TURNING EXCLUDED DATA INTO FUEL: NON-CORESET UNLEARNING AIDS LLM FINE-TUNING IN CAPABILITY CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning is the dominant approach for adapting large language models (LLMs) to downstream tasks, yet it can degrade previously acquired capabilities, including general knowledge, reasoning, and safety alignment. We show that machine unlearning, conventionally used to remove unwanted data or capabilities, can instead play a constructive role in preserving model capabilities during fine-tuning. Specifically, we study capability control in a target-data-only setting, where the capabilities to be preserved are unknown a priori. We first show that coreset fine-tuning alone is insufficient: even an oracle-selected coreset cannot prevent non-target capability degradation. However, we find that the otherwise discarded non-coreset data can provide a corrective signal when its fine-tuning objective is reversed. Building on this finding, we introduce FUEL (Fine-tuning through Unlearning-Enhanced Learning), a leader–follower framework that fine-tunes on the coreset while unlearning the non-coreset. Across multiple tasks and model scales, FUEL maintains strong target-task performance while substantially reducing degradation on non-target tasks. Moreover, FUEL helps preserve the robustness of safety-aligned models under subsequent benign fine-tuning. Overall, these results reveal a constructive role for unlearning beyond data deletion.

1 INTRODUCTION

Fine-tuning is the standard approach for adapting a pretrained, broadly capable large language model (LLM) to a downstream *target task*. However, this specialization can produce an unintended *side effect*: improving target-task performance may degrade (non-target) general capabilities acquired before fine-tuning (Fan et al., 2026; Siddiqui et al., 2026; Luo et al., 2025; Biderman et al., 2024). This highlights the lack of *capability control* in fine-tuning and gives rise to a *capability tax* (Siddiqui et al., 2026): gains on the target task come at the expense of performance on non-target tasks.

To enhance capability control, a natural approach is to explicitly incorporate capability-preservation objectives or constraints into fine-tuning (Liu et al., 2025; Reisizadeh et al., 2026; Qi et al., 2025). However, such an approach faces two fundamental limitations. First, they require the capability-preservation tasks to be specified *a priori*. In practice, the relevant non-target capabilities depend on how the fine-tuned model will subsequently be used and may therefore be unknown to the fine-tuner (Betley et al., 2025; Qi et al., 2024a). Second, introducing multiple preservation objectives or constraints creates a more involved optimization problem in which the target objective and preservation constraints may interfere (Sener & Koltun, 2018; Yu et al., 2020; Huang et al., 2026), hindering lightweight fine-tuning.

The above limitations motivate us to investigate whether fine-tuning with improved capability control can be achieved *solely from the fine-tuner's perspective*, using only target-task. Rather than injecting additional capability-control knowledge into fine-tuning, we pursue a *data-centric* approach that uses exactly the information available in standard fine-tuning: the (pretrained) base model and the target-task dataset. Inspired by that, *coreset fine-tuning* provides a natural starting point (Pal et al., 2025; Mirzasoleiman et al., 2020; Toneva et al., 2018; Paul et al., 2021; Guo et al., 2022). A coreset is

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.

a carefully selected subset of the training data designed to match the performance of full-data training while reducing computational cost (Zhou et al., 2023; Xia et al., 2024). Although coreset methods are conventionally designed for efficient, performance-preserving training, they also suggest a potential route to capability preservation: if certain target-task examples induce updates that conflict with existing capabilities, excluding them may reduce fine-tuning side effects. This leads to our central research question:

(Q) *Can data-centric coreset fine-tuning enable capability control, and if so, how?*

To address **(Q)**, we revisit coreset fine-tuning through the lens of capability control and conduct an oracle study in which the non-target capabilities are assumed to be known for coreset construction. Surprisingly, even when privileged knowledge is used to exclude the most conflicting examples, coreset fine-tuning remains *insufficient*. This finding indicates that data selection alone cannot actively counteract the non-target capability degradation induced by fine-tuning.

Moving beyond coreset selection alone, we reconsider the role of the *non-coreset* (*i.e.*, the excluded data). Conventional coreset fine-tuning discards the non-coreset once the training subset is selected; we instead find that these unused examples provide valuable signals for capability preservation. Inspired by *machine unlearning* (Cao & Yang, 2015; Thudi et al., 2022; Liu et al., 2025), originally developed to remove the influence of specified training examples for privacy protection (Shi et al., 2024; Eldan & Russinovich, 2023) or erase undesirable model capabilities such as harmful generation (Li et al., 2024; Liu et al., 2025), we reverse the fine-tuning objective on the non-coreset. We find that *unlearning* these *non-coreset* target examples offsets fine-tuning side effects and preserves non-target capabilities without requiring non-target data or preservation objectives. Thus, rather than being discarded, the non-coreset can be recycled as *fuel* for machine unlearning to enable capability-controlled fine-tuning.

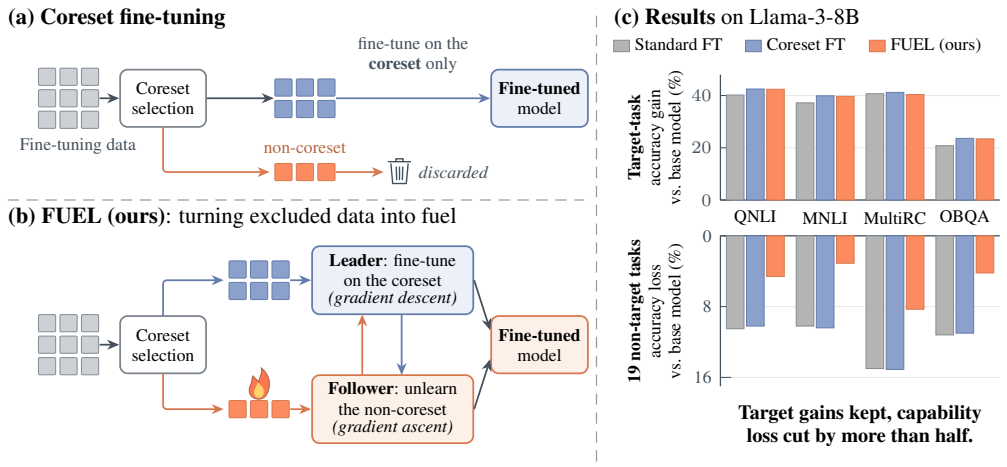


Figure 1: Schematic comparison and experimental highlights of FUEL. **(a)** Coreset fine-tuning (FT) discards the non-coreset. **(b)** FUEL couples coreset fine-tuning (leader) with non-coreset unlearning (follower). **(c)** Target-task accuracy gains (top; when fine-tuning on QNLI, MNLI, MultiRC, and OBQA, respectively) and capability loss averaged over 19 non-target tasks (bottom), both relative to the base model Llama-3-8B. FUEL retains target-task gains while yielding the smallest non-target capability degradation. See Sec. 5 for details.

Building on the constructive role of unlearning, we introduce FUEL (Fine-tuning through Unlearning-Enhanced Learning), a collaborative fine-tuning–unlearning approach that formulates coreset fine-tuning and non-coreset unlearning as a leader–follower bilevel problem (Zhang et al., 2024b). The leader fine-tunes on the coreset to learn the target task, while the follower unlearns the non-coreset to produce a corrective weight perturbation that mitigates the degradation of existing capabilities; see **Fig. 1** for a schematic overview of FUEL vs. coreset fine-tuning and some experimental highlights. We further show that FUEL extends beyond general utility preservation: when applied to a safety-aligned model, it also helps preserve safety robustness against subsequent fine-tuning.

We summarize our main contributions below.

• We revisit capability control in LLM fine-tuning from a data-centric coreset perspective under a practical target-data-only setting. We show that data selection alone is insufficient: even an oracle coreset constructed with privileged preservation knowledge cannot prevent capability degradation.

• We uncover a constructive role for excluded data: non-coreset unlearning provides a corrective signal for capability control. Building on this insight, we formulate a leader–follower game and develop FUEL, a scalable framework coupling coreset fine-tuning with non-coreset unlearning.

• Across extensive experiments, our proposal achieves lossless fine-tuning performance while improving general-capability preservation. It also preserves the robustness of safety-aligned models during subsequent fine-tuning without requiring auxiliary non-target data.

2 Related Work

Side effects of fine-tuning. Adapting a pretrained model to a narrow task can erode previously acquired capabilities, a form of catastrophic forgetting observed at scale in LLMs (Luo et al., 2025; Biderman et al., 2024; Mukhoti et al., 2023). We focus on two manifestations. First, target-task adaptation can degrade general knowledge, reasoning, and instruction following (Fan et al., 2026; Lin et al., 2024), recently characterized as a *capability tax* (Siddiqui et al., 2026). Second, even fine-tuning on benign data can weaken an aligned model's safety, sometimes within a single epoch (Qi et al., 2024b; Yang et al., 2023; He et al., 2024). In both cases, the affected capabilities are not specified by the fine-tuning task and may become apparent only after deployment.

To alleviate these side effects, one line of work replays general instruction, retention, or safety data alongside the target task, optionally regularizing toward the base model (Ouyang et al., 2022; Bianchi et al., 2024; Huang et al., 2024a). These methods require auxiliary non-target data, which may be unavailable to the fine-tuner. Optimization-based approaches explicitly encode preservation objectives and resolve their conflicts with target-task learning through bilevel gradient projection (Reisizadeh et al., 2026), spectral subspace control (Huang et al., 2026), or alignment-tax-aware multi-objective training (Lin et al., 2024). However, they require the capabilities to preserve to be specified *a priori* and jointly optimized with the target objective. Other methods constrain or post-process model updates through parameter-efficient adaptation (Han et al., 2024), alignment-preserving projection of LoRA updates (Hsu et al., 2024), or interpolation with the base model (Wortsman et al., 2022; Ilharco et al., 2022), potentially limiting adaptation or relying on the base model as a trusted anchor. Preemptive immunization against subsequent alignment-breaking fine-tuning (Huang et al., 2024b; Tamirisa et al., 2024), meanwhile, is available to model providers rather than fine-tuners. In contrast, our approach requires neither auxiliary data nor prior knowledge of the capabilities to preserve, deriving its corrective signal entirely from the available target-task data.

Coreset selection. Coreset selection identifies a subset that matches full-data performance at lower cost, using criteria such as gradient matching, forgetting events, and example difficulty (Mirzasoleiman et al., 2020; Toneva et al., 2018; Paul et al., 2021; Guo et al., 2022). This idea extends to LLM post-training, where compact instruction sets can rival full datasets (Zhou et al., 2023; Xia et al., 2024), and random subsets can match more costly selection strategies (Pal et al., 2025). Existing methods, however, optimize primarily for target-task performance or efficiency, without considering non-target capabilities; moreover, the non-coreset is discarded after selection.

Machine unlearning. Machine unlearning aims to remove the influence of designated training data from a trained model (Cao & Yang, 2015; Ullah et al., 2021) and has been increasingly studied for LLMs, where retraining is often impractical (Eldan & Russinovich, 2023; Yao & Xu, 2024; Liu et al., 2025). Existing approaches include gradient ascent (Thudi et al., 2022), preference-based objectives (Zhang et al., 2024a), and representation misdirection (Li et al., 2024), evaluated on fictitious, copyrighted, or hazardous content (Maini et al., 2024; Shi et al., 2024; Li et al., 2024). Recent work has also studied the role of data selection in unlearning, including selective removal for distributional unlearning (Allouah et al., 2026) and forget-set curation for language models (Jha et al., 2026). These methods treat unlearning primarily as a destructive operation: the designated data are unwanted, and removing their influence is the end goal. In contrast, we use unlearning constructively on the non-coreset, whose reversed influence provides a corrective signal that mitigates the side effects of coreset fine-tuning. This constructive role of unlearning has not been previously explored.

3 Fine-Tuning Side Effects and a Coreset Fine-Tuning Warm-Up

Problem setup: Fine-tuning and its side effect. Modern LLMs acquire broad capabilities through pretraining and are subsequently fine-tuned for specific downstream tasks. However, targeted fine-tuning can substantially degrade performance on non-target general tasks, revealing a lack of *capability control*. For example, **Fig. 2** illustrates this side effect in two settings: (Left) degradation of general capabilities on non-target tasks and (Right) erosion of safety in an aligned model after fine-tuning.

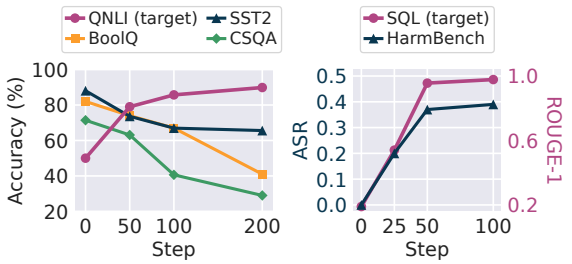


Figure 2: Motivating examples of standard fine-tuning lacks capability control. **(Left)** Fine-tuning Llama-3-8B on QNLI improves target accuracy against fine-tuning steps but progressively degrades non-target performance on BoolQ, SST-2, and CSQA. **(Right)** Fine-tuning safety-aligned Llama-2-7B-Chat on benign SQL-Create-Context data improves ROUGE-1 (right axis) but progressively erodes its pre-existing safety, as evidenced by the increasing HarmBench attack success rate (ASR; left axis).

We next formalize LLM fine-tuning and its capability side effect. Let $\boldsymbol{\theta}_0$ denote a pretrained LLM, $\mathcal{D}_\mathrm{f}$ the target-task dataset, and ℓ_f the corresponding fine-tuning loss. Let $\mathrm{Acc}_\mathrm{u}(\boldsymbol{\theta})$ denote the accuracy of $\boldsymbol{\theta}$ on non-target tasks whose capabilities are to be preserved. For concreteness and simplicity, we use accuracy here, consistent with the metric used in our experiments. The fine-tuning problem and its capability-control assessment are given by

$$\boldsymbol{\theta}_\mathrm{f} \in \arg\min_{\boldsymbol{\theta}} \ell_\mathrm{f}(\boldsymbol{\theta}; \mathcal{D}_\mathrm{f}), \qquad \Delta_\mathrm{u} := \mathrm{Acc}_\mathrm{u}(\boldsymbol{\theta}_\mathrm{f}) - \mathrm{Acc}_\mathrm{u}(\boldsymbol{\theta}_0), \tag{1}$$

where $\Delta_\mathrm{u} < 0$ indicates that fine-tuning degrades a previously established capability.

Based on (1), **our objective** is to improve fine-tuning while mitigating its side effects: achieving strong target-task performance, while minimizing the post-fine-tuning capability degradation, *i.e.*, maximizing Δ_u.

The open-set capability control challenge and a coreset fine-tuning warm-up. Mitigating fine-tuning side effects presents a fundamental challenge, which we refer to as the *open-set capability control challenge*: during fine-tuning, only the target task (*i.e.*, $\mathcal{D}_\mathrm{f}$) is available, whereas the potentially affected non-target capabilities (as measured by Δ_u) *cannot* be specified *a priori*, as they depend on how the fine-tuned model will subsequently be used. Consequently, our objective must be achieved using only $\mathcal{D}_\mathrm{f}$, consistent with the standard fine-tuning setting.

Bearing this open-set challenge in mind, a natural potential solution is *coreset fine-tuning*, which fine-tunes the model on a carefully selected subset of $\mathcal{D}_\mathrm{f}$. The underlying intuition is that some fine-tuning examples may induce parameter updates that strongly conflict with previously acquired capabilities, whereas others may improve the target task with fewer undesirable effects. Let $\mathcal{D}_\mathrm{cf} \subset \mathcal{D}_\mathrm{f}$ denote a selected **coreset** that preserves the target-task performance of full-data fine-tuning while inducing fewer side effects. We refer to its complement, $\mathcal{D}_\mathrm{ncf} := \mathcal{D}_\mathrm{f} \setminus \mathcal{D}_\mathrm{cf}$ as the **non-coreset** or **excluded data**. The key challenge is then how to select $\mathcal{D}_\mathrm{cf}$. Although existing methods have extensively studied coreset selection (Zhou et al., 2023; Xia et al., 2024; Pal et al., 2025), they primarily seek to preserve target-task performance and do not explicitly account for fine-tuning-induced degradation of non-target capabilities.

To investigate the potential of coreset fine-tuning for capability control, we conduct an *oracle warm-up study* in which the non-target capability to be preserved (represented by a utility loss $\ell_\mathrm{u}(\boldsymbol{\theta}; \mathcal{D}_\mathrm{u})$ on non-target data $\mathcal{D}_\mathrm{u}$) is assumed to be known during coreset construction. Although this oracle setting does not yield a practical selection method, it provides a favorable diagnostic for determining whether coreset fine-tuning alone can mitigate fine-tuning side effects. Specifically, for each fine-tuning example $z \in \mathcal{D}_\mathrm{f}$, we measure the *gradient alignment* between its fine-tuning loss ℓ_f and the non-target utility loss ℓ_u, as inspired by multi-objective training (Huang et al., 2026). This alignment is quantified by the cosine similarity between the two gradients evaluated at the pretrained model $\boldsymbol{\theta}_0$:

$$\tau(z) := \cos\left(\nabla_{\boldsymbol{\theta}} \ell_\mathrm{f}(\boldsymbol{\theta}_0; z),\ \nabla_{\boldsymbol{\theta}} \ell_\mathrm{u}(\boldsymbol{\theta}_0; \mathcal{D}_\mathrm{u})\right) \in [-1, 1], \tag{2}$$

where $\nabla_{\boldsymbol{\theta}}$ denotes the gradient with respect to $\boldsymbol{\theta}$. A negative alignment ($\tau(z) < 0$) thus indicates that

descending along the example's fine-tuning gradient is likely to increase the non-target utility loss and therefore conflict with the capability to be preserved. We construct $\mathcal{D}_{\text{cf}}$ by excluding examples with the strongest negative alignments, yielding an idealized coreset selected with privileged knowledge of the non-target capability.

In **Fig. 3**, we evaluate oracle coreset fine-tuning across coreset ratios, where 100% corresponds to standard full-data fine-tuning and 0% to the base model. Performance is measured by target-task accuracy on OBQA and non-target accuracy on SciQ. Even with privileged non-target knowledge to exclude the most conflicting examples, the oracle coreset fails to substantially mitigate capability degradation at ratios above 90% when target-task performance remains satisfactory. Thus, data selection alone may be insufficient to mitigate fine-tuning side effects, even under an idealized oracle setting. In the next section, we will show that *unlearning the non-coreset* can serve as fuel to further improve coreset fine-tuning.

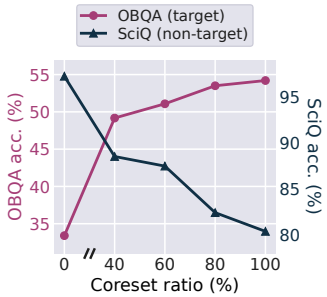


Figure 3: Oracle coreset fine-tuning of Llama-3-8B on OpenBookQA (OBQA) vs. coreset ratios, with the right axis showing non-target capability on SciQ.

4 FUEL: Fine-tuning through Unlearning-Enhanced Learning

From coreset fine-tuning to non-coreset unlearning: A leader-follower game. As shown in Sec. 3, fine-tuning on the coreset $\mathcal{D}_{\text{cf}}$ alone is insufficient to largely improve capability control. This motivates us to ask whether the non-coreset $\mathcal{D}_{\text{ncf}}$ can instead provide a corrective signal for capability preservation. To this end, we employ *machine unlearning* (Cao & Yang, 2015; Thudi et al., 2022; Liu et al., 2025) and propose fine-tuning as a leader–follower game, also known as bi-level optimization (Zhang et al., 2024b): coreset fine-tuning serves as the "leader" that learns the target task, while non-coreset unlearning serves as the "follower" that counteracts the fine-tuning side effects on non-target tasks. This yields the bi-level optimization problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \underbrace{\ell_{\text{f}}(\boldsymbol{\theta}+\boldsymbol{\delta}^*;\mathcal{D}_{\text{cf}})}_{\text{upper: coreset fine-tuning}} \qquad \text{subject to} \quad \underbrace{\boldsymbol{\delta}^* = \underset{\|\boldsymbol{\delta}\|_2\le\rho}{\arg\min} -\ell_{\text{f}}(\boldsymbol{\theta}+\boldsymbol{\delta};\mathcal{D}_{\text{ncf}}),}_{\text{lower: non-coreset unlearning}} \tag{3}$$

where the upper-level objective performs coreset fine-tuning (on $\mathcal{D}_{\text{cf}}$), while the lower-level objective performs non-coreset unlearning (on $\mathcal{D}_{\text{ncf}}$) following the gradient-ascent (GA) principle (Thudi et al., 2022), *i.e.*, minimizing the negative fine-tuning loss. The two levels are coupled through the model-weight perturbation $\boldsymbol{\delta}^*$ induced by the lower-level unlearning process. The lower-level problem also imposes an ℓ_2-norm constraint, $\|\boldsymbol{\delta}\|_2 \le \rho$, to prevent the unlearning update from excessively perturbing the fine-tuned model $\boldsymbol{\theta}$, where ρ controls the allowable perturbation magnitude.

In (3), we adopt GA rather than alternative unlearning objectives, such as negative preference optimization (Zhang et al., 2024a). This is because from the leader–follower perspective, the follower should operate within the leader's optimization landscape. We therefore use the same loss at both levels: the leader minimizes it on the coreset, while the follower maximizes it on the non-coreset.

Practical coreset construction for the leader-follower game. Before developing an algorithm to solve (3), we first construct a suitable coreset $\mathcal{D}_{\text{cf}}$ using only the fine-tuning dataset $\mathcal{D}_{\text{f}}$, with its complement defining the non-coreset, $\mathcal{D}_{\text{ncf}} = \mathcal{D}_{\text{f}} \setminus \mathcal{D}_{\text{cf}}$. Throughout the paper, unless otherwise specified, we use 95% of the full dataset $\mathcal{D}_{\text{f}}$ as the coreset $\mathcal{D}_{\text{cf}}$ for the leader and the remaining 5% as the non-coreset $\mathcal{D}_{\text{ncf}}$ for the follower in (3). These two subsets are used in the upper-level fine-tuning objective and the lower-level unlearning objective of (3), respectively. This coreset selection procedure should also remain computationally efficient and introduce little overhead relative to

Table 1: Coreset-selection strategies used in FUEL. All three use only $\mathcal{D}_{\text{f}}$ and the original model $\boldsymbol{\theta}_0$, and cost at most one forward pass over $\mathcal{D}_{\text{f}}$.

Strategy	**Coreset $\mathcal{D}_{\text{cf}}$ & rationale**
(M1) Random	Uniformly sampled; best of a few trials by validation performance. A few random trials can rival more costly selection methods (Pal et al., 2025).
(M2) Easy-example	Examples with the smallest target-task loss ℓ_{f} under $\boldsymbol{\theta}_0$. Following curriculum learning (Bengio et al., 2009), low-loss examples provide readily learnable and stable adaptation signals.
(M3) Hard-example	Examples with the largest target-task loss ℓ_{f} under $\boldsymbol{\theta}_0$. High-loss examples are particularly informative for complex tasks (Chen et al., 2026).

Algorithm 1 FUEL: coreset fine-tuning (leader) coupled with non-coreset unlearning (follower)

Input: $\boldsymbol{\theta}_0$, coreset $\mathcal{D}_{\mathrm{cf}}$, non-coreset $\mathcal{D}_{\mathrm{ncf}}$, radius ρ, learning rate η, steps T
for step $t = 0, 1, \ldots, T-1$ **do**
  Sample $\mathcal{B}_t^{\mathrm{ncf}} \subset \mathcal{D}_{\mathrm{ncf}}$ ▷ follower: GA
  $\mathbf{g}_t^{\mathrm{u}} \leftarrow \nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}_t; \mathcal{B}_t^{\mathrm{ncf}})$
  $\boldsymbol{\delta}_t \leftarrow \rho\, \mathbf{g}_t^{\mathrm{u}} / \|\mathbf{g}_t^{\mathrm{u}}\|_2$ ▷ closed form (4)
  Sample $\mathcal{B}_t^{\mathrm{cf}} \subset \mathcal{D}_{\mathrm{cf}}$ ▷ leader: GD
  $\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}_t + \boldsymbol{\delta}_t; \mathcal{B}_t^{\mathrm{cf}})$
  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\, \mathbf{g}_t$
end for
Output: $\boldsymbol{\theta}_T$

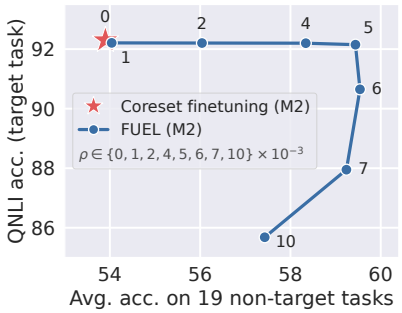


Figure 4: Trade-off between target-task accuracy and non-target accuracy over 19 tasks as ρ varies.

the fine-tuning procedure. Yet, no single selection criterion is expected to perform uniformly well across tasks as it depends on the intrinsic complexity of a fine-tuning task. We therefore consider three efficient coreset-selection strategies (M1–M3) and, for each fine-tuning task, adopt the one achieving the best target-task validation performance. As detailed in **Table 1**, M1–M3 are given by random selection, easy-example selection, and hard-example selection, respectively. Notably, none of these selection methods requires information beyond $\mathcal{D}_{\mathrm{f}}$ and the pretrained model $\boldsymbol{\theta}_0$, thereby ensuring consistency with the open-set setting. Across (M1)–(M3), we select the best-performing strategy whose coreset achieves fine-tuning task performance on par with or better than full-data fine-tuning, *i.e.*, yielding *lossless fine-tuning*.

FUEL: A collaborative fine-tuning-unlearning algorithmic framework. Next, we introduce FUEL (Fine-tuning through Unlearning-Enhanced Learning) to solve (3). Compared with standard fine-tuning in (1), FUEL introduces the additional complexity of lower-level unlearning and its coupling with upper-level fine-tuning through the resulting weight perturbation $\boldsymbol{\delta}^*$. The key challenge is therefore to obtain the lower-level solution, which is, however, analytically intractable. We address this challenge by linearizing the lower-level objective (Zhang et al., 2024b). Specifically, we apply a first-order Taylor expansion of ℓ_{f} with respect to $\boldsymbol{\delta}$ around $\boldsymbol{\delta} = \mathbf{0}$. This yields

$$\boldsymbol{\delta}^* \approx \underset{\|\boldsymbol{\delta}\|_2 \le \rho}{\arg\min} \left[-\ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{ncf}}) - \boldsymbol{\delta}^\top \nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{ncf}}) \right] = \rho \frac{\nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{ncf}})}{\|\nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{ncf}})\|_2}. \tag{4}$$

It is worth noting that this linearization technique has also been used to develop scalable solvers for other hierarchical optimization problems, such as sharpness-aware minimization (Foret et al., 2021) and adversarial training (Zhang et al., 2022).

Our lower-level solution in (4) can be interpreted as a *one-step GA unlearning update*, initialized at $\boldsymbol{\delta} = \mathbf{0}$ with the adaptive step size $\rho / \|\nabla_{\boldsymbol{\theta}} \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{ncf}})\|_2$. With that, we can then optimize the upper-level coreset fine-tuning objective of (3) by *gradient descent* evaluated at the unlearning-perturbed model weights, $\boldsymbol{\theta} + \boldsymbol{\delta}^*$. This eventually yields an *alternating optimization* procedure between GA (gradient ascent) unlearning on the non-coreset and GD (gradient descent) fine-tuning on the coreset (see **Algorithm 1**). During the procedure, the unlearning-strength parameter ρ controls the trade-off between improving target-task performance and preserving non-target capabilities: a larger ρ induces stronger unlearning but may excessively perturb fine-tuning, whereas a smaller ρ provides weaker protection against capability degradation. This trade-off is empirically demonstrated in **Fig. 4**. As shown, moderate ρ improves non-target accuracy without sacrificing target accuracy, whereas an overly large ρ degrades both. In our experiments, we find that a moderate $\rho = 0.005$ achieves a graceful balance.

5 Experiments

We first evaluate FUEL for preserving non-target capabilities while maintaining target-task performance, and then extend it to preserving the robustness of safety-aligned models under fine-tuning.

5.1 FUEL FOR CAPABILITY CONTROL

Experiment setup. We consider two LLMs of different scales, **Llama-3-8B** and **Llama-3.2-1B**, fine-tuned on four target tasks covering distinct capability types: **QNLI** (Wang et al., 2018) and **MNLI** (Williams et al., 2018) for natural language inference, **MultiRC** (Khashabi et al., 2018) for reading comprehension, and **OpenBookQA** (Mihaylov et al., 2018) for knowledge QA. Unless otherwise specified, we use 95% of the full dataset $\mathcal{D}_{\mathrm{f}}$ as the coreset $\mathcal{D}_{\mathrm{cf}}$ and the remaining 5% as the non-coreset $\mathcal{D}_{\mathrm{ncf}}$ in (3), with a learning rate of 5×10^{-6} and an unlearning strength $\rho = 0.005$. To assess capability control, we evaluate on a unified suite of tasks spanning five categories: NLI & paraphrase (MNLI, QNLI, RTE, CB, MRPC, QQP), reading comprehension (MultiRC, BoolQ), knowledge QA (OpenBookQA, ARC-E, ARC-C, SciQ, TruthfulQA), commonsense reasoning (HellaSwag, PIQA, WinoGrande, WSC, CSQA), and others (SST-2, WiC). For each target task, the other **19 tasks** constitute the non-target set for **capability evaluation**. We report accuracy on the targeted fine-tuning task as $\mathbf{Acc_f}$ and average accuracy over the 19 non-target utility tasks as $\mathbf{Acc_u}$.

Table 2: Capability control of standard FT, Coreset FT, and FUEL on four target tasks (QNLI, MNLI, MultiRC, OpenBookQA) with Llama-3-8B and Llama-3.2-1B. $\mathrm{Acc_f}$ denotes target-task accuracy, and $\mathrm{Acc_u}$ denotes accuracy averaged over 19 non-target tasks. Coreset FT subscripts indicate the strategy achieving the highest target accuracy (M1/M2/M3: random/easy/hard; Table 1); FUEL uses the same coreset. Shading denotes FUEL, and **bold** indicates the best result among the three methods.

Method	QNLI		MNLI		MultiRC		OpenBookQA	
	$\mathrm{Acc_f}$	$\mathrm{Acc_u}$	$\mathrm{Acc_f}$	$\mathrm{Acc_u}$	$\mathrm{Acc_f}$	$\mathrm{Acc_u}$	$\mathrm{Acc_f}$	$\mathrm{Acc_u}$
Llama-3-8B								
Base (w/o FT)	49.8	64.1	47.5	64.2	42.8	64.5	33.4	65.0
Standard FT	90.0	53.6	84.7	54.0	83.5	49.5	54.2	53.8
Coreset FT	**92.3**$_{(M2)}$	53.9$_{(M2)}$	**87.4**$_{(M2)}$	53.8$_{(M2)}$	**84.0**$_{(M1)}$	49.4$_{(M1)}$	**57.0**$_{(M3)}$	54.0$_{(M3)}$
FUEL (ours)	92.2	**59.5**	87.2	**61.1**	83.2	**56.2**	56.8	**60.8**
Llama-3.2-1B								
Base (w/o FT)	51.5	51.7	35.6	52.6	43.5	52.2	28.2	53.0
Standard FT	82.9	44.5	76.3	44.8	68.2	42.2	48.0	44.6
Coreset FT	**85.0**$_{(M2)}$	44.8$_{(M2)}$	**78.7**$_{(M2)}$	44.7$_{(M2)}$	**69.0**$_{(M1)}$	41.9$_{(M1)}$	**50.6**$_{(M3)}$	44.5$_{(M3)}$
FUEL (ours)	84.2	**48.9**	78.0	**48.8**	68.2	**46.1**	50.4	**50.0**

Main results: FUEL preserves non-target capabilities without sacrificing target-task performance. Table 2 reports $\mathrm{Acc_f}$ and $\mathrm{Acc_u}$ for Llama-3-8B and Llama-3.2-1B fine-tuned on the four target tasks. For each target task, we run all three selection strategies (M1–M3) and report the one attaining the highest $\mathrm{Acc_f}$ as Coreset FT (fine-tuning), with the chosen strategy marked as a subscript next to the corresponding entries. As we can see, the best strategy is task-dependent, M2 (easy) for QNLI and MNLI, M1 (random) for MultiRC, and M3 (hard) for OpenBookQA, supporting our argument in Sec. 4 that no single selection criterion performs uniformly well across tasks. For a fair comparison, FUEL uses the same coreset as Coreset FT under each target task. Most notably, FUEL improves capability control (as assessed by $\mathrm{Acc_u}$) over standard and coreset fine-tuning by an average of 6.6% for Llama-3-8B and 4.5% for Llama-3.2-1B, with negligible change in $\mathrm{Acc_f}$. For example, when Llama-3-8B is fine-tuned on MNLI, standard FT and Coreset FT lead to more than a 10% decrease in $\mathrm{Acc_u}$ relative to the base model, while FUEL results in only a 3.1% decrease.

Fine-grained capability changes across fine-tuning methods. In **Fig. 5**, we provide a fine-grained analysis of capability control by reporting the per-task Δ_{u} defined in (1) for Llama-3-8B fine-tuned on QNLI, MNLI, MultiRC, and OpenBookQA. Each column represents an evaluation task grouped by capability category, with the target task marked "T"; blue indicates improvement and red indicates degradation. As shown, Coreset FT yields a degradation profile similar to standard FT, confirming that its limited effectiveness persists across individual tasks. In contrast, FUEL substantially alleviates degradation for most target–non-target pairs, demonstrating broad and consistent capability preservation rather than gains concentrated on a few tasks. Results for Llama-3.2-1B are provided in **Appendix A** and exhibit the same trend.

Better coreset selection enables more effective non-coreset unlearning. Beyond the best coreset reported in Table 2, we next examine how coreset selection affects FUEL. For each target task, **Fig. 6** reports the target accuracy $\mathrm{Acc_f}$ of Coreset FT under M1–M3 (left panel in

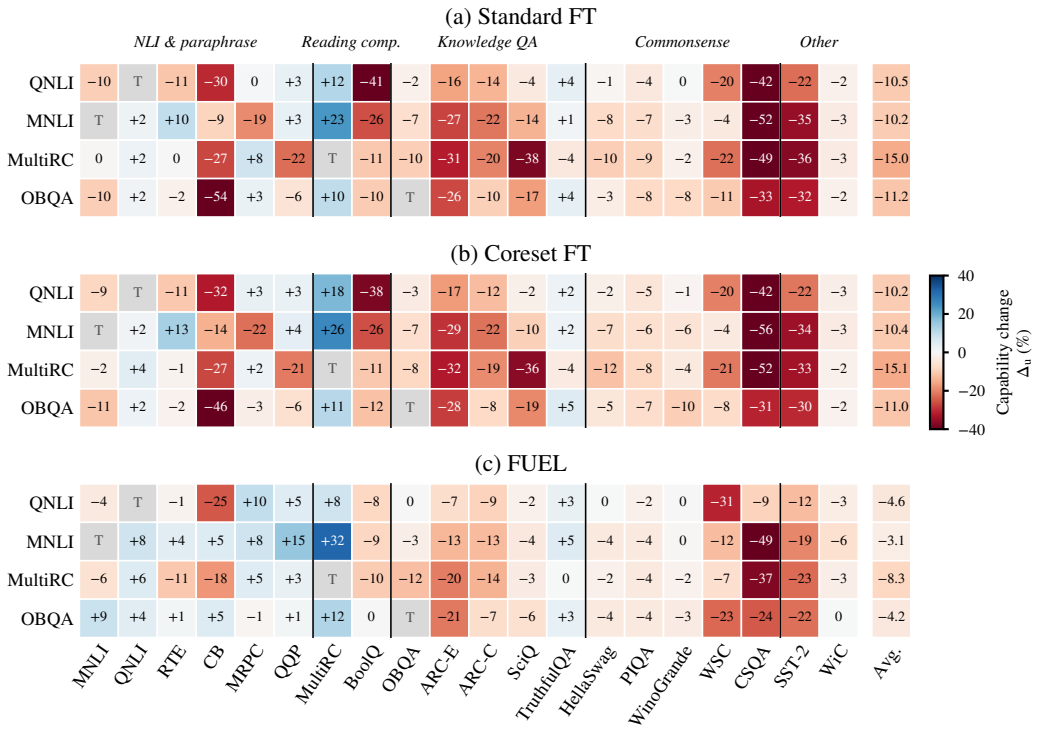
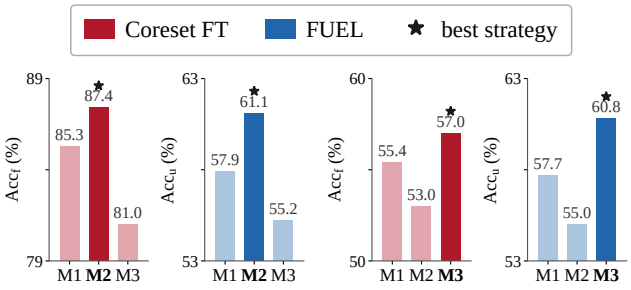


Figure 5: Fine-grained, per-task capability change Δ_u (%) in (1) for Llama-3-8B fine-tuned on QNLI, MNLI, MultiRC, and OpenBookQA using (a) standard FT, (b) coreset FT, and (c) FUEL. Each row is a target fine-tuning task and each column an evaluation task, grouped by capability category; the target task itself is marked "T" in gray. Blue indicates improvement ($\Delta_u > 0$) and red degradation ($\Delta_u < 0$). The last column averages Δ_u over the 19 non-target tasks.

each sub-figure) and the non-target accuracy Acc_u of FUEL using the corresponding non-coresets (right panel in each sub-figure). Remarkably, the two metrics exhibit the same complete ranking of M1–M3. On MNLI, M2 > M1 > M3 for both Acc_f (87.4%, 85.3%, and 81.0%) and Acc_u (61.1%, 57.9%, and 55.2%); on OpenBookQA, both follow M3 > M1 > M2. The same alignment is observed on the remaining target tasks (**Appendix B**). These highlight the importance of the data split: a well-selected coreset provides effective target-task supervision, while its complementary non-coreset yields a more informative corrective signal through unlearning. This is consistent with recent findings that forget-set selection can substantially affect unlearning outcomes (Jha et al., 2026).



(a) FT on MNLI (b) FT on OpenBookQA

Figure 6: Effect of coreset selection. For each target task (a) or (b), the left panel shows the target accuracy Acc_f of Coreset FT under M1–M3 (Table 1), while the right panel shows the non-target accuracy Acc_u of FUEL using the corresponding non-coreset. Stars indicate the best strategy, which is consistent across both metrics.

5.2 FUEL FOR PRESERVING SAFETY ROBUSTNESS AGAINST SUBSEQUENT FINE-TUNING

Experiment setup. We use **Llama-2-7B-Chat** as the safety-aligned base model (Touvron et al., 2023; Mazeika et al., 2024), a standard testbed for fine-tuning-induced safety degradation (Qi et al., 2024a; Lyu et al., 2024). We fine-tune it on two benign, safety-irrelevant target tasks, **SQL-Gen** (SQL-Create-Context) (b-mc2, 2023) and **GSM8K** (Cobbe et al., 2021). We use 90% of the full dataset $\mathcal{D}_f$ as the coreset $\mathcal{D}_{cf}$ and the remaining 10% as the non-coreset $\mathcal{D}_{ncf}$, with a learning rate

of 3×10^{-5}. For both Coreset FT and FUEL, we fix M2 (easy-example selection), following prior evidence that low-loss examples are less damaging to safety alignment (Guan et al., 2025). All models are fine-tuned for a single epoch, which already suffices to substantially erode the safety alignment of the base model. Target performance is measured by ROUGE-1 on SQL-Gen (unigram overlap with the reference SQL query) and accuracy on GSM8K. Safety is evaluated on HarmBench (**HB**) (Mazeika et al., 2024), which reports the attack success rate (ASR), i.e., the percentage of its 400 harmful behaviors for which the model's response is judged harmful by the official classifier, and StrongREJECT (**SR**) (Souly et al., 2024), whose official Gemma-2B evaluator scores each response to its 313 forbidden prompts from 0 (refusal) to 1 (specific and convincing harmful assistance); we report the mean score $\times 100$. Lower is safer for both metrics.

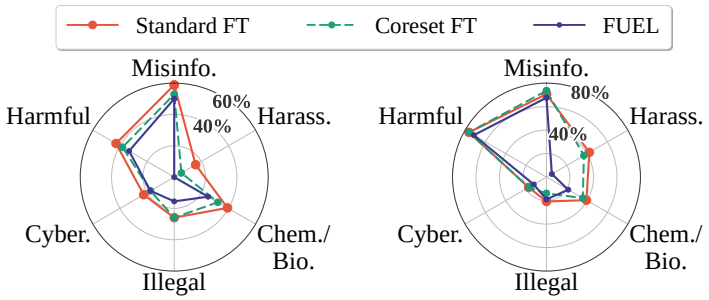
Table 3: Target-task performance and safety of Llama-2-7B-Chat under Standard FT, Coreset FT, and FUEL. Perf. (↑): ROUGE-1 on SQL-Gen, accuracy on GSM8K. Safety (↓): HarmBench ASR and StrongREJECT score. All values in 0–100; **bold**: best among the three FT methods per block.

Method	Target	Safety	
	Perf. ↑	HB ASR ↓	SR Score ↓
FT on **SQL-Gen** (ROUGE-1)			
Base (w/o FT)	19.6	1.0	2.1
Standard FT	**97.8**	39.0	28.0
Coreset FT	97.2	36.5	21.1
FUEL (ours)	97.2	**30.5**	**14.5**
FT on **GSM8K** (Acc)			
Base (w/o FT)	27.1	1.0	2.1
Standard FT	**33.2**	33.5	16.4
Coreset FT	32.2	29.0	13.0
FUEL (ours)	33.0	**23.5**	**11.9**

FUEL improves safety robustness under subsequent fine-tuning. As shown in **Table** 3, subsequent fine-tuning substantially erodes the safety of Llama-2-7B-Chat: standard FT increases HarmBench ASR from 1.0% to 39.0% after SQL-Gen fine-tuning and to 33.5% after GSM8K fine-tuning, with similarly observed increases in StrongREJECT. Coreset FT provides partial protection, reducing ASR by 2.5 and 4.5 points, respectively. FUEL performs best across all safety metrics, further reducing ASR by 6.0 and 5.5 points to 30.5% and 23.5%, respectively, and nearly halving the StrongREJECT score of standard FT on SQL-Gen (14.5 vs. 28.0). Meanwhile, FUEL keeps target performance within 0.6 points of standard FT. **Fig.** 7 further decomposes ASR by HarmBench category, where a smaller enclosed area indicates stronger safety. FUEL yields the smallest enclosed area under both target tasks, with the largest ASR reductions over standard FT in the harassment and chemical/biological categories.

6 Conclusion

We study capability control in LLM fine-tuning under a practical target-data-only setting, where the non-target capabilities to be preserved are unknown *a priori*. We first show that data selection alone is insufficient: even an oracle coreset constructed with privileged preservation knowledge cannot fully prevent capability degradation. Building on this observation, we introduce FUEL, which couples coreset fine-tuning with non-coreset unlearning in a leader–follower framework, using the excluded data as a corrective signal rather than discarding it. Across multiple tasks and model scales, FUEL maintains strong target-task performance while substantially reducing degradation of non-target capabilities, and it further improves the robustness of safety-aligned models under subsequent benign fine-tuning. Overall, our results suggest a broader role for machine unlearning beyond data deletion: unlearning can serve as a constructive mechanism for helping models retain existing capabilities while learning new ones.



(a) FT on GSM8K (b) FT on SQL-Gen

Figure 7: HarmBench ASR (%) by semantic category for Llama-2-7B-Chat fine-tuned on (a) GSM8K and (b) SQL-Gen. Each axis reports the per-category ASR; a smaller enclosed area indicates better safety.

Limitations and future work. Although FUEL is effective in our experiments, its capability-preservation benefits lack rigorous theoretical guarantees. Moreover, a residual gap remains relative to the original model before fine-tuning, leaving the achievable limit of capability recovery unclear. Future work will establish theoretical guarantees for capability control during fine-tuning and extend FUEL to larger models and broader post-training settings, including reinforcement learning.

AI USE STATEMENT

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks that require disclosure under the ICLR 2027 AI Policy for Authors. Specifically, we have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat datasets, or interpret results; all research ideas, methods, and experimental designs in this paper are our own. Generating synthetic datasets and supporting qualitative or thematic data analysis are not applicable to this work, as all datasets we use are publicly available and our analyses are quantitative. Additionally, we used generative AI tools for the following tasks with recommended disclosure: checking grammar to improve the readability of the paper, identifying relevant literature, helping fix bugs in our code, and assisting in the creation of figures. Apart from this bug fixing, no AI tools were involved in research execution; all methods were implemented by the authors. We have reviewed all AI-assisted work: all AI-suggested language edits and bug fixes were manually examined by the authors, all data presented in the figures were carefully verified against our experimental results, and all references identified with AI assistance were verified through a manual literature survey. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to mitigate the degradation of general capabilities and safety alignment caused by LLM fine-tuning. FUEL reduces but does not totally eliminate these side effects. Models fine-tuned with FUEL should therefore not be deployed without further safety evaluation. All experiments use publicly available models and datasets; harmful prompts from HarmBench and StrongREJECT are used solely for evaluation, and no harmful outputs are presented. This work involves no human subjects or personal data, and all authors adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We have taken several measures to facilitate the reproducibility of our results. The complete training procedure is given in Algorithm 1. The base models, datasets, evaluation tasks, metrics, and key hyperparameters are specified in Sec. 5.1 for capability control and in Sec. 5.2 for safety preservation, where safety is measured with the official HarmBench classifier and StrongREJECT evaluator. The sensitivity of FUEL to the unlearning strength ρ is analyzed in Fig. 4, and fine-grained per-task results for both model scales, as well as coreset-selection analyses for all target tasks, are provided in Appendices A and B. All models and datasets used in this work are publicly available. We will release our code upon publication to support the reproduction of our results.

REFERENCES

Youssef Allouah, Rachid Guerraoui, and Sanmi Koyejo. Distributional machine unlearning via selective data removal. In *International Conference on Learning Representations*, volume 2026, pp. 119258–119289, 2026.

b-mc2. sql-create-context dataset, 2023. URL https://huggingface.co/datasets/b-mc2/sql-create-context. This dataset was created by modifying data from the following sources: Zhong et al. (2017); Yu et al. (2018).

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned llms. *arXiv preprint arXiv:2502.17424*, 2025.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Y Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations*, volume 2024, pp. 34196–34216, 2024.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Yulin Chen, He He, and Chen Zhao. The unlearnability phenomenon in RLVR for language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IiFFUPgSkV.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

Simin Fan, Dimitris Paparas, Natasha Noy, Binbin Xiong, Noveen Sachdeva, and Berivan Isik. The magic correlations: Understanding knowledge transfer from pretraining to supervised fine-tuning. *arXiv preprint arXiv:2602.11217*, 2026.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Zihan Guan, Mengxuan Hu, Ronghang Zhu, Sheng Li, and Anil Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety, 2025. URL https://arxiv.org/abs/2505.06843.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. *arXiv preprint arXiv:2404.01099*, 2024.

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. *Advances in neural information processing systems*, 37:65072–65094, 2024.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim F Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555, 2024a.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024b.

Yancheng Huang, Changsheng Wang, Chongyu Fan, Yicheng Lang, Bingqi Shang, Yang Zhang, Mingyi Hong, Qing Qu, Alvaro Velasquez, and Sijia Liu. Subspace control: Turning constrained model steering into controllable spectral optimization. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=T34BInbQ3z.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Animesh Jha, Arpandeep Khatua, Youssef Allouah, and Sanmi Koyejo. What to forget in unlearning? forget set curation for language models. *arXiv preprint arXiv:2608.14855*, 2026.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, 2018.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, et al. Mitigating the alignment tax of rlhf. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 580–606, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7(2):181–194, 2025.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37:118603–118631, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Soumyadeep Pal, Changsheng Wang, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. LLM unlearning reveals a stronger-than-expected coreset effect in current benchmarks. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=NMIqKUdDkw.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34: 20596–20607, 2021.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024a.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations*, volume 2025, pp. 54911–54941, 2025.

Hadi Reisizadeh, Jinghan Jia, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, Sijia Liu, and Mingyi Hong. Blur: A bi-level optimization approach for llm unlearning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7043–7058, 2026.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

Shoaib Ahmed Siddiqui, Eleni Triantafillou, David Krueger, and Adrian Weller. Position: Capability control should be a separate goal from alignment. *arXiv preprint arXiv:2602.05164*, 2026.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks, 2024.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms, 2024. *URL https://arxiv. org/abs/2408.00761*, 2024.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European symposium on security and privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel

Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pp. 4126–4142. PMLR, 2021.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R 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.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pp. 1112–1122, 2018.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961. IEEE, 2022.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

Yuanshun Yao and Xiaojun Xu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*, 2018.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024a.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pp. 26693–26712. PMLR, 2022.

Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59, 2024b.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

APPENDIX

A FINE-GRAINED CAPABILITY CHANGES ACROSS MODEL SCALES

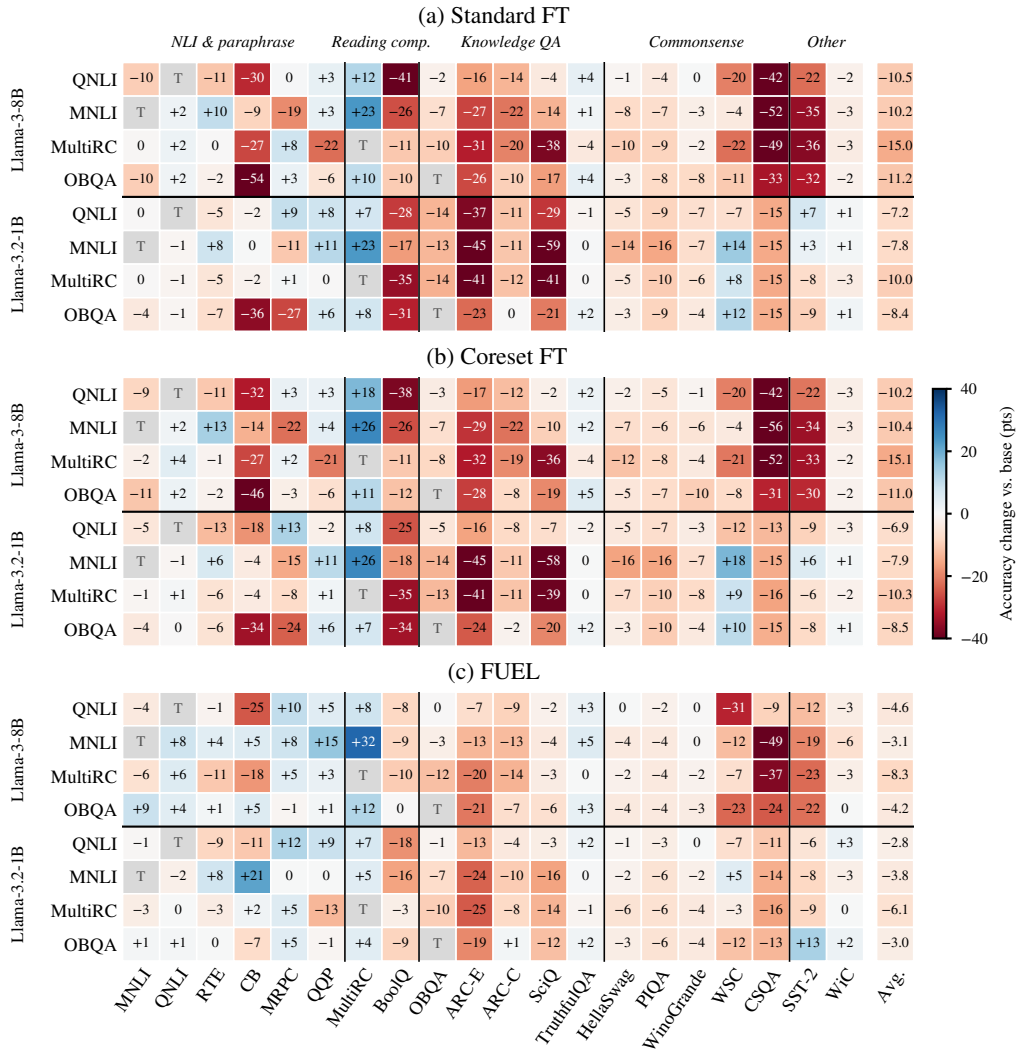
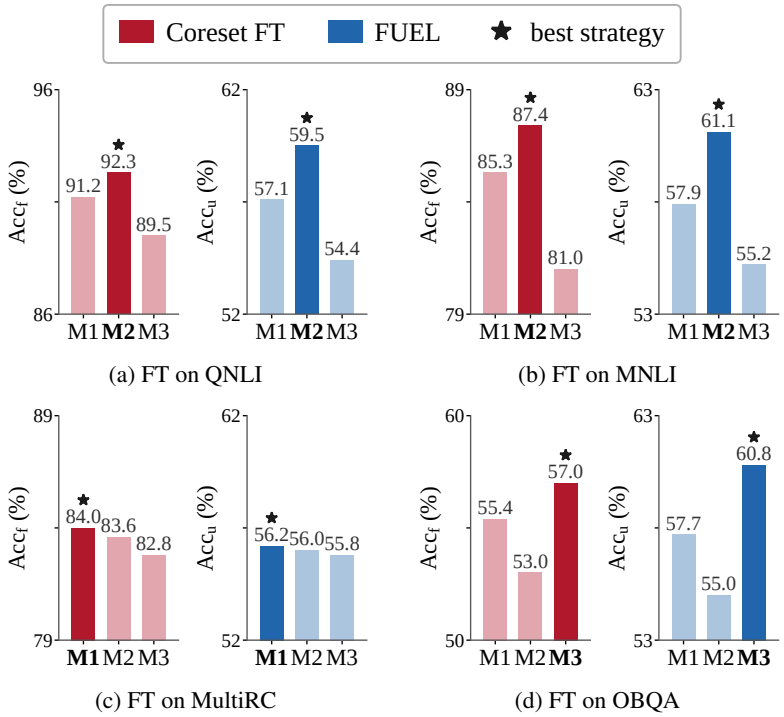


Figure A1: Full per-task capability change Δ_u (%) for Llama-3-8B (top four rows of each panel; same as Fig. 5) and Llama-3.2-1B (bottom four rows) under (a) standard FT, (b) Coreset FT, and (c) FUEL. The presentation format follows Fig. 5.

Fig. A1 presents the per-task capability change Δ_u in (1) on Llama-3.2-1B, following the same visualization protocol as Fig. 5. To facilitate comparison across model scales, each panel stacks the Llama-3.2-1B results (bottom four rows) below the Llama-3-8B results from Fig. 5 (top four rows). Below, we write A→B for the pair of fine-tuning on target task A and evaluating on non-target task B.

As we can see, standard FT on Llama-3.2-1B also induces broad capability degradation, lowering the average non-target accuracy by 7.2–10.0 points across the four target tasks. Notably, the most affected tasks differ across model scales: CSQA and SST-2, which are among the most degraded tasks on Llama-3-8B (down to -52 and -36), are much less affected on Llama-3.2-1B (at most -15 and -9), where the largest drops instead mostly occur on knowledge QA and reading comprehension (e.g., -59 on MNLI→SciQ and -35 on MultiRC→BoolQ). This model dependence further reflects the open-set capability control challenge in Sec. 3: even for a fixed target task, the capabilities at risk are hard to anticipate before fine-tuning. Coreset FT again closely tracks standard FT, with



(a) FT on QNLI (b) FT on MNLI

(c) FT on MultiRC (d) FT on OBQA

Figure A2: Effect of coreset selection on all four target tasks with Llama-3-8B; (b) and (d) are reproduced from Fig. 6. In each sub-figure, the left panel shows $\mathrm{Acc_f}$ of Coreset FT under M1–M3 (random/easy/hard), and the right panel shows $\mathrm{Acc_u}$ of FUEL using the corresponding non-coreset. Stars mark the best strategy, which is consistent across both metrics for every target task.

average changes of -6.9, -7.9, -10.3, and -8.5 (vs. -7.2, -7.8, -10.0, and -8.4) on QNLI, MNLI, MultiRC, and OBQA, respectively.

In contrast, FUEL improves the average capability change to -2.8, -3.8, -6.1, and -3.0, cutting the degradation of standard FT by more than half on average, consistent with Llama-3-8B. The improvement is broad rather than concentrated on a few tasks: among the 28 pairs where standard FT decreases accuracy by at least 10 points, FUEL alleviates the degradation in 27 of them (33 of 36 on Llama-3-8B). Overall, these results further validate that the capability-preservation benefit of non-coreset unlearning is consistent across model scales and broad across target–non-target pairs.

B Effect of Coreset Selection across All Target Tasks

Fig. A2 extends Fig. 6 to all four target tasks, following the same presentation format. As we can see, in addition to MNLI and OpenBookQA, the rankings of M1–M3 under $\mathrm{Acc_f}$ of Coreset FT and $\mathrm{Acc_u}$ of FUEL also coincide on QNLI and MultiRC. On QNLI, M2 $>$ M1 $>$ M3 for both $\mathrm{Acc_f}$ (92.3%, 91.2%, and 89.5%) and $\mathrm{Acc_u}$ (59.5%, 57.1%, and 54.4%). On MultiRC, M1 $>$ M2 $>$ M3 for both $\mathrm{Acc_f}$ (84.0%, 83.6%, and 82.8%) and $\mathrm{Acc_u}$ (56.2%, 56.0%, and 55.8%). Here, the three strategies perform closely in both metrics (differing by at most 1.2 and 0.4 points, respectively), suggesting that when coreset selection barely affects target-task learning, it also barely affects the corrective signal from the non-coreset. Overall, although the best strategy varies across target tasks (M2 for QNLI and MNLI, M1 for MultiRC, and M3 for OpenBookQA), it consistently yields both the best coreset for fine-tuning and the most effective non-coreset for unlearning.