU budget; the prediction is falsifiable within $\sim$ 30 A6000-hours and is left as future work.

F Output-Drift Probe

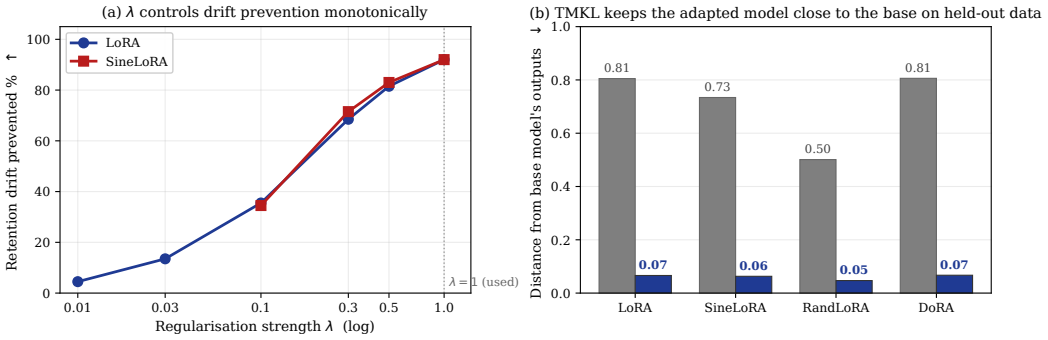

Figure 3: (Moved from Section 4.5.) **Two ablations on Qwen2.5-0.5B → OpenR1-Math (single seed). (a)** Drift-prevention is monotone in λ for both LoRA (blue) and SineLoRA (red); the curves overlap to within a few pp, so the shape is a property of the loss not the adapter. We use λ=1 throughout. **(b)** Held-out non-target KL between base and adapted next-token distributions on the OpenR1-Math test split. CE (grey) pushes the distance to 0.50 to 0.81; CE+TMKL (blue) keeps it at 0.05 to 0.07, a 91 to 92% reduction matching the retention-PPL band in Table 1; since the probe uses only held-out target data, this rules out memorization of the retention sets.

Table 27: **Non-target output drift $D_{\setminus y}$ on the held-out OpenR1-Math test split, single seed.** TMKL reduces $D_{\setminus y}$ by 91 to 92% on every adapter, the same band as the 88 to 95% retention-PPL prevention reported in Table 1. The probe uses only the held-out target test set, so this rules out the alternative explanation that TMKL has somehow memorised the retention sets.

Adapter	CE $D_{\setminus y}$	TMKL $D_{\setminus y}$	Reduction
LoRA	0.805	0.066	−92%
SineLoRA	0.734	0.063	−91%
RandLoRA	0.501	0.047	−91%
DoRA	0.806	0.067	−92%

To verify that TMKL directly controls the quantity it regularises, we measure non-target output drift on the held-out target test set:

$$D_{\setminus y} \;=\; \mathbb{E}_{t\in\mathcal{M}}\Big[\mathrm{KL}\Big(p^{\setminus y_t}_{\mathrm{base},t}\;\Big\|\;p^{\setminus y_t}_{\mathrm{adapted},t}\Big)\Big].$$

This is the validation-set version of the TMKL training objective. If TMKL is doing what the derivation in Section 3.2 claims, then this number should drop dramatically under TMKL training relative to CE training, regardless of any retention-set perplexity.

The same magnitude and direction as the retention-PPL prevention in Table 1, computed on a completely independent quantity (no retention data is involved in the probe), supports the interpretation that the retention gains come from suppressing non-target output drift rather than from under-adaptation. Figure 3 (in Section 4.5) visualises both the λ monotonicity and the held-out $D_{\setminus y}$ reduction graphically.

G Method Derivations and Local Interpretation

This appendix collects the derivations referenced in Section 3.1 and Section 3.2: the LoRA-family parameterization recalled for completeness, the algebraic derivation of the full-KL decomposition (Equation (2)), and a local geometric interpretation of $\mathcal{L}_{\setminus y}$ as a Fisher-weighted Jacobian penalty in the LoRA-admissible update space.

G.1 LoRA-Family Adapter Parameterization

For each selected linear layer with pretrained weight $W^0_\ell \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}$, vanilla LoRA introduces a trainable low-rank update

$$W_\ell(\phi) = W^0_\ell + s_\ell B_\ell A_\ell, \qquad A_\ell \in \mathbb{R}^{r\times d_{\mathrm{in}}}, \quad B_\ell \in \mathbb{R}^{d_{\mathrm{out}}\times r},$$

where $r \ll \min(d_{\mathrm{in}}, d_{\mathrm{out}})$, $s_\ell = \alpha/r$ is the LoRA scaling factor, and $\phi = \{A_\ell, B_\ell\}_\ell$ collects all trainable adapter parameters. The pretrained weights θ_0 remain frozen throughout adaptation. For the other LoRA-family adapters evaluated in this paper (AdaLoRA, VeRA, DoRA, SineLoRA, PiSSA, RandLoRA), ϕ denotes the corresponding trainable adapter parameters in each case. Target-Masked KL does not depend on the specific adapter parameterization: it only requires the adapted model distribution $p_{\mathrm{adapted},t}$.

G.2 Derivation of the Full-KL Decomposition

We restate Equation (2): for two distributions p_b, p_a on a finite vocabulary $\mathcal{V}$ and a designated target token $y \in \mathcal{V}$, with renormalized non-target distributions $p_b^{\setminus y}(c) = p_b(c)/(1-p_b(y))$ and $p_a^{\setminus y}(c) = p_a(c)/(1-p_a(y))$ for $c \neq y$,

$$\begin{aligned}\mathrm{KL}(p_b\,\|\,p_a) = p_b(y)\log\frac{p_b(y)}{p_a(y)} + \big(1-p_b(y)\big)\log\frac{1-p_b(y)}{1-p_a(y)} \\ + \big(1-p_b(y)\big)\,\mathrm{KL}\Big(p_b^{\setminus y}\;\Big\|\;p_a^{\setminus y}\Big).\end{aligned} \tag{3}$$

Derivation. Split the KL sum at the target token:

$$\mathrm{KL}(p_b \,\|\, p_a) = \sum_{c\in\mathcal{V}} p_b(c) \log \frac{p_b(c)}{p_a(c)} = p_b(y) \log \frac{p_b(y)}{p_a(y)} + \sum_{c\neq y} p_b(c) \log \frac{p_b(c)}{p_a(c)}.$$

For $c \neq y$, by definition of the renormalized distributions,

$$p_b(c) = \big(1 - p_b(y)\big)\, p_b^{\setminus y}(c), \qquad p_a(c) = \big(1 - p_a(y)\big)\, p_a^{\setminus y}(c).$$

Substituting,

$$\log \frac{p_b(c)}{p_a(c)} = \log \frac{1 - p_b(y)}{1 - p_a(y)} + \log \frac{p_b^{\setminus y}(c)}{p_a^{\setminus y}(c)}.$$

Multiplying by $p_b(c) = (1 - p_b(y))\, p_b^{\setminus y}(c)$ and summing over $c \neq y$,

$$\sum_{c\neq y} p_b(c) \log \frac{p_b(c)}{p_a(c)} = \big(1 - p_b(y)\big) \log \frac{1 - p_b(y)}{1 - p_a(y)} \cdot \underbrace{\sum_{c\neq y} p_b^{\setminus y}(c)}_{=1} + \big(1 - p_b(y)\big) \underbrace{\sum_{c\neq y} p_b^{\setminus y}(c) \log \frac{p_b^{\setminus y}(c)}{p_a^{\setminus y}(c)}}_{=\mathrm{KL}(p_b^{\setminus y} \| p_a^{\setminus y})}.$$

Combining with the target-token term gives Equation (3). □

Each term is non-negative when summed in the natural pairing: terms (i) and (ii) together form the binary KL divergence between $\big(p_b(y), 1 - p_b(y)\big)$ and $\big(p_a(y), 1 - p_a(y)\big)$, and term (iii) is a non-negative weighted KL on the non-target simplex. Individually, terms (i) and (ii) can take either sign; only their sum is constrained to be non-negative.

Lemma (binary-KL gradient opposes cross-entropy). Let $K(p_b, p_a) := p_b(y) \log \frac{p_b(y)}{p_a(y)} + (1 - p_b(y)) \log \frac{1-p_b(y)}{1-p_a(y)}$ denote the sum of terms (i) and (ii) in Equation (3), viewed as a function of $p_a(y)$ at fixed $p_b(y) \in (0, 1)$. Then

$$\frac{\partial K}{\partial p_a(y)} = \frac{p_a(y) - p_b(y)}{p_a(y)\big(1 - p_a(y)\big)},$$

which is strictly positive whenever $p_a(y) > p_b(y)$. The cross-entropy gradient $\partial_{p_a(y)}\big(-\log p_a(y)\big) = -1/p_a(y)$ is strictly negative. Hence whenever the adapted target probability exceeds the base target probability (the regime cross-entropy actively drives toward under distribution shift), the binary-KL gradient and the cross-entropy gradient have opposite signs, and a full-distribution KL regularizer fights cross-entropy on the very token cross-entropy is trying to learn. Target-Masked KL drops K from the regularizer by construction (Equation (1)) and therefore avoids the conflict globally, not just locally near $p_a(y) \approx p_b(y)$.

Derivation of the lemma. Differentiating K in $p_a(y)$ at fixed $p_b(y)$:

$$\frac{\partial K}{\partial p_a(y)} = -\frac{p_b(y)}{p_a(y)} + \frac{1 - p_b(y)}{1 - p_a(y)} = \frac{-p_b(y)\big(1 - p_a(y)\big) + p_a(y)\big(1 - p_b(y)\big)}{p_a(y)\big(1 - p_a(y)\big)} = \frac{p_a(y) - p_b(y)}{p_a(y)\big(1 - p_a(y)\big)},$$

which has the sign of $p_a(y) - p_b(y)$. □

G.3 Local LoRA-Space Interpretation as a Fisher-Weighted Jacobian Penalty

We give a local interpretation of $\mathcal{L}_{\setminus y}$ as a quadratic form on the adapter perturbation $\delta\phi$, with curvature determined by the LoRA Jacobian and the Fisher information of the non-target categorical distribution.

Let $z_{\text{adapted},t} \in \mathbb{R}^{|\mathcal{V}|}$ denote the adapted logits at token position t, and let $z^{\setminus y_t}_{\text{adapted},t}$ denote the logits restricted to the non-target vocabulary. Around the frozen base model, a first-order expansion in the adapter parameters gives

$$z^{\setminus y_t}_{\text{adapted},t} \approx z^{\setminus y_t}_{\text{base},t} + J^{\phi}_{t,\setminus y_t}\,\delta\phi,$$

where $J^{\phi}_{t,\setminus y_t}$ is the Jacobian of the non-target logits with respect to ϕ. Because the base weights are frozen, this Jacobian contains only directions reachable by the LoRA adapters; this is what we mean by "LoRA-admissible" updates. Note that for a softmax output, the renormalized non-target distribution $p^{\setminus y}$ equals the softmax of the non-target logit subvector $z^{\setminus y}$ (the target logit drops out of the partition function), so the Fisher information of $p^{\setminus y}$ with respect to $z^{\setminus y}$ is the standard categorical Fisher of the non-target distribution.

For small adapter-induced logit changes, the KL divergence between two categorical distributions admits a standard second-order local approximation in the logits (the first-order term vanishes at $z_a = z_b$ since $\partial_{z_a}\mathrm{KL}(p_b\|p_a) = p_a - p_b$, and the Hessian $\partial^2_{z_a}\mathrm{KL}(p_b\|p_a) = F(p_a)$ equals $F(p_b)$ at the expansion point):

$$\mathrm{KL}\Big(p^{\setminus y_t}_{\text{base},t}\,\Big\|\,p^{\setminus y_t}_{\text{adapted},t}\Big) \approx \frac{1}{2}\,\delta\phi^{\top}\left(J^{\phi}_{t,\setminus y_t}\right)^{\top} F\Big(p^{\setminus y_t}_{\text{base},t}\Big)\,J^{\phi}_{t,\setminus y_t}\,\delta\phi,$$

where

$$F\Big(p^{\setminus y_t}_{\text{base},t}\Big) = \mathrm{Diag}\Big(p^{\setminus y_t}_{\text{base},t}\Big) - p^{\setminus y_t}_{\text{base},t}\left(p^{\setminus y_t}_{\text{base},t}\right)^{\top}$$

is the Fisher information matrix of the non-target categorical distribution with respect to its logits. The expansion is valid for $\|\delta\phi\|$ small enough that $\|z^{\setminus y_t}_{\text{adapted},t} - z^{\setminus y_t}_{\text{base},t}\|$ remains in the regime where the second-order Taylor expansion of KL in logits is accurate.

Under this approximation, Target-Masked KL penalizes adapter directions $\delta\phi$ that, after projection through the LoRA Jacobian, induce large changes in the base model's non-target predictive geometry. Two consequences follow. First, Target-Masked KL is a local output-space regularizer rather than a parameter-space constraint: the penalty depends on the LoRA adapters only through their effect on the non-target logits. Second, LoRA determines the admissible update directions through $J^{\phi}_{t,\setminus y_t}$, and Target-Masked KL discourages those directions from inducing large non-target distributional changes. Together these two observations explain why Target-Masked KL is compatible with any LoRA-family parameterization that exposes a differentiable map from ϕ to the adapted output distribution.

Strong-regularisation scaling law and empirical check. The same quadratic approximation makes a non-trivial prediction. Writing $H = (1/|\mathcal{M}|)\sum_{t\in\mathcal{M}}\big(J^{\phi}_{t,\setminus y_t}\big)^{\top} F\big(p^{\setminus y_t}_{\text{base},t}\big) J^{\phi}_{t,\setminus y_t}$, in the strong-regularization regime the equilibrium adapter perturbation $\delta\phi^*$ that minimizes $\mathcal{L}_{\mathrm{CE}} + \lambda\,\mathcal{L}_{\setminus y}$ satisfies $\lambda H\delta\phi^* \approx -\nabla\mathcal{L}_{\mathrm{CE}}$ (in the column space of H, where the LoRA-admissible directions live), so $\|\delta\phi^*\| \propto 1/\lambda$. Retention drift in held-out PPL is, to first order, linear in $\delta\phi^*$ around the base, so $\Delta\mathrm{PPL} \approx \nabla\mathrm{PPL}^{\top}\delta\phi^* \propto 1/\lambda$. (Note that the regularizer *value* itself, $\mathcal{L}_{\setminus y}(\delta\phi^*) = \frac{1}{2}(\delta\phi^*)^{\top}H\delta\phi^* \propto 1/\lambda^2$, is one order steeper than the drift; we do not test the regularizer value, only the drift.) This predicts a log-log slope of -1 for residual retention drift versus λ in the strong-regularization regime. Fitting the LoRA λ-sweep in Table 15 for $\lambda \in [0.3, 1]$ gives an empirical WT-103 slope of -0.94, within 6% of the theoretical -1; LAMBADA gives -1.35, consistent in direction but suggesting beyond-leading-order Fisher structure on LAMBADA positions. The closeness of the WT-103 slope to the predicted value is direct empirical support for the local-geometry interpretation.

G.4 Pinsker bound: from $\mathcal{L}_{\setminus y}$ to a capability-retention guarantee

The held-out output-drift probe (Table 27) shows empirically that minimising $\mathcal{L}_{\setminus y}$ at training time tightly controls the same quantity at evaluation time. The link between this surrogate and a guarantee on *any* downstream capability that depends on the non-target distribution is one application of Pinsker's inequality (Cover, 1999, Lemma 11.6.1). For each supervised position t,

$$\tfrac{1}{2}\,\big\|p^{\setminus y_t}_{\text{base},t} - p^{\setminus y_t}_{\text{adapted},t}\big\|^2_{\mathrm{TV}} \;\le\; \mathrm{KL}\big(p^{\setminus y_t}_{\text{base},t}\,\big\|\,p^{\setminus y_t}_{\text{adapted},t}\big).$$

Averaging over $\mathcal{M}$ and applying Jensen's inequality,

$$\mathbb{E}_{t\in\mathcal{M}}\Big[\big\|p^{\setminus y_t}_{\text{base},t} - p^{\setminus y_t}_{\text{adapted},t}\big\|_{\mathrm{TV}}\Big] \;\le\; \sqrt{2\,\mathcal{L}_{\setminus y}}.$$

Any bounded statistic of the conditional non-target distribution (e.g., the probability assigned to a specified non-target candidate, or a likelihood ratio between two non-target alternatives) therefore deviates by at most $\sqrt{2\mathcal{L}_{\setminus y}}$ in expectation. With our held-out values of $\mathcal{L}_{\setminus y} \approx 0.05$ to 0.07 nats under TMKL (Table 27), this bound is ≤ 0.37 TV per position. The bound is loose (Pinsker is tight only for binary distributions), but it makes the surrogate-to-capability link explicit: any downstream task whose decision rule is a bounded function of the conditional non-target distribution inherits a TV-distance guarantee from the same surrogate that TMKL minimises. This is information-theoretically consistent with the empirical preservation of factual recall, math reasoning, code, and multilingual capabilities under TMKL (Table 7).

G.5 Training-Time Cost

Computing $\mathcal{L}_{\setminus y}$ requires one forward pass through the frozen base model f_{θ_0} in addition to the standard forward and backward passes through the LoRA-adapted model. The base-model forward pass is cached once per training batch, has no associated backward pass (no gradients are propagated through θ_0), and can be run in inference mode (no activation memory for backward, optional half-precision). The renormalization and KL computation themselves are $O(|\mathcal{V}|)$ per supervised position and add negligible overhead relative to the forward pass. At inference time the regularizer is discarded entirely, and the deployed adapted model is identical in form to one trained with cross-entropy alone.