TOWARDS BUILDING NON-FINE-TUNABLE FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-sourcing foundation models (FMs) enables broad reuse but also exposes model trainers to economic and safety risks from unrestricted downstream fine-tuning. We address this problem by building *non-fine-tunable* foundation models: models that remain broadly usable in their released form while yielding limited adaptation gains under task-agnostic unauthorized fine-tuning. We propose **Private Mask Pre-Training (PMP)**, a pre-training framework that concentrates representation learning into a sparse subnetwork identified early in training. The binary mask defining this subnetwork is kept private, and only the final dense weights are released. This forces unauthorized fine-tuning without access to the mask to update parameters misaligned with pretraining subspace, inducing an intrinsic mismatch between the fine-tuning objective and the pre-training geometry. We provide theoretical analysis showing that this mismatch destabilizes gradient-based adaptation and bounds fine-tuning gains. Empirical results on large language models demonstrating that PMP preserves base model performance while consistently degrading unauthorized fine-tuning across a wide range of downstream tasks, with the strength of non-fine-tunability controlled by the mask ratio.

1 INTRODUCTION

Foundation models (FMs), particularly large language models (LLMs), have achieved remarkable progress in recent years. Through large-scale pre-training on diverse data, these models exhibit strong general-purpose capabilities in language understanding, generation Achiam et al. (2023); Team et al. (2024); Comanici et al. (2025), and reasoning Guo et al. (2025); Jaech et al. (2024), and have become a core infrastructure for modern AI systems. Their impact now extends far beyond research, reshaping everyday applications such as information access, software development, education, and creative work. Training these models requires enormous investments in data, computation, and human expertise. In practice, FM trainers have followed two primary paths: keeping models closed and monetizing access through APIs Achiam et al. (2023); Comanici et al. (2025), or releasing pre-trained weights publicly Guo et al. (2025); Touvron et al. (2023) to accelerate scientific innovation and community influence.

While open-weight releases have benefited the ecosystem by enabling advances in efficiency Lin et al. (2024), interpretability Singh et al. (2024), and post-training optimization Hu et al. (2022); Yu et al. (2025), At the same time, they introduce two fundamental challenges for FM trainers. Economically, once a FM is released, users can perform low-cost adaptations on downstream tasks, effectively capturing value that would otherwise belong to the original trainer. From a safety perspective, trainers lose control over the model's downstream lifecycle: through unauthorized fine-tuning, an FM can be retrained on harmful or illegal data, such as content involving violence, extremism, discrimination, or other sensitive domains Li et al. (2026); Poppi et al. (2024); Wang et al. (2024). The resulting models may generate harmful outputs while still being associated with the original pre-trained FM, potentially damaging the trainer's reputation and exposing them to legal or regulatory risk. *These issues highlight the absence of mechanisms that allow trainers to open-source FMs while retaining meaningful control over adaptation.*

Motivated by these concerns, we consider the problem of building **non-fine-tunable FMs**: pre-trained foundation models that remain broadly usable in their released form, but for which standard gradient-based fine-tuning without authorization yields substantially reduced gains. To reconcile open-source

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.

utility with developer control, a non-fine-tunable FM must satisfy three core desiderata: ❶ **Capability preservation**: the defense must not degrade the core performance, accuracy, or generation quality of the base model. ❷ **Controllability**: unauthorized fine-tuning should be ineffective, while "authorized" adaptation by trusted users remains possible. ❸ **Black-box generality**: the FM trainer cannot assume any prior knowledge of the downstream user's data, objectives, or optimization strategies. Existing defenses often rely on task-specific unlearning or "immunization" strategies that solve a mini-max problem to suppress specific harmful domains Deng et al. (2024); Wang et al. (2024). These methods share a fundamental limitation: *they require access to the data or task objectives they intend to restrict*. Such assumptions are unrealistic in real-world scenarios where trainers have no visibility into how a released model will be repurposed.

To address these challenges, we propose **Private Mask Pre-Training (PMP)**, a pre-training paradigm that induces non-fine-tunability as an intrinsic property of the model's pre-training geometry. PMP first runs a short warm-up pre-training phase on a small subset of data and uses an early-bird lottery ticket criterion Chen et al. (2021) to select a sparse binary mask that identifies a stable subnetwork. The mask is then kept private and applied throughout pre-training so that all gradient updates are restricted to the masked parameters, while the masked-out parameters remains frozen. Although the forward pass uses the full dense parameterization, representation learning is concentrated into the selected sparse subnetwork. After convergence, the model trainer releases only the final dense weights and does not disclose the mask. Without access to the mask, unauthorized fine-tuning cannot restrict updates to the true subnetwork and is forced to co-update both the trained ticket and the untrained complement, creating an intrinsic mismatch between the conditional objective solved during pre-training and the joint objective optimized during fine-tuning. This mismatch destabilizes optimization and significantly degrades adaptation, as formalized by our theoretical analysis. Importantly, our experiments demonstrate that with a reasonable mask ratio (e.g., 70%), PMP preserves the pre-trained performance of the base model, while substantially degrading unauthorized fine-tuning across diverse downstream tasks, datasets, and optimization settings, without requiring any prior knowledge of downstream data. When the private mask is provided, authorized fine-tuning can partially mitigate the degradation caused by PMP, often achieving better performance than unauthorized fine-tuning.

We summarize our main contributions as follows:

- We identify non-fine-tunability as a fundamental pre-training-level problem for FMs and articulate the key challenges faced by model trainers in controlling unauthorized fine-tuning.
- We propose PMP, a general pre-training paradigm that preserves base model utility, enables authorized fine-tuning, and restricts unauthorized fine-tuning in a task-agnostic black-box setting.
- We provide a theoretical analysis demonstrating that PMP induces an intrinsic optimization objective mismatch that injects destabilizing perturbations during unauthorized updates.
- We empirically validate PMP on LLMs, showing that it consistently suppresses adaptation gains across diverse benchmarks while maintaining competitive pre-trained performance.

2 METHOD

We propose PMP to obtain a non-fine-tunable FM. As shown in Figure 1, the core idea is to concentrate pre-training into a sparse subnetwork identified early in training, while keeping the corresponding mask private. After pre-training, only the final dense parameters are released. Without access to the mask, downstream fine-tuning is forced to update the full parameter vector in a way that is misaligned with how the model was pre-trained, which makes optimization unstable and can substantially deteriorate fine-tuning outcomes.

PMP proceeds in three stages: (I) private early-bird mask discovery that identifies a sparse ticket; (II) private mask pre-training that updates only the ticket parameters; and (III) releasing final weights without the mask.

2.1 PRIVATE EARLY-BIRD MASK DISCOVERY

Let model parameters be $\theta \in \mathbb{R}^d$ with initialization $\theta^{(0)}$. We seek a binary mask $M \in \{0,1\}^d$ at sparsity ratio $\rho \in (0,1)$; denote its complement by $\bar{M} = \mathbf{1} - M$. Masked parameters are defined via

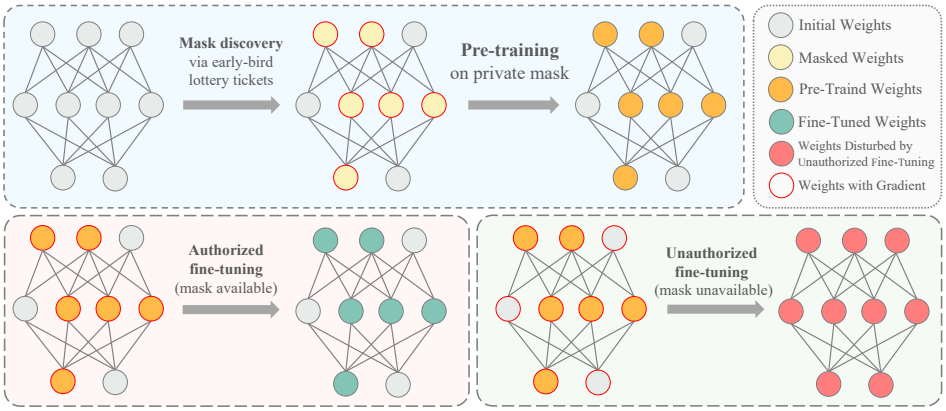


Figure 1: Overview of Private Mask Pretraining (PMP). Top: the foundation model trainer discovers an Early-Bird lottery ticket and performs pretraining by restricting updates to a private mask. Bottom: after model release, authorized fine-tuning with access to the mask updates only the pretrained ticket and enables effective adaptation, while unauthorized fine-tuning without the mask perturbs both trained and untrained parameters, leading to degraded performance.

element-wise product $\odot$:

$$\theta_M = \theta \odot M, \qquad \theta_{\bar{M}} = \theta \odot \bar{M}, \qquad \|M\|_0 = \rho d. \tag{1}$$

Using a small warm-up subset $\mathcal{D}_{\text{pre}}$, we identify an early-bird lottery ticket by extracting a stable global top-ρ mask from gradient magnitudes. In the warm-up step t, let $\mathcal{B}^{(t)} \subset \mathcal{D}_{\text{pre}}$ denote the mini-batch sampled at step t, and let

$$g^{(t)} = \left|\nabla_\theta \mathcal{L}(\theta^{(t)}; \mathcal{B}^{(t)})\right| \in \mathbb{R}^d_{\geq 0}$$

be the element-wise absolute gradient. We form a candidate mask $M^{(t)} \in \{0,1\}^d$ by keeping exactly the top-k entries of $g^{(t)}$ globally across all parameters, where $k = \lfloor \rho d \rfloor$:

$$M_i^{(t)} = \mathbf{1}\left[g_i^{(t)} \geq \text{TopK}\big(g^{(t)}, k\big)\right], \qquad i = 1, \ldots, d, \tag{2}$$

where $\text{TopK}(g, k)$ returns the k-th largest value of g (ties broken arbitrarily to ensure $\|M^{(t)}\|_0 = k$). We declare the early-bird ticket once consecutive masks become stable, measured by the IoU

$$\text{IoU}(M^{(t)}, M^{(t-1)}) = \frac{\|M^{(t)} \odot M^{(t-1)}\|_0}{\|M^{(t)} \vee M^{(t-1)}\|_0}, \tag{3}$$

and select the first $M^{(t)}$ that remains above a threshold for several successive steps. The resulting mask M is then fixed and kept private for all subsequent training:

$$M \leftarrow \text{EarlyBird}\big(\{\nabla \mathcal{L}(\theta^{(t)}; \mathcal{B}^{(t)})\}_{t \leq t_{\text{EB}}}, \rho\big). \tag{4}$$

2.2 Private Mask Pre-Training

Let $\mathcal{D}_{\text{train}}$ be the full pre-training dataset. Starting from $\theta^{(0)}$, PMP performs standard pre-training while restricting optimization to the ticket parameters θ_M. At each iteration, a mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ is sampled, and gradient descent is applied only to θ_M, while the complement $\theta_{\bar{M}}$ is kept fixed:

$$\theta_M \leftarrow \theta_M - \eta \nabla_{\theta_M} \mathcal{L}\big(\theta_M + \theta_{\bar{M}}; \mathcal{B}\big), \quad \theta_{\bar{M}} \leftarrow \theta_{\bar{M}}. \tag{5}$$

Although the forward computation uses the full parameter vector, all parameter updates are confined to the sparse ticket defined by M. This concentrates learned representations into a low-dimensional subspace determined early in training, while leaving the complement parameters unadapted.

2.3 RELEASE WITHOUT MASK

At convergence, we publish only the final parameters and keep the mask M private. Since downstream users do not know which parameters constitute the true ticket, fine-tuning necessarily updates the full parameter vector. This creates a mismatch between the constrained optimization used during pre-training (updates restricted to θ_M) and the unconstrained optimization used during adaptation (updates applied to both θ_M and $\theta_{\bar{M}}$), which can destabilize optimization and lead to substantially worse fine-tuning results. We next provide a theoretical explanation of this mismatch effect.

3 THEORETICAL ANALYSIS

We provide a theoretical explanation for why fine-tuning a PMP-pre-trained model without access to the private mask can become unstable and yield only limited adaptation gains. Our analysis formalizes *non-fine-tunability* as a property of the achievable fine-tuning improvement under unknown-mask optimization, and explains how PMP enforces this property through an intrinsic optimization mismatch.

We decompose parameters as $\theta = (\theta_M, \theta_{\bar{M}})$ using the mask in Eq. equation 1, and define the expected pre-training objective

$$L_{\mathrm{pre}}(\theta_M, \theta_{\bar{M}}) \;=\; \mathbb{E}_{z\sim\mathcal{D}_{\mathrm{train}}}\big[\ell(f(\theta_M + \theta_{\bar{M}}); z)\big]. \tag{6}$$

Let $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$ denote the point produced by PMP, where $\theta^{(0)}_{\bar{M}}$ is the frozen complement throughout pre-training. We assume that PMP reaches a *conditional optimum*, where θ_M is optimal given fixed $\theta^{(0)}_{\bar{M}}$, but not necessarily optimal over the full parameter space:

$$\theta^\star_M \in \arg\min_{\theta_M} L_{\mathrm{pre}}(\theta_M, \theta^{(0)}_{\bar{M}}), \quad \nabla_{\theta_M} L_{\mathrm{pre}}(\theta^\star_M, \theta^{(0)}_{\bar{M}}) = 0.$$

Fine-tuning gain and non-fine-tunability. For a downstream task, define the fine-tuning objective as

$$L_{\mathrm{ft}}(\theta_M, \theta_{\bar{M}}) \;=\; L_{\mathrm{pre}}(\theta_M, \theta_{\bar{M}}) \;+\; \Delta(\theta_M, \theta_{\bar{M}}), \tag{7}$$

where Δ captures task-specific supervision or distributional shift. We measure fine-tuning effectiveness by the improvement in pre-training loss, $\mathrm{Gain} = L_{\mathrm{pre}}(\theta^\star_M, \theta^{(0)}_{\bar{M}}) - L_{\mathrm{pre}}(\theta_M, \theta_{\bar{M}})$, achieved through optimization on L_{ft}. A model is *non-fine-tunable* if this gain remains small under unknown-mask fine-tuning.

Assumption 1 (Asymmetric local geometry). In a neighborhood of $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$, the loss landscape of L_{pre} is comparatively flat along the θ_M directions due to extensive optimization, while exhibiting non-negligible positive curvature along the $\theta_{\bar{M}}$ directions, which were never adapted during PMP pre-training.

Proposition 1 (Instability and limited gains under unknown-mask fine-tuning). Consider one step of stochastic gradient descent on L_{ft} starting from $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$ with step size η:

$$\theta^+_M = \theta^\star_M - \eta g_M, \qquad \theta^+_{\bar{M}} = \theta^{(0)}_{\bar{M}} - \eta g_{\bar{M}},$$

where $(g_M, g_{\bar{M}})$ is a stochastic gradient of L_{ft} at $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$. Under Assumption 1 and for sufficiently small η,

$$\mathbb{E}\big[L_{\mathrm{pre}}(\theta^+_M, \theta^+_{\bar{M}})\big] \;\geq\; L_{\mathrm{pre}}(\theta^\star_M, \theta^{(0)}_{\bar{M}}) \;+\; c\,\eta^2, \tag{8}$$

for some constant $c > 0$ whenever $g_{\bar{M}} \neq 0$ with nonzero probability. As a result, when fine-tuning is performed without the private mask and induces nontrivial updates on the complement parameters $\theta_{\bar{M}}$, the optimization dynamics are *prone to destabilizing perturbations* of the pre-trained solution. Consequently, unknown-mask fine-tuning *often yields limited or unstable fine-tuning gains*, as progress driven by task-specific gradients is counteracted by curvature-induced degradation along previously unadapted directions.

Proof sketch. We perform a second-order Taylor expansion of L_{pre} around $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$:

$$L_{\mathrm{pre}}(\theta^+_M, \theta^+_{\bar{M}}) \approx L_{\mathrm{pre}}(\theta^\star_M, \theta^{(0)}_{\bar{M}}) - \eta\langle \nabla L_{\mathrm{pre}}, g\rangle + \frac{\eta^2}{2} g^\top H_{\mathrm{pre}} g,$$

Table 1: GLUE *dev* results (%). For each model (GPT-2, TinyLlama) and each stage (Base, LoRA fine-tuned, Full fine-tuned), we report results *w/o* PMP and with **Private Mask Pre-Training (PMP)** under identical pre-training and fine-tuning protocols. Base (↑): higher is better; Fine-Tuned (↓): lower is better.

Model	Stage	Variant	CoLA	SST-2	MRPC	QQP	STS-B	MNLI-m	MNLI-mm	QNLI	RTE
GPT-2	Base(↑)	W/O PMP	38.35	50.00	67.16	37.22	12.52	35.90	36.02	49.35	52.71
		PMP	50.34	50.00	58.63	37.42	11.88	35.00	35.10	48.20	51.00
	LoRA Fine-Tuned(↓)	W/O PMP	57.53	68.58	68.38	78.42	17.40	40.89	45.05	51.58	57.76
		PMP	**52.73**	**57.91**	**56.13**	**71.68**	**14.67**	**32.72**	**42.34**	**45.25**	**49.82**
	Full Fine-Tuned(↓)	W/O PMP	53.49	79.24	68.14	56.61	20.58	36.41	39.05	51.24	55.60
		PMP	**50.05**	**65.94**	**61.40**	**38.81**	**18.30**	**33.28**	**31.80**	**50.54**	**47.29**
TinyLlama	Base(↑)	W/O PMP	37.01	50.69	31.62	36.84	16.13	36.77	33.35	50.58	52.71
		PMP	63.66	50.80	40.20	38.66	16.53	32.81	32.60	50.72	53.79
	LoRA Fine-Tuned(↓)	W/O PMP	68.36	80.62	71.08	82.10	26.33	66.91	68.65	78.73	54.51
		PMP	**65.96**	**72.13**	**63.97**	**74.66**	**22.40**	**61.15**	**62.42**	**62.47**	**44.04**
	Full Fine-Tuned(↓)	W/O PMP	68.36	83.14	71.32	86.51	32.47	71.17	71.34	79.37	54.15
		PMP	**61.64**	**76.03**	**69.36**	**78.20**	**22.43**	**59.52**	**58.48**	**63.75**	**46.57**

where $g = (g_M, g_{\bar{M}})$ and H_{pre} is the Hessian of L_{pre} at $(\theta^\star_M, \theta^{(0)}_{\bar{M}})$. Since $\nabla_{\theta_M} L_{\text{pre}}(\theta^\star_M, \theta^{(0)}_{\bar{M}}) = 0$, the first-order term depends only on the complement component $g_{\bar{M}}$ and has no guaranteed sign. Under Assumption 1, the Hessian block corresponding to $\theta_{\bar{M}}$ exhibits positive curvature, so the quadratic term contributes positively in expectation whenever $g_{\bar{M}} \neq 0$. This explains why unconstrained fine-tuning that co-updates $\theta_{\bar{M}}$ tends to destabilize the pre-trained solution and limits effective adaptation without access to the private mask.

4 Experiments

4.1 Experimental Set-Ups

Model, data, and initialization. We pre-train a LLaMA-style decoder-only Transformer with a TinyLlama-1.1B–scale configuration (22 layers, hidden size 2048, 32 attention heads, 4 KV heads, intermediate size 5632, RoPE $\theta = 10{,}000$) from random initialization. Pre-training is performed on SlimPajama-6B (`DKYoon/SlimPajama-6B`, `text` field) using the HuggingFace streaming pipeline with online shuffling (buffer size 50k). Documents are tokenized without truncation, separated by an EOS token, and packed into fixed-length blocks of 256 tokens. Each block is trained with the standard causal language modeling objective, with labels equal to input IDs. Both Base and PMP models use the same architecture and training budget.

PMP procedure. We run $t_{\text{EB}} = 500$ warm-up iterations and construct a global exact Top-k mask from element-wise absolute gradients, with sparsity $\rho = 0.70$. We select the earliest mask whose IoU is at least 0.99 for 5 steps.

Downstream fine-tuning. We evaluate fine-tunability on GLUE Wang et al. (2018) by fine-tuning each released checkpoint independently on the official training split and reporting results on the validation split. For tasks whose primary metric is accuracy (SST-2, MNLI matched/mismatched, QNLI, and RTE), we follow the standard GLUE protocol and report accuracy; for pairwise sentence classification tasks (MRPC and QQP), we report the standard GLUE metrics (accuracy and F1). However, some GLUE tasks adopt canonical metrics that are not on the same scale as accuracy (e.g., MCC for CoLA and Pearson/Spearman correlation for STS-B), making direct cross-task aggregation (e.g., an average score) ill-posed. To enable a unified, accuracy-based aggregate across tasks, we additionally introduce an *accuracy-style auxiliary metric* for CoLA and STS-B by mapping their outputs to a form amenable to accuracy computation. This auxiliary metric is used solely to improve cross-task comparability and aggregation, and is not intended to replace the canonical task metrics. Specifically, for **CoLA** (binary grammatical acceptability), we take the argmax over the two logits to obtain the predicted label $\hat{y}$ and compute accuracy as the fraction of examples with $\hat{y} = y$. For **STS-B** (semantic textual similarity; regression with gold scores in $[0, 5]$), we discretize both predictions

and gold scores by clipping to $[0, 5]$ and rounding to the nearest integer, yielding ordinal labels in $\{0, 1, 2, 3, 4, 5\}$, and then compute standard classification accuracy on these discretized labels.

4.2 The Non-Fine-Tunability of PMP

We evaluate the non-fine-tunability of PMP on the GLUE development benchmarks, comparing models trained with and without PMP under identical fine-tuning protocols. Table 1 reports results for GPT-2 and TinyLlama, where we separately examine base performance before fine-tuning and performance after fine-tuning without access to the private mask. For base models, higher values indicate better performance, while for fine-tuned models, lower values indicate worse adaptation.

Across both architectures, PMP preserves the pretrained capability of the released base models. For GPT-2, PMP achieves base performance comparable to training without PMP across all GLUE tasks, with only minor fluctuations that do not indicate systematic degradation. Similarly, for TinyLlama, PMP maintains strong base performance and even improves results on several tasks such as CoLA and MRPC, demonstrating that concentrating representation learning into a sparse ticket during pretraining does not harm general-purpose capability. In contrast, fine-tuning behavior differs markedly. When fine-tuning is performed without access to the private mask, models pretrained with PMP consistently exhibit substantially worse fine-tuned performance than their non-PMP counterparts. For GPT-2, PMP leads to clear degradation across nearly all tasks, including SST-2, MRPC, QQP, MNLI, QNLI, and RTE. The same pattern is observed for TinyLlama, where PMP results in significantly lower fine-tuned scores across the full suite of GLUE benchmarks. Notably, this degradation is broad and consistent rather than confined to a small subset of tasks, indicating a general failure of effective adaptation rather than task-specific brittleness.

Overall, these results demonstrate a clear decoupling between pretrained capability and fine-tunability. While PMP-pretrained models remain competitive foundation models in their released form, unauthorized fine-tuning without the private mask becomes substantially less effective across architectures and tasks. This empirical evidence supports PMP as a practical pretraining-level mechanism for inducing non-fine-tunability without sacrificing base model quality.

4.3 Authorized Fine-Tuning Performance

A central goal of PMP is to preserve practical usability while enabling *authorized* adaptation. In this setting, the model owner provides the private mask to the user, who then fine-tunes by updating only the masked subspace (i.e., θ_M) while keeping the complement fixed. We compare this authorized procedure against *unauthorized* fine-tuning, which applies standard full-parameter updates without the private mask. Figure 2 shows the resulting GLUE performance on TinyLlama. Authorized fine-tuning consistently matches or exceeds unauthorized fine-tuning across tasks, with the largest gaps appearing on CoLA, QQP, STS-B, and MNLI. This trend supports our core hypothesis: when the fine-tuning updates are aligned with the pre-trained ticket (authorized access), adaptation is more stable and effective, whereas unconstrained full-space updates without the mask tend to underperform due to the induced optimization mismatch.

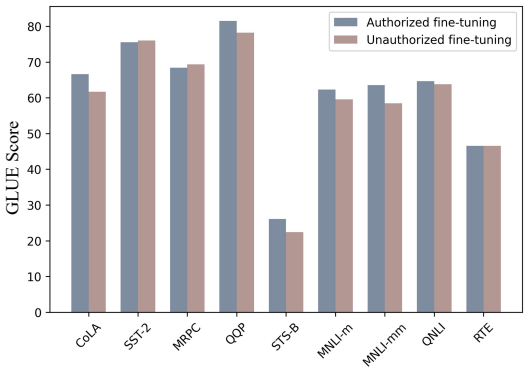
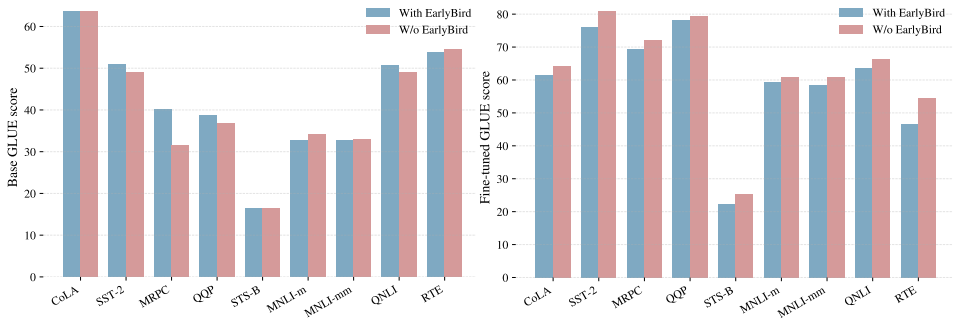


Figure 2: **Authorized vs. unauthorized fine-tuning on GLUE (TinyLlama).** Authorized fine-tuning (with access to the private mask) updates only the masked subspace, while unauthorized fine-tuning performs standard full-parameter updates without the mask. Authorized fine-tuning consistently outperforms unauthorized fine-tuning.

4.4 Ablation Studies

Early-Bird Mask Selection. We further analyze the effect of EarlyBird mask selection by explicitly studying its role as an early-bird lottery ticket in PMP. In this experiment, we compare PMP using the EarlyBird lottery ticket mask discovered during warm-up with a variant that uses a randomly sampled



(a) Base Acc. Higher is better.

(b) Fine-tuned Acc. Lower is better.

Figure 3: **Effect of EarlyBird mask selection on TinyLlama in PMP.** We compare PMP with EarlyBird and PMP with random mask selection on (a) base performance and (b) fine-tuned performance.

mask of the same sparsity ratio ρ. All other components, including the model architecture, pretraining schedule, and unauthorized fine-tuning protocol, are kept identical. As shown in Figure 3, both variants maintain comparable base performance, confirming that sparsity alone does not inherently degrade pretrained capability. Notably, PMP with the EarlyBird lottery ticket mask consistently achieves equal or slightly better base performance across most GLUE tasks, indicating that the early-bird ticket captures a subnetwork that is particularly effective for representation learning during pretraining.

The impact of EarlyBird selection is substantially more pronounced under fine-tuning. When the EarlyBird lottery ticket mask is replaced by a random mask, fine-tuned performance improves across most tasks, suggesting that the model becomes easier to adapt under unauthorized fine-tuning. In contrast, PMP with the EarlyBird lottery ticket mask yields consistently lower fine-tuned scores, demonstrating a stronger non-fine-tunability effect. This behavior aligns with the lottery ticket hypothesis: the EarlyBird mask identifies a sparse subnetwork that emerges early and remains stable throughout training, concentrating pretrained representations into a well-organized parameter subspace. As a consequence, pretraining becomes more effective within this ticket, while unauthorized fine-tuning, which cannot target the same subspace, is more likely to disrupt the learned structure and result in degraded adaptation. Together, these results show that using an EarlyBird lottery ticket mask is critical for simultaneously improving pretraining effectiveness and amplifying the non-fine-tunability induced by PMP.

Mask Ratio. The mask ratio determines the fraction of parameters included in the pretrained ticket and directly controls the strength of non-fine-tunability. Figure 4 (left) shows QNLI base and fine-tuned accuracy as a function of the mask ratio, where 1.0 corresponds to training without PMP. Without PMP, fine-tuning yields substantial gains, indicating strong adaptation capacity. In contrast, once PMP is applied, fine-tuning improvements are sharply suppressed and remain limited across mask ratios from 0.9 to 0.5, while base performance remains largely stable. Among these settings, a mask ratio of 0.7 achieves the best trade-off between base accuracy and fine-tuning suppression, and is therefore used as the default throughout our experiments. Overall, these results demonstrate that the mask ratio serves as a continuous control knob that modulates the degree of fine-tuning suppression induced by PMP.

Fine-Tuning Learning Rate. We examine whether the degradation induced by PMP can be mitigated by tuning the fine-tuning learning rate. Figure 4 (middle) reports the average GLUE performance under three commonly used learning rates, comparing standard fine-tuning without PMP and fine-tuning on PMP models. Without PMP, fine-tuning performance remains consistently high across learning rates, indicating that standard pretraining supports robust adaptation under typical hyperparameter choices. In contrast, PMP consistently suppresses fine-tuning performance for all learning rates considered. The performance gap between PMP and the baseline remains largely unchanged as the learning rate varies, and increasing the learning rate does not recover fine-tuning effectiveness. This shows that the non-fine-tunability induced by PMP is not a consequence of suboptimal hyperparameter selection, but rather reflects a fundamental mismatch between the

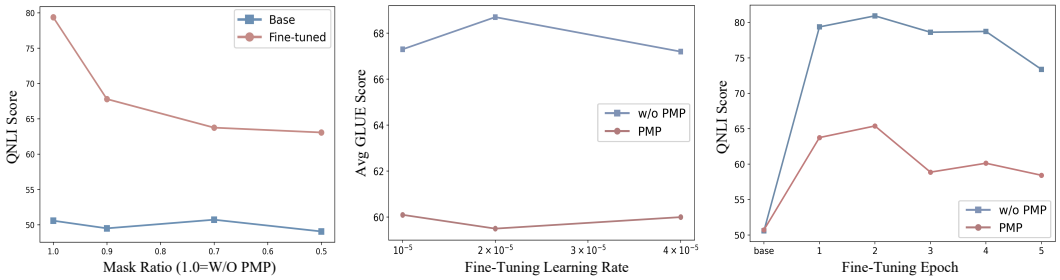


Figure 4: **Impact of Mask Ratio and Optimization Hyperparameters on Fine-Tuning Performance.** The left panel shows QNLI performance under different mask ratios, where standard fine-tuning yields substantial gains while PMP consistently suppresses adaptation with minimal impact on performance. The middle panel evaluates sensitivity to the fine-tuning learning rate, and the right panel reports performance across fine-tuning epochs. Across all settings, PMP consistently limits fine-tuning and increasing optimization strength does not recover performance.

pretraining and fine-tuning optimization objectives. Consequently, common learning-rate tuning strategies are insufficient to bypass PMP.

Fine-Tuning Epoch. We further examine the effect of fine-tuning duration by varying the number of training epochs. As shown in Figure 4 (right), standard fine-tuning rapidly improves downstream performance but degrades with over-training, whereas PMP consistently suppresses adaptation across epochs. Notably, increasing the number of fine-tuning epochs does not recover performance under PMP, indicating that its effect is not sensitive to training duration.

5 Discussion and Analysis

5.1 Understanding PMP via Loss Landscape

To empirically validate the *Optimization Mismatch* hypothesis proposed in Section 3, we visualize the loss landscape of the PMP-trained model. Unlike standard pre-training, which typically seeks a minimum that is flat in all directions, PMP theoretically creates a highly *anisotropic* landscape—flat within the authorized subspace but sharp along the frozen directions.

We verify this by plotting the 1D loss interpolation $f(\alpha) = \mathcal{L}(\theta^* + \alpha \cdot \delta)$ along two distinct random directions: (1) **Authorized Direction (δ_M):** A random vector normalized and confined strictly within the private mask subspace (i.e., $\delta_M \odot (1 - M) = 0$). (2) **Unauthorized Direction (δ_{Full}):** A random vector normalized in the full parameter space, representing the trajectory of standard, unauthorized fine-tuning. As illustrated in Figure 5, the geometric distinction is pronounced. The **Authorized (Blue)** trajectory remains within a low-loss basin, exhibiting a smooth and flat curvature that preserves the model's pre-trained capabilities. In sharp contrast, the **Unauthorized (Red)** trajectory immediately encounters **steep optimization barriers**. Even with minute perturbations ($\alpha \approx \pm 0.1$), the loss along the full space direction spikes significantly. This visualization provides a physical intuition for how PMP works: without applying the private mask, gradient updates inevitably possess components along the frozen, high-curvature directions (the "steep walls" of the valley). This forces the model parameters to ascend these sharp barriers, leading to the rapid destabilization and performance degradation observed in our main experiments. The mismatched objectives between pre-training and fine-tuning form a pitfall for unauthorized fine-tuning.

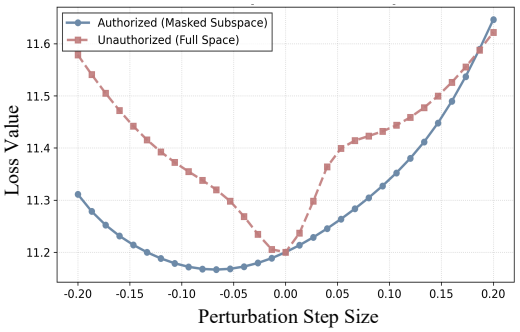


Figure 5: **1D Loss Landscape Visualization.** We compare the loss curve along an authorized direction (confined to the mask, **Blue**) versus an unauthorized direction (full parameter space, **Red**).

5.2 Robustness Against Potential Mask Detection

We consider a gradient-observing adversary who does not know the mask realization, the mask ratio, or the gradient magnitude distributions induced by masking. In this setting, gradient magnitudes alone do not provide an identifiable signal for distinguishing masked and unmasked parameters from a single or limited number of observations, as absolute gradient scales lack semantic meaning without distributional priors. Figure 6 shows the gradient magnitude distributions of masked and unmasked parameters for illustration. Although a distributional shift can be observed *a posteriori* when mask labels are known, the substantial overlap between the two distributions prevents reliable parameter-wise mask detection. Consequently, without prior knowledge of the mask distribution, an adversary cannot effectively recover the private mask.

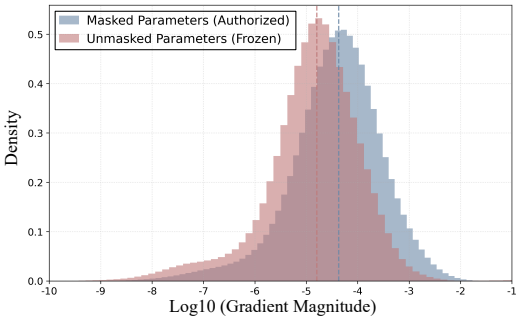


Figure 6: **Gradient magnitude distributions of masked (authorized) and unmasked (frozen) parameters on a held-out input.** For visualization clarity, gradients close to zero are omitted; these near-zero gradients predominantly correspond to unmasked (frozen) parameters.

6 Related Work

Restriction on Foundation Model Fine-Tuning. Existing efforts on restricting foundation model fine-tuning primarily focus on preventing unauthorized adaptation to specific harmful datasets or domains. Wang et al. Wang et al. (2024) propose *Moderator*, which performs an active defense by first specifying a target domain and then applying task-vector-based unlearning to suppress adaptation within that domain. Deng et al. Deng et al. (2024) introduce *Sophon*, which mimics the fine-tuning dynamics of a target domain and explicitly constrains fine-tuning gradients during adaptation. A related line of work, often referred to as model immunization Rosati et al. (2024); Zheng & Yeh (2024); Li et al. (2025); Sarker et al., injects task-specific disturbing signals into pretrained models to impair fine-tunability for particular concepts or domains. Existing approaches rely on prior knowledge of downstream tasks or access to fine-tuning data/objectives, which is unrealistic in practice.

Lottery Ticket and Masked Training. The lottery ticket hypothesis shows that over-parameterized neural networks often contain sparse subnetworks that can achieve performance comparable to the full model Frankle & Carbin (2018). Subsequent work demonstrates that exploiting such lottery tickets can significantly improve training efficiency and stability Frankle et al. (2019); Morcos et al. (2019), with particular benefits in distributed learning systems Li et al. (2020) and large foundation models Panda et al. (2024); Yuan et al. (2025). Most of these approaches identify lottery tickets after full or partial training. More recent work explores discovering lottery tickets at early stages of training to further reduce memory and computational overhead Chen et al. (2021).

7 Conclusion

This paper studied the problem of building non-fine-tunable FMs, motivated by the need to reconcile open model release with control over downstream adaptation. To this end, we proposed PMP, a pre-training paradigm that concentrates learning into a private sparse subnetwork and releases only the final dense parameters. By withholding the mask, PMP prevents unauthorized fine-tuning from aligning with the optimization geometry induced during pre-training. Our theoretical analysis shows that fine-tuning without access to the private mask optimizes a mismatched objective, injecting destabilizing updates along unadapted parameter directions. Experiments on LLMs confirm this analysis: PMP preserves strong base performance while consistently degrading unauthorized fine-tuning across architectures, tasks, and hyperparameter choices. Overall, PMP reframes non-fine-tunability as a pre-training-level property rather than a task-specific unlearning. We hope this work encourages further research into controllable FM deployment, where openness and safety can coexist without requiring assumptions about downstream data or training procedures.

Reproducibility Statement

We have made every effort to ensure the reproducibility of our results. The full PMP procedure, including early-bird mask discovery, private mask pre-training and release without the mask, is specified in Sec. 2. The model architectures, pre-training data and pipeline, mask discovery settings, downstream fine-tuning protocol and evaluation metrics are described in Sec. 4, and the complete training hyperparameters, hardware and baseline configurations are provided in Appendix A. All datasets used in this work are publicly available, and all baselines are run with their official implementations under the same settings as PMP.

AI Use Statement

Large language models appear in this work in two distinct roles. First, LLMs are the *subject* of our study: the decoder-only Transformers pre-trained with and without PMP and evaluated in experiments are research objects, not tools used to produce the paper. Second, we used LLM-based coding agents and assistants as auxiliary tools during research and writing, as described below.

LLM agents were used to help write and debug experiment code (e.g., training, fine-tuning and evaluation scripts) and to help analyse and summarise experimental results. All code was reviewed and run by the authors, and every reported number was produced by experiments conducted by the authors. LLMs were used to polish the language of text written by the authors, improving clarity and readability.

LLMs were not used for any other purpose. All research ideas, the design of PMP, the theoretical analysis, the experimental design and the conclusions were developed by the authors, who reviewed all LLM-assisted content and take full responsibility for the paper.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. Earlybert: Efficient bert training via early-bird lottery tickets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2195–2207, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Jiangyi Deng, Shengyuan Pang, Yanjiao Chen, Liangming Xia, Yijie Bai, Haiqin Weng, and Wenyuan Xu. Sophon: Non-fine-tunable learning to restrain task transferability for pre-trained models. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 2553–2571, 2024.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020.

Boheng Li, Renjie Gu, Junjie Wang, Leyi Qi, Yiming Li, Run Wang, Zhan Qin, and Tianwei Zhang. Towards resilient safety-driven unlearning for diffusion models against downstream fine-tuning. *arXiv preprint arXiv:2507.16302*, 2025.

Xian Li, Yuanning Han, Di Liu, Pengcheng An, and Shuo Niu. When generative ai is intimate, sexy, and violent: Examining not-safe-for-work (nsfw) chatbots on flowgpt. *arXiv preprint arXiv:2601.14324*, 2026.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *Advances in neural information processing systems*, 32, 2019.

Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. Lottery ticket adaptation: Mitigating destructive interference in llms. *arXiv preprint arXiv:2406.16797*, 2024.

Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Safe-clip: Removing nsfw concepts from vision-and-language models. In *European Conference on Computer Vision*, pp. 340–356. Springer, 2024.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

Najibul Haque Sarker, Zaber Ibn Abdul Hakim, Alvi Md Ishmam, Chia-Wei Tang, and Chris Thomas. Model immunization by trapping harmful finetuning. In *Lock-LLM Workshop: Prevent Unauthorized Knowledge Use from Large Language Models*.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Peiran Wang, Qiyu Li, Longxuan Yu, Ziyao Wang, Ang Li, and Haojian Jin. Moderator: Moderating text-to-image diffusion models through fine-grained context-based policies. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1181–1195, 2024.

Model	Stage	Variant	COLA	SST2	MRPC	QQP	STS-B	MNLI-m	MNLI-mm	QNLI	RTE	AVG
PMP	Base	With PMP	63.66	50.80	40.20	38.66	16.53	32.81	32.60	50.72	53.79	42.20
SOPHON	Base	With Sophon	68.94	51.15	68.87	37.31	23.00	32.25	32.58	50.52	47.29	45.77
REPNOISE	Base	With RepNoise	30.87	49.08	31.62	63.18	49.53	31.82	31.82	50.53	47.29	42.86
PMP	Fine-Tuned	With PMP	61.64	76.03	69.36	78.20	22.43	59.52	58.48	63.75	46.57	59.55
SOPHON	Fine-Tuned	With Sophon	69.13	77.41	71.08	78.78	26.60	51.87	52.46	64.80	53.79	60.66
REPNOISE	Fine-Tuned	With RepNoise	66.73	49.27	39.95	63.13	50.07	32.78	32.93	50.35	48.38	48.18

Table 2: Comparison with SOPHON and REPNOISE under identical evaluation settings.

Model	Stage	Variant	COLA	SST2	MRPC	QQP	STS-B	MNLI-m	MNLI-mm	QNLI	RTE	AVG
–	Base	W/O Defense	37.01	50.69	31.62	36.84	16.13	36.77	33.35	50.58	52.71	38.41
PMP	Base	With PMP	63.66	50.80	40.20	38.66	16.53	32.81	32.60	50.72	53.79	42.20
SOPHON	Base	With Sophon	68.94	51.15	68.87	37.31	23.00	32.25	32.58	50.52	47.29	45.77
REPNOISE	Base	With RepNoise	30.87	49.08	31.62	63.18	49.53	31.82	31.82	50.53	47.29	42.86
–	Fine-Tuned	W/O Defense	63.36	83.14	71.32	86.51	32.47	71.17	71.34	79.37	54.15	68.65
PMP	Fine-Tuned	With PMP	61.64	76.03	69.36	78.20	22.43	59.52	58.48	63.75	46.57	59.55
SOPHON	Fine-Tuned	With Sophon	69.13	77.41	71.08	78.78	26.60	51.87	52.46	64.80	53.79	60.66
REPNOISE	Fine-Tuned	With RepNoise	66.73	49.27	39.95	63.13	50.07	32.78	32.93	50.35	48.38	48.18

Table 3: Full comparison with the no-defense baseline.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Fei Yuan, Chang Ma, Shuai Yuan, Qiushi Sun, and Lei Li. Ks-lottery: Finding certified lottery tickets for multilingual transfer in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9077–9090, 2025.

Amber Yijia Zheng and Raymond A Yeh. Imma: Immunizing text-to-image models against malicious adaptation. In *European Conference on Computer Vision*, pp. 458–475. Springer, 2024.

A Additional Experimental Set-Up and Results

A.1 Hardware and Hyperparameters

All pre-training runs are executed on a single NVIDIA H20 GPU. We optimize using AdamW with a cosine learning-rate schedule, using a base learning rate of $\eta = 2 \times 10^{-5}$, 2,000 learning-rate warmup optimizer updates, and a total of 20,000 optimizer updates. We use a micro-batch size of 4 sequences with gradient accumulation of 8, resulting in an effective batch size of 32 sequences per optimizer update. We clip the global gradient norm to 1.0 and fix the random seed to 42.

A.2 Comparison with Existing Defenses

Trade-off between capability and adaptation. Table 2 compares PMP with SOPHON Deng et al. (2024) and REPNOISE Rosati et al. (2024). PMP achieves a strong reduction in fine-tuning effectiveness while maintaining competitive base performance. Compared with SOPHON, PMP obtains lower fine-tuned average performance (59.55 vs. 60.66), indicating stronger suppression of unauthorized adaptation. Compared with REPNOISE, PMP preserves substantially better fine-tuned utility while still reducing adaptation gains, suggesting a more balanced trade-off between base capability and controllability.

A key distinction is that PMP operates in a task-agnostic and black-box setting, without requiring access to downstream data or task objectives. In contrast, task-specific defenses typically rely on stronger assumptions about the target domain. Despite this weaker threat model, PMP achieves comparable or stronger degradation of fine-tuning gains, supporting the claim that non-fine-tunability can be induced as a pre-training-level property rather than through task-specific suppression.

Method	COLA	SST2	MRPC	QQP	STS-B	MNLI-m	MNLI-mm	QNLI	RTE	AVG
Base performance before fine-tuning (higher is better)										
W/O Defense	37.01	50.69	31.62	36.84	16.13	36.77	33.35	50.58	52.71	38.41
PMP	63.66	50.80	40.20	38.66	16.53	32.81	32.60	50.72	53.79	42.20
SOPHON	68.94	51.15	68.87	37.31	23.00	32.25	32.58	50.52	47.29	45.77
REPNOISE	30.87	49.08	31.62	63.18	49.53	31.82	31.82	50.53	47.29	42.86
Vaccine	30.90	49.08	31.62	67.30	21.60	34.00	35.10	47.30	52.70	41.07
Performance after full fine-tuning (lower is better for non-fine-tunability)										
W/O Defense	63.36	83.14	71.32	86.51	32.47	71.17	71.34	79.37	54.15	68.65
PMP	61.64	76.03	69.36	78.20	22.43	59.52	58.48	63.75	46.57	59.55
SOPHON	69.13	77.41	71.08	78.78	26.60	51.87	52.46	64.80	53.79	60.66
REPNOISE	66.73	49.27	39.95	63.13	50.07	32.78	32.93	50.35	48.38	48.18
Vaccine	69.10	52.64	67.65	67.30	28.10	33.20	33.30	55.40	46.57	50.36

Table 4: Extended comparison across PMP, SOPHON, REPNOISE, and Vaccine. Base performance evaluates released-model utility, while full fine-tuning performance reflects non-fine-tunability (lower is better).

Setting	PMP	SOPHON	REPNOISE	Vaccine
Base	0.05	0.05	0.05	0.05
Base + FT	0.47	0.47	0.47	0.47
Defended	0.05	0.05	0.05	0.05
Defended + FT	**0.08**	0.18	0.08	0.28

Table 5: Harmfulness on BeaverTails under harmful fine-tuning. Lower is better.

Magnitude of fine-tuning gains. Table 3 includes the no-defense baseline. Without any defense, full fine-tuning substantially improves the average GLUE score from 38.41 to 68.65, confirming that the released model is highly adaptable under standard fine-tuning. PMP reduces the fine-tuned average to 59.55, showing that it meaningfully limits downstream adaptation while preserving reasonable base performance. This is consistent with our objective: PMP is not intended to completely eliminate all fine-tuning gains, but to make unauthorized adaptation systematically less effective.

Comparison with additional baselines. Table 4 further includes Vaccine. Overall, Vaccine shows behavior similar to REPNOISE: it can reduce fine-tuned performance, but its base performance is less stable across tasks. PMP achieves a more consistent balance: it maintains reasonable base model utility and still reduces the average fine-tuned score from 68.65 to 59.55. These results suggest that PMP provides a practical trade-off between preserving released-model capability and limiting unauthorized adaptation.

Safety-oriented evaluation. Table 5 evaluates robustness against harmful fine-tuning on BeaverTails. Standard fine-tuning substantially increases harmfulness from 0.05 to 0.47. In contrast, PMP maintains low harmfulness after fine-tuning, increasing only to 0.08. PMP outperforms SOPHON and Vaccine and matches REPNOISE under this safety-oriented metric. This result directly supports the safety motivation of non-fine-tunable models: unauthorized fine-tuning should not easily convert a safe released model into a harmful adapted model.

Summary. Overall, these additional experiments strengthen the empirical support for PMP. Compared with no defense, PMP substantially reduces fine-tuning gains. Compared with existing defenses, PMP achieves a competitive or stronger trade-off while operating under weaker assumptions. Moreover, the BeaverTails results show that PMP also improves robustness against harmful fine-tuning, further supporting its role as a general pre-training-level mechanism for controllable model release.