TCHEBYCHEFF UNLEARN: EXPLORING THE PARETO FRONT IN MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning aims to forget target information while keeping the model useful. Applications differ in how much they need to forget and how much utility loss they can accept, necessitating a family of models with different trade-offs instead of a single model. We propose to generate the family by exploring the Pareto front in the forget-retain loss space, treating the forget loss and the retain loss as two objectives. A common approach is linear scalarization, which minimizes a weighted sum of the two losses. However, sweeping the weights cannot reach nonconvex regions of the Pareto front, so some trade-offs remain out of reach. In light of the above limitation of linear scalarization, recent work has also explored adaptive gradient methods from the multi-task learning literature, yet they offer no control over which Pareto-optimal solution they converge to. In this work, we introduce Tchebycheff Unlearn, which uses Tchebycheff scalarization to construct a family of models with different forget–retain trade-offs. We show that varying only the forget reference suffices to represent every attainable Pareto loss pair, with fixed positive task scales and retain reference. We evaluate our approach on the WMDP, TOFU, and MUSE datasets. Our empirical results show that Tchebycheff Unlearn improves both Pareto front coverage and the persistence of intermediate solutions compared with linear scalarization, enabling more comprehensive Pareto front exploration.

1 INTRODUCTION

Large language models (LLMs) can retain private information, copyrighted material, or harmful content that developers or data owners later seek to remove (Maini et al., 2024; Li et al., 2024; Shi et al., 2025). Machine unlearning addresses this need by modifying a trained model to suppress the influence of designated data while preserving capabilities learned from the remaining data. These goals are difficult to reconcile because forgetting and retention act through shared model parameters: updates that suppress target information can also damage unrelated knowledge and language abilities, whereas overly conservative updates may leave the target information accessible. Unlearning must therefore be evaluated in terms of both forgetting quality and retained utility.

Unlearning methods express these goals through a forget loss and a retain loss. The forget loss encourages changes that suppress the designated information. For example, negative preference optimization (NPO) reduces forget-set likelihood relative to a reference model (Zhang et al., 2024), while representation misdirection unlearning (RMU) redirects internal representations (Li et al., 2024). The retain loss encourages the model to preserve its performance on retained data, for example by maintaining likelihood on retained examples or matching the original model's representations. Multi-objective optimization (MOO) methods coordinate these losses to pursue forgetting while limiting damage to retained capabilities (Pan et al., 2025; Jin et al., 2025). We study these objectives in forget–retain loss space, with both losses written as minimization objectives. The *Pareto front* consists of attainable loss pairs for which neither loss can be reduced without increasing the other.

Practical applications often call for a family of unlearned models rather than a single one, because the preferred balance between forgetting and retention depends on the application. One application may seek the strongest forgetting within a retention budget (Entesari et al., 2025), i.e., $\min_\theta \ell_f(\theta)$ s.t. $\ell_r(\theta) \le \tau_r$; another may seek the lowest retention cost that meets a forgetting requirement (Choi et al., 2026), i.e., $\min_\theta \ell_r(\theta)$ s.t. $\ell_f(\theta) \le \tau_f$. As requirements vary, different Pareto solutions become

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.

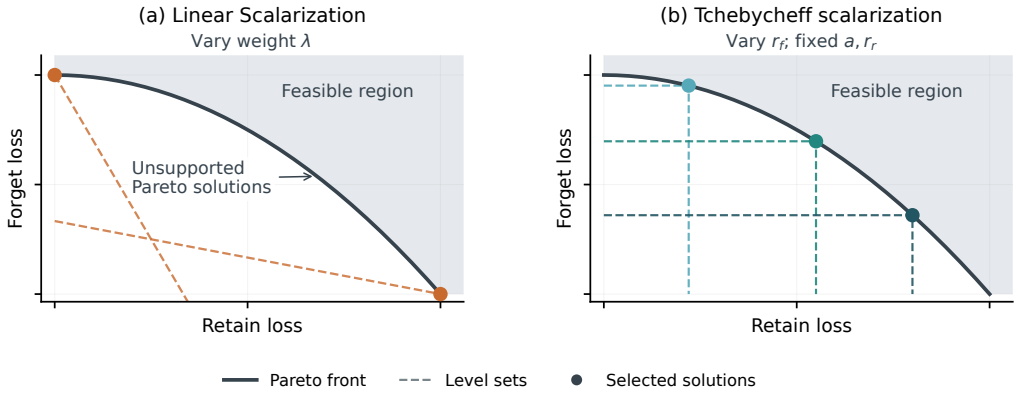

Figure 1: **Finding Pareto solutions does not ensure Pareto front coverage.** On this illustrative nonconvex Pareto front, weighted sums select Pareto-optimal endpoints but miss the interior unsupported Pareto solutions. Tchebycheff scalarization can select those points by changing the forget reference.

preferable. A family of models covering these trade-offs allows a suitable model to be selected for each requirement, thus motivating the exploration of the Pareto front in forget–retain loss space.

For large neural networks, solving these nonconvex constrained problems globally is generally intractable. A practical alternative is to explore a range of forget–retain trade-offs and then select the best model that satisfies the constraints. The most common way to explore trade-offs is linear scalarization, which constructs a family of models by varying the weight in a weighted sum of the forget and retain losses:

$$\min_{\theta} \ \lambda a_f \ell_f(\theta) + (1-\lambda) a_r \ell_r(\theta).$$

Here $a_f, a_r > 0$ are fixed task scales, and $\lambda \in [0,1]$ controls the relative importance of forgetting and retention. Under exact optimization, every positive weighted sum yields a Pareto-optimal solution. However, finding Pareto-optimal solutions does not ensure coverage of the front. The level sets of a weighted sum are parallel lines in forget–retain loss space, and its minimum occurs where one of these lines supports the attainable region from below. Hence, a nonconvex Pareto front can contain solutions that have no such supporting line and therefore cannot be achieved by linear scalarization, regardless of how the weights are chosen (Hu et al., 2023; Liu et al., 2026). In Figure 1, weighted sums can only select the two endpoints, leaving the intermediate Pareto solutions inaccessible.

In light of the above gap, we introduce *Tchebycheff Unlearn*, a framework for exploring the Pareto front in forget–retain loss space through Tchebycheff scalarization (Bowman, 1976; Lin et al., 2024). Given a reference $r = (r_f, r_r)$, it minimizes the largest scaled loss deviation:

$$\min_{\theta} \max \left\{ a_f\big(\ell_f(\theta) - r_f\big),\ a_r\big(\ell_r(\theta) - r_r\big) \right\}.$$

The reference determines which compromise the objective selects, and the geometry allows it to represent Pareto solutions in nonconvex regions that weighted sums cannot reach. We show that, with the task scales and the retain reference held fixed, varying only the forget reference suffices to represent every attainable Pareto loss pair. We therefore construct the model family by choosing a set of forget references and optimizing a separate model at each fixed reference. The resulting models provide candidate compromises for different requirements, as illustrated by the ideal reference sweep in Figure 1.

We evaluate Tchebycheff Unlearn on WMDP (Li et al., 2024), TOFU (Maini et al., 2024), and MUSE (Shi et al., 2025), optimizing each Tchebycheff objective with online mirror descent for Tchebycheff scalarization (OMD–TCH) (Liu et al., 2026). On WMDP, the empirical Pareto front contains nonconvex regions in which Tchebycheff Unlearn provides intermediate trade-offs that weighted-sum sweeps miss. Training trajectories under linear scalarization jump abruptly between high- and low-forget-loss regimes, whereas Tchebycheff Unlearn reaches and sustains intermediate forget-loss levels. On TOFU, the reference sweep improves the attained loss pairs over the weighted-sum sweep, and on MUSE the two yield comparable empirical fronts.

Our contributions are threefold. First, we introduce Tchebycheff Unlearn, a framework based on Tchebycheff scalarization for constructing a family of models with different forget–retain loss trade-offs. Second, we show that varying only the forget reference is sufficient to represent every attainable Pareto loss pair, with fixed positive task scales and retain reference. Third, experiments on standard unlearning benchmarks show that Tchebycheff Unlearn achieves broader empirical Pareto front coverage than weighted sums and sustain intermediate forget-loss levels during training.

2 Related Work

Unlearning objectives and gradient coordination. Unlearning methods address forgetting and retention through both the choice of losses and the coordination of their gradients. NPO (Zhang et al., 2024) suppresses forget-set likelihood relative to a reference model, while RMU (Li et al., 2024) redirects forget-set representations toward a fixed target and preserves retained-data representations. Beyond loss design, multi-task optimization provides several ways to combine competing objectives. NGDiff (Jin et al., 2025) combines normalized forget and retain gradients to balance their contributions to the update. MOLLM (Pan et al., 2025) formulates LLM unlearning as a multi-objective problem and computes a common descent direction for its unlearning and utility-preservation objectives. RLW (Lin et al., 2022) samples task weights at each optimization step. PCGrad (Yu et al., 2020) projects conflicting task gradients onto the normal plane of one another before combining them. IMTL-G (Liu et al., 2021) chooses task weights so that the aggregate gradient has equal projections onto the normalized task gradients. These methods address loss design and gradient coordination, but balancing the objectives within a single run does not establish coverage of the Pareto front. We study this coverage question through a Tchebycheff reference sweep, focusing on forget–retain compromises that linear scalarization can miss.

Constrained and controllable unlearning. Constrained formulations express which compromises meet a specified requirement. Constrained Entropic Unlearning (Entesari et al., 2025) minimizes a forgetting objective subject to a retain-loss bound, using a primal–dual solver PDU. SAUL (Choi et al., 2026) minimizes retain loss subject to a lower bound on forget-set cross-entropy, using sharpness-aware augmented-Lagrangian updates. Other methods express priorities through the optimization structure. BLUR (Reisizadeh et al., 2026) minimizes retain loss over the minimizers of forget loss, giving forgetting a hierarchical priority. EUPMU (Zhou et al., 2025) bounds the increase in retain loss at each update and approximates the resulting constrained update through implicit gradient surgery. Sweeping constraint thresholds could provide a route to exploring different Pareto compromises (Feng et al., 2025), while Tchebycheff Unlearn offers a complementary approach through a Tchebycheff reference sweep without imposing hard loss bounds.

Tchebycheff scalarization and online optimization. Scalarization provides another way to specify and explore objective trade-offs. Weighted sums select solutions through supporting hyperplanes and can miss nonconvex regions of the Pareto front (Hu et al., 2023). Classical Tchebycheff scalarization minimizes the largest scaled deviation from a reference point; its level-set geometry allows it to represent Pareto solutions in such regions (Bowman, 1976; Lin et al., 2024). To facilitate gradient-based optimization of the nonsmooth maximum, Lin et al. (2024) develop a smooth Tchebycheff objective using a log-sum-exp approximation, while OMD–TCH (Liu et al., 2026) takes an online optimization approach, reformulating the maximum over objectives as a maximization over simplex weights and updating those weights by online mirror descent. In this work, we show that varying only the forget reference suffices to represent every attainable Pareto loss pair, and our experiments examine how this reference sweep affects empirical Pareto front coverage and the persistence of intermediate solutions in machine unlearning.

3 Problem Formulation

Let θ_0 be a trained language model, D_f the data designated for forgetting, and D_r the retained data. The forget loss $\ell_f(\theta)$ encourages suppression of the designated information, while the retain loss $\ell_r(\theta)$ encourages preservation of performance on retained data. Both are written as minimization objectives. Unlearning starts from θ_0 and updates parameters in a domain Θ, and our objective is to construct a family of models spanning different Pareto compromises between these two losses.

We consider loss formulations based on two widely used unlearning methods: representation-based RMU (Li et al., 2024) and likelihood-based NPO (Zhang et al., 2024). We use RMU forget and retain losses on WMDP and MUSE, and pair NPO with retained-data cross-entropy on TOFU.

Representation losses. RMU (Li et al., 2024) defines both objectives using hidden representations. Let $h_\theta(x,t) \in \mathbb{R}^d$ denote the representation at a selected layer and token position t of sequence x, and let $\mathbf{u}(x)$ be a random unit vector and held fixed during training. With steering magnitude $c > 0$, the forget and retain losses are

$$\ell_f^{\mathrm{RMU}}(\theta) = \mathbb{E}_{(x,t)\sim D_f}\left[\frac{1}{d}\left\|h_\theta(x,t) - c\mathbf{u}(x)\right\|_2^2\right], \; \ell_r^{\mathrm{RMU}}(\theta) = \mathbb{E}_{(x,t)\sim D_r}\left[\frac{1}{d}\left\|h_\theta(x,t) - h_{\theta_0}(x,t)\right\|_2^2\right].$$

The forget loss redirects representations toward the fixed target, while the retain loss penalizes deviations from the original model's representations.

Likelihood losses. Let $\pi_\theta(y \mid x)$ be the probability of response y given prompt x, and let π_{θ_0} be the frozen initial model. Letting $\sigma(z) = 1/(1+e^{-z})$ and $\beta > 0$ be an inverse-temperature parameter, NPO (Zhang et al., 2024) reduces forget-set likelihood relative to this model through

$$\ell_f^{\mathrm{NPO}}(\theta) = -\frac{2}{\beta}\mathbb{E}_{(x,y)\sim D_f}\left[\log\sigma\left(-\beta\log\frac{\pi_\theta(y \mid x)}{\pi_{\theta_0}(y \mid x)}\right)\right].$$

We pair this forget loss with retained-data cross-entropy (CE), written as

$$\ell_r^{\mathrm{CE}}(\theta) = -\mathbb{E}_{(x,y)\sim D_r}[\log\pi_\theta(y \mid x)].$$

Given the forget and retain losses defined above, the loss vector and its attainable set are

$$\ell(\theta) = (\ell_f(\theta), \ell_r(\theta)), \qquad \mathcal{L} = \{\ell(\theta) : \theta \in \Theta\}.$$

A loss vector ℓ dominates ℓ' if $\ell_f \le \ell'_f$ and $\ell_r \le \ell'_r$, with at least one strict inequality. An attainable loss vector is Pareto-optimal if no other attainable vector dominates it, and the set of these vectors is the *Pareto front* $\mathcal{P}(\mathcal{L})$. A nonconvex Pareto front may contain loss vectors that cannot be obtained by linear scalarization, regardless of how the weights are chosen (Hu et al., 2023; Liu et al., 2026).

4 Tchebycheff Unlearn

Tchebycheff Unlearn constructs a family of candidate models by optimizing Tchebycheff objectives at different references. We first establish how this objective family represents Pareto trade-offs in forget–retain loss space, then describe how to optimize it in a reference sweep.

4.1 Pareto front exploration with Tchebycheff scalarization

In multi-objective optimization, Tchebycheff scalarization minimizes the largest scaled deviation of the objectives from a reference point (Bowman, 1976; Lin et al., 2024). In the context of machine unlearning, we apply this scalarization to the forget loss $\ell_f(\theta)$ and the retain loss $\ell_r(\theta)$, yielding

$$\min_{\theta\in\Theta} S_{a,r}\big(\ell(\theta)\big), \qquad S_{a,r}(\ell) = \max\left\{a_f(\ell_f - r_f),\; a_r(\ell_r - r_r)\right\},$$

where $a_f, a_r > 0$ are fixed task scales and $r = (r_f, r_r)$ is the reference point. These reference coordinates control the selected compromise.

In linear scalarization, a weighted sum attains its minimum where one of its level lines supports the attainable loss set from below (Hu et al., 2023; Liu et al., 2026). The Tchebycheff objective has a different geometry: an upper bound on its value imposes simultaneous upper bounds on both losses. For any level $t \in \mathbb{R}$, we have

$$S_{a,r}(\ell) \le t \quad \Longleftrightarrow \quad \begin{cases} \ell_f \le r_f + t/a_f, \\ \ell_r \le r_r + t/a_r. \end{cases} \tag{1}$$

In retain–forget loss space, these bounds define a corner at $(r_r + t/a_r,\; r_f + t/a_f)$, where both scaled loss deviations equal t. By choosing the reference appropriately, we can place this corner at a desired

Pareto loss pair. Any attainable pair with a smaller Tchebycheff objective would then have to reduce both losses relative to that target, contradicting its Pareto optimality. The target therefore minimizes the Tchebycheff objective. The following proposition formalizes this construction and shows that varying only the forget reference is sufficient, with fixed positive task scales and retain reference. The proof is deferred to Appendix A.

Proposition 1 (Single-coordinate sweep). *Fix $a_f, a_r > 0$ and a retain reference $\bar{r}_r \in \mathbb{R}$. For every $\ell^\star = (\ell_f^\star, \ell_r^\star) \in \mathcal{P}(\mathcal{L})$, choose*

$$r_f^* = \ell_f^\star - \frac{a_r}{a_f}(\ell_r^\star - \bar{r}_r).$$

Then $\ell^\star$ is the unique minimizing loss vector of $S_{a,(r_f^*,\bar{r}_r)}$ over $\mathcal{L}$.

As r_f ranges over $\mathbb{R}$, the resulting solution family represents every attainable Pareto loss pair with fixed positive task scales and retain reference. Therefore, a reference sweep can represent the nonconvex Pareto regions that linear scalarization may not reach, thus supplying candidates for different application requirements.

4.2 Optimizing the Tchebycheff objective

For each reference in the sweep, we need to optimize the corresponding Tchebycheff objective. A practical challenge is that directly optimizing this maximum can lead to oscillation (Lin et al., 2024; Liu et al., 2026). When one scaled deviation is larger, the update follows only that objective's gradient. As the losses evolve, the active objective can switch, causing abrupt changes in the update direction. To address this issue, following Liu et al. (2026), we first rewrite the maximum over the two scaled deviations as a maximization over their relative weights, yielding the equivalent min–max formulation

$$\min_{\theta\in\Theta} \max_{\mathbf{w}\in\Delta_2} \sum_{i\in\{f,r\}} w_i a_i\big(\ell_i(\theta) - r_i\big), \qquad \Delta_2 = \{\mathbf{w} \geq 0 : w_f + w_r = 1\}.$$

In our optimization algorithm, we treat the relative weights $\mathbf{w} = (w_f, w_r) \in \Delta_2$ as optimization variables and update them jointly with θ using online mirror descent OMD–TCH (Liu et al., 2026). The model update aims to reduce the current weighted loss, while the weight update shifts emphasis toward the larger scaled deviation. Because the weights evolve from their previous values, the gradient mixture can change more gradually, helping mitigate oscillation.

More precisely, at iteration k, let $\widehat{\ell}_{k,i}$ denote the sampled loss for objective i at model θ_k. Holding the current weights $\mathbf{w}_k$ fixed, we combine the two loss gradients and take an AdamW step:

$$g_k = \sum_{i\in\{f,r\}} w_{k,i} a_i \nabla_\theta \widehat{\ell}_{k,i}, \qquad \theta_{k+1} = \mathrm{AdamW}(\theta_k, g_k).$$

The loss evaluations also determine the next relative weights. Since the inner objective is linear in $\mathbf{w}$, its gradient with respect to w_i is the scaled deviation $a_i(\widehat{\ell}_{k,i} - r_i)$. For this step, we adopt Projected Gradient Descent (PGD) and Exponentiated Gradient (EG), the two OMD–TCH weight updates described by Liu et al. (2026). With step size $\eta > 0$, the PGD update is

$$\text{PGD:} \quad \mathbf{w}_{k+1} = \Pi_{\Delta_2}\left(\mathbf{w}_k + \eta \begin{bmatrix} a_f(\widehat{\ell}_{k,f} - r_f) \\ a_r(\widehat{\ell}_{k,r} - r_r) \end{bmatrix}\right), \tag{2}$$

where Π_{Δ_2} projects onto the simplex. The EG update is

$$\text{EG:} \quad w_{k+1,i} = \frac{w_{k,i} \exp\Big[\eta a_i(\widehat{\ell}_{k,i} - r_i)\Big]}{\sum_{j\in\{f,r\}} w_{k,j} \exp\Big[\eta a_j(\widehat{\ell}_{k,j} - r_j)\Big]}, \qquad i \in \{f, r\}. \tag{3}$$

Both rules ascend the inner objective using losses evaluated at θ_k, and the new weights affect the following model update. If the scaled forget deviation exceeds the scaled retain deviation, the relative weight on forgetting increases unless already at the boundary, while the opposite comparison favors retention.

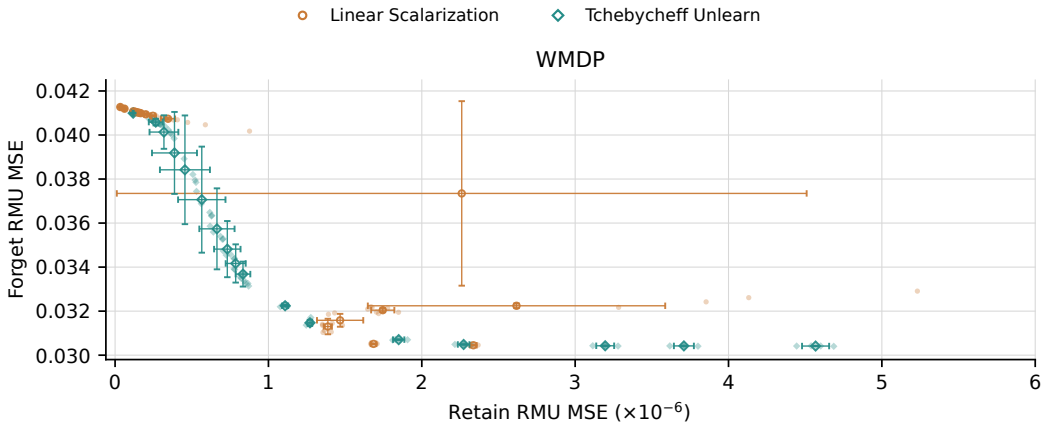

Figure 2: **Tchebycheff Unlearn reaches intermediate WMDP trade-offs missed by weighted sums.** The sweeps use $\lambda \in [0.01, 0.999]$, $r_f \in [0.02, 0.06]$, and $r_r = 0$ at 2,400 updates. Figure 6 provides the annotated version with every sweep setting labeled over the full loss range.

These adaptive weights also explain how the reference guides model optimization. At the same model, scales, and current weights, lowering r_f increases the scaled forget deviation and favors greater emphasis on forgetting in the next weight update. If the scaled retain deviation subsequently becomes larger, the update shifts emphasis toward retention. A fixed reference can therefore induce different gradient mixtures as the losses evolve, whereas linear scalarization keeps its coefficients fixed. This response can help sustain intermediate compromises, a behavior we examine through training trajectories in Section 5.2.

5 Experiments

Our experiments examine the range of candidate models supplied by Tchebycheff Unlearn in forget–retain loss space and the persistence of their intermediate compromises.

Experimental setting. WMDP uses Zephyr-7B-beta with RMU on biology and cybersecurity forget corpora and WikiText retention data. TOFU uses the benchmark's full-data-finetuned Llama-3.2-1B-Instruct model with NPO and retain cross-entropy, separately on forget01 and forget05. MUSE-News and MUSE-Books each use their benchmark target model with RMU. Each comparison starts from the same model and keeps the component losses and task scales fixed across methods. The detailed setups of task scales and relative weight update methods in each dataset are deferred to Appendix B.1.

Each run starts from θ_0 and returns the checkpoint iterate specified in the experimental protocol. The sweeps use five seeds per setting. The WMDP seed-0 comparison additionally includes NGDiff (Jin et al., 2025), RLW (Lin et al., 2022), PCGrad (Yu et al., 2020), IMTL-G (Liu et al., 2021), BLUR (Reisizadeh et al., 2026), EUPMU (Zhou et al., 2025), PDU (Entesari et al., 2025), and SAUL (Choi et al., 2026) as baseline methods, using their update rules with the same RMU losses. Appendix B provides the experimental settings and annotated loss-space sweeps.

In the loss-space plots that include five random seeds, open markers show per-setting means of the losses across repeated runs, error bars show one sample standard deviation, and faint points show individual outcomes. For each seed, the nondominated loss pairs form an *empirical Pareto front*.

5.1 WMDP: Exploring a Nonconvex Empirical Pareto Front

Figure 2 shows that weighted-sum outcomes concentrate in two regions: low retain loss with relatively high forget loss, and lower forget loss with substantially higher retain loss. Tchebycheff Unlearn supplies intermediate loss pairs between these regions. These models reduce forget loss relative to the first group while incurring less retain loss than the second. This coverage gap persists after refining

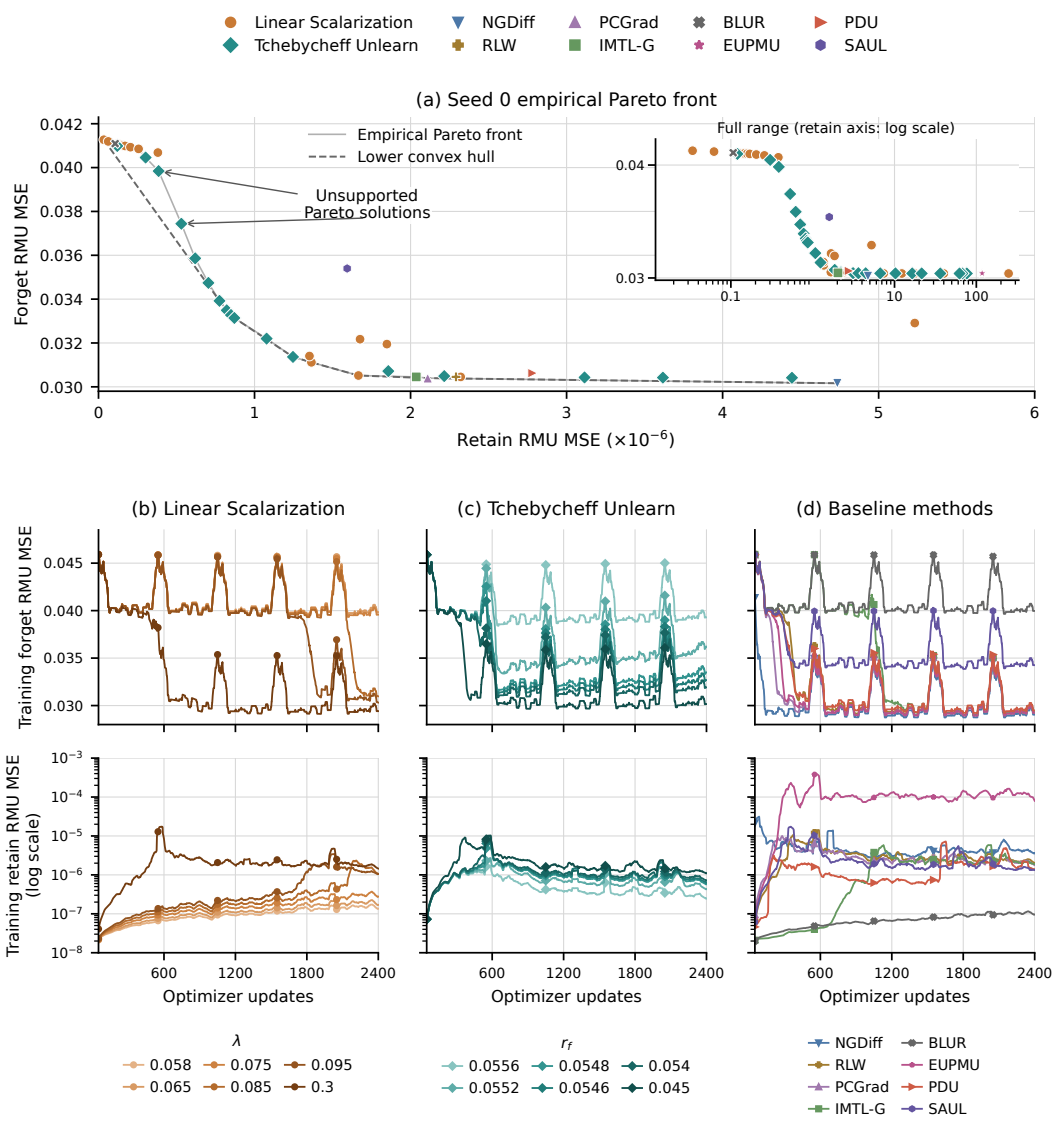

Figure 3: **Intermediate WMDP compromises and their training dynamics.** (a) Loss pairs at 2,400 updates from both sweeps and 8 baselines. The dashed line shows the lower convex hull of the observed loss pairs; arrows identify unsupported reference solutions, i.e., nondominated loss pairs that cannot minimize any positively weighted sum over the observed set. The inset includes the full loss range on a logarithmic retain axis. (b–d) Training curves for Linear Scalarization, Tchebycheff Unlearn, and the baseline methods. Curves show 50-update means over updates 50–2,400.

the weight grid around the transition. The reference sweep therefore adds models at intermediate retention costs, expanding the choices available between the two regimes.

This coverage gap has a geometric explanation. Figure 3(a) compares seed-0 endpoints from both sweeps and the eight baselines in the original linear loss coordinates. The joint empirical Pareto front contains intermediate reference solutions above its lower convex hull. These unsupported solutions cannot minimize any positively weighted sum, regardless of how densely the weights are swept. Tchebycheff Unlearn's contribution is the range of intermediate compromises, which remains visible when these additional baseline methods are included.

These intermediate solutions illustrate the value of the representation result in Proposition 1: varying only r_f can represent every attainable Pareto loss pair, including those in nonconvex regions. This result motivates refining the reference grid to explore the intermediate region at finer resolution.

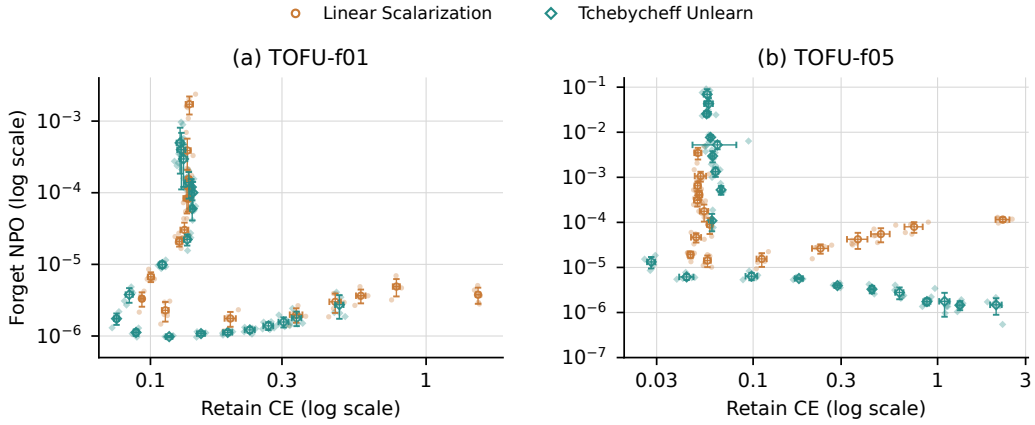

Figure 4: **Tchebycheff Unlearn yields improved empirical loss fronts on TOFU.** Loss pairs for (a) TOFU-f01 at 300 updates and (b) TOFU-f05 at 600 updates. The sweeps use $\lambda \in [0.01, 0.999]$, $r_f \in [-6, 8]$, and $r_r = 0$. Figures 7-8 provide the annotated versions with sweep settings labeled.

5.2 WMDP: SUSTAINING INTERMEDIATE SOLUTIONS

Having compared the loss pairs reached by the sweeps, we next examine how intermediate compromises develop and persist during training. Figure 3(b–d) connects the seed-0 endpoint comparison to the corresponding training runs. Under linear scalarization, several settings remain near a common high-forget-loss regime before dropping abruptly to a lower one. Changing λ shifts the timing of this drop: for example, the transition occurs earlier for $\lambda = 0.095$ than for $\lambda = 0.085$. Intermediate forget-loss levels along these curves can therefore reflect a brief transition between regimes.

By contrast, Tchebycheff Unlearn reaches distinct intermediate forget-loss regimes and maintains them over extended training intervals. For example, $r_f = 0.0548$ and $r_f = 0.0546$ produce separated intermediate levels despite recurring fluctuations and gradual drift. The corresponding retain losses rise early and then decline, so the retention cost continues to evolve while intermediate forgetting levels persist. The trajectories thus complement the endpoint geometry: Tchebycheff Unlearn both reaches intermediate compromises and sustains their forgetting levels through continued optimization.

Most baseline methods in Figure 3(d) exhibit a limitation similar to that of linear scalarization: their trajectories concentrate in high- or low-forget-loss regimes. SAUL sustains an intermediate forget-loss level, but its final loss pair is dominated by reference-sweep solutions and therefore does not lie on the empirical Pareto front.

5.3 WHY FIXED WEIGHTS CAN BE SENSITIVE NEAR A TRANSITION

The geometry of scalarization also helps explain this contrast in training behavior. Consider a supporting level line that touches two separated Pareto points on opposite sides of an unsupported region. At the corresponding critical weight, the two points have equal scalarized losses. Nearby weights only weakly favor one of them, although their forget–retain compromises differ substantially. Consequently, the optimization state and stochastic updates can influence which region a run reaches, leading to large variation across runs. This pattern is visible at $\lambda = 0.08$ in Figure 2: individual outcomes fall into separated forget-loss regions, while their mean lies between them.

Tchebycheff Unlearn, in contrast, adjusts its relative weights in response to the scaled loss deviations. When $a_r(\ell_r - r_r)$ exceeds $a_f(\ell_f - r_f)$, the weight update favors retention; the reverse inequality favors forgetting. In particular, a decrease in forget loss can shift emphasis toward retention even if retain loss has not increased. With an appropriate reference, this feedback can counter continued movement toward an extreme and help sustain an intermediate compromise. Thus, although the reference stays fixed, the gradient mixture adapts to the current losses, providing a mechanism for maintaining intermediate solutions.

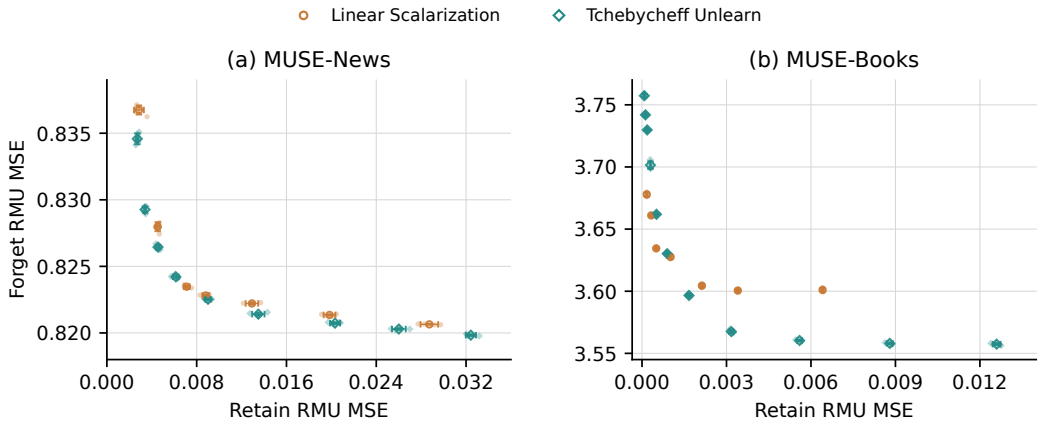

Figure 5: **Tchebycheff Unlearn explores similar compromises on MUSE-News and MUSE-Books.** Loss pairs for (a) MUSE-News and (b) MUSE-Books at 200 updates. The sweeps use $\lambda \in [0.1, 0.9]$ and $r_r = 0$, with $r_f \in [0.6, 1.0]$ for News and $r_f \in [3.5, 4.0]$ for Books. Figures 9-10 provide the annotated versions with sweep settings labeled.

5.4 TOFU AND MUSE: EXPLORATION ACROSS OTHER LOSS LANDSCAPES

We next examine reference sweeps with likelihood-based objectives on TOFU-f01 and TOFU-f05 (Figure 4). In both splits, Tchebycheff Unlearn improves the attained loss pairs over linear scalarization: every model returned by the weighted-sum sweep is dominated by at least one reference-sweep model. Moreover, each dominating reference solution also achieves a lower value of the weighted-sum model's own objective. The TOFU results thus complement the coverage advantage on WMDP with improved solution quality under likelihood-based objectives.

On MUSE-News, both methods provide gradual compromises across a similar loss region (Figure 5(a)), and Tchebycheff Unlearn offers comparable trade-offs. On MUSE-Books, the reference sweep improves some intermediate loss pairs and extends to lower forget losses at higher retention cost (Figure 5(b)). Across both datasets, lowering the forget reference guides the sweep toward lower forget loss at greater retention cost.

6 DISCUSSION AND CONCLUSION

We study how different scalarization techniques shape the exploration of the forget–retain loss landscape, and introduce Tchebycheff Unlearn to construct such a family through a reference sweep using OMD–TCH. We show that, with fixed positive task scales and a fixed retain reference, every attainable Pareto-optimal loss pair is the unique minimizing loss vector for some forget reference, including unsupported Pareto pairs that no weighted sum can select as a global minimizer.

We numerically evaluated the performance of Tchebycheff Unlearn on multiple datasets. On WMDP, reference sweeps provide intermediate loss compromises absent from the tested weighted-sum sweeps, and representative trajectories maintain intermediate forget-loss levels over extended portions of the observed training interval. On TOFU and MUSE, both methods reach broadly similar loss regions, while reference sweeps of Tchebycheff Unlearn may slightly improve attained solution quality. Together, these results show how Tchebycheff Unlearn can expand the range of candidate models and sustain intermediate solutions. Such broader loss-space exploration provides more choices for applications with different forgetting requirements and retention budgets.

Limitations. Our analysis concerns surrogate-loss trade-offs and does not establish corresponding improvements in behavioral forgetting or retained utility. Practical coverage depends on the sampled references, task scales, and training budget; the representation theorem does not guarantee that finite sweeps and neural optimization recover the full Pareto front. The trajectory evidence comes from a limited training interval and does not establish convergence or uniformly lower variability.

AI USE STATEMENT

In this work, we used generative AI tools for literature search, drafting sections of the paper, language polishing, and running experiments. We have not used AI for generating synthetic datasets or proving mathematical claims. We have reviewed all AI-assisted work. For example, we checked LLM-generated references through a manual literature review. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B specifies the implementation details, training configurations, and complete sweep grids. The code is submitted as supplementary materials.

REFERENCES

V. Joseph Bowman, Jr. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In Hervé Thiriez and Stanley Zionts (eds.), *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*, pp. 76–86. Springer, Berlin, Heidelberg, 1976. doi: 10.1007/978-3-642-87563-2_5.

Jaewan Choi, Junyoung Yang, and Sangdon Park. SAUL: Sharpness-aware augmented-Lagrangian unlearning. *arXiv preprint arXiv:2608.16249*, 2026.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zico Kolter, Zachary Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ZtB34bQI54.

XiaoHua Feng, Yuyuan Li, Chaochao Chen, Li Zhang, Longfei Li, Jun Zhou, and Xiaolin Zheng. Controllable unlearning for image-to-image generative models via ϵ-constrained optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9OJflnNu6C.

Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=6EqUpqMnwl.

Xiaomeng Jin, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Mingyi Hong. Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11278–11294, 2025. URL https://aclanthology.org/2025.naacl-long.563/.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xlr6AUDuJz.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jjtFD8A1Wx.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth Tchebycheff scalarization for multi-objective optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=m4dO5L6eCp.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=IMPnRXEWpvr.

Meitong Liu, Xiaoyuan Zhang, Chulin Xie, Kate Donahue, and Han Zhao. Adaptive online mirror descent for Tchebycheff scalarization in multi-objective learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=MUTRffB3af.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=B41hNBoWLo.

Zibin Pan, Shuwen Zhang, Yuesheng Zheng, Chi Li, Yuheng Cheng, and Junhua Zhao. Multi-objective large language model unlearning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 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. URL https://aclanthology.org/2026.eacl-long.331/.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Shiji Zhou, Tianbai Yu, Zhi Zhang, Heng Chang, Xiao Zhou, Dong Wu, and Han Zhao. Efficient utility-preserving machine unlearning with implicit gradient surgery. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vzcVDwLtwA.

A Proof of Proposition 1

Proposition 1 (Single-coordinate sweep). *Fix $a_f, a_r > 0$ and a retain reference $\bar{r}_r \in \mathbb{R}$. For every $\ell^\star = (\ell_f^\star, \ell_r^\star) \in \mathcal{P}(\mathcal{L})$, choose*

$$r_f^* = \ell_f^\star - \frac{a_r}{a_f}(\ell_r^\star - \bar{r}_r).$$

Then $\ell^\star$ is the unique minimizing loss vector of $S_{a,(r_f^*,\bar{r}_r)}$ over $\mathcal{L}$.

Proof. Let $\ell^\star = (\ell_f^\star, \ell_r^\star)$ be the target Pareto loss pair, and set $v = a_r(\ell_r^\star - \bar{r}_r)$. The prescribed forget reference makes the two scaled deviations equal at this target:

$$a_f(\ell_f^\star - r_f^*) = a_f\left[\ell_f^\star - \left(\ell_f^\star - \frac{a_r}{a_f}(\ell_r^\star - \bar{r}_r)\right)\right] = v,$$
$$a_r(\ell_r^\star - \bar{r}_r) = v.$$

Hence $S_{a,(r_f^*,\bar{r}_r)}(\ell^\star) = \max\{v, v\} = v$.

Now consider any attainable loss vector $\ell = (\ell_f, \ell_r) \in \mathcal{L}$ satisfying $S_{a,(r_f^*,\bar{r}_r)}(\ell) \le v$. Since this objective is the maximum of the two scaled deviations, each deviation must be at most v:

$$a_f(\ell_f - r_f^*) \le v, \qquad a_r(\ell_r - \bar{r}_r) \le v.$$

Dividing by the positive task scales and substituting the definitions of r_f^* and v gives

$$\ell_f \le r_f^* + \frac{v}{a_f} = \ell_f^\star, \qquad \ell_r \le \bar{r}_r + \frac{v}{a_r} = \ell_r^\star.$$

If $\ell \neq \ell^\star$, at least one of these inequalities is strict, so ℓ dominates $\ell^\star$. This contradicts the Pareto optimality of the target. Therefore $\ell^\star$ is the only attainable loss vector whose objective value is at most v.

Because $\ell^\star$ is itself attainable and achieves v, the minimum is attained there, and every distinct attainable loss vector has a strictly larger objective value. Thus $\ell^\star$ is the unique minimizing loss vector. It is worth noting that this uniqueness concerns the loss pair; multiple parameter values may realize the same pair. □

B Experimental Details and Annotated Sweeps

B.1 Experimental Settings

The implementation is based on OpenUnlearning (Dorna et al., 2025). Table 1 summarizes the training and sweep configurations. All sweeps use seeds 0–4. Within each study, runs for all sweep settings start from the same initial model and are evaluated at the listed checkpoint. The complete λ and r_f grids are labeled in Figures 6–10.

Table 1: Training and sweep settings. Sweep size gives the numbers of λ and r_f values, respectively. Solver and η apply to reference sweeps.

Study	Sweep size	Model LR	Batch Size	Updates	(a_f, a_r)	Solver (η)
WMDP	23 / 25	10^{-5}	4	2,400	$(1, 30)$	PGD (0.1)
TOFU-f01	16 / 20	10^{-5}	32	300	$(1, 1)$	EG (0.01)
TOFU-f05	16 / 20	10^{-5}	32	600	$(1, 1)$	EG (0.01)
MUSE-News	7 / 9	5×10^{-5}	32	200	$(1, 1)$	EG (0.1)
MUSE-Books	7 / 11	10^{-5}	32	200	$(1, 10)$	EG (0.1)

Training uses constant model learning rates and zero weight decay. WMDP and TOFU use AdamW; MUSE uses paged AdamW. Reference sweeps fix $r_r = 0$ and initialize $\mathbf{w}_0 = (0.5, 0.5)$. The relative weights are updated once per model step from window-mean losses according to Equation (2) or (3). For RMU on WMDP and MUSE, we use layer-7 activations, steering magnitude 6.5, and train only the down-projection weights in layers 5–7.

B.2 WMDP: RMU Loss

We use `HuggingFaceH4/zephyr-7b-beta` with biology and cybersecurity forget corpora and WikiText retention data. Training alternates forget domains, pairing each forget batch with a retain batch. Runs with the same seed share the data stream and steering vectors. Maximum sequence lengths are 512, 768, and 512 for biology, cybersecurity, and retention. Independent evaluation

covers 24,453 biology, 1,000 cybersecurity, and 2,891 retain sequences. MSE is averaged over all hidden-state elements, including padding.

The eight baselines in Figure 3 use the same RMU losses, task scales, trainable parameters, and model optimizer. We adapt their gradient updates to this common setting. NGDiff uses a fixed learning rate without AutoLR. BLUR uses $\gamma = 0.0375$. EUPMU initializes its auxiliary weight to zero and uses auxiliary Adam with learning rate 0.3, weight decay 0.01, and error parameter 0.003. PDU uses a raw retain-loss budget $b_r = 10^{-6}$, initial dual value 30, and dual step size 1. SAUL uses a forget-loss threshold $\epsilon_f = 0.036$, initial dual value 1, dual step size 10^{-3}, and sharpness-aware perturbation radius $\rho_f = \rho_r = 10^{-4}$. Its forget and retain updates maintain separate AdamW states, and each optimizer call counts as one update.

Figure 3 compares all 48 sweep settings and 8 baselines on seed 0. Its training curves use the 50-update means of minibatch training losses; the displayed trajectories and their corresponding endpoints come from the same runs.

B.3 TOFU: NPO Loss

Both splits use `open-unlearning/tofu_Llama-3.2-1B-Instruct_full` as the initial model and frozen NPO reference. We set $\beta = 0.1$, use maximum sequence length 512, and apply no gradient clipping. Evaluation covers all 40 forget and 3,960 retain examples for f01, and all 200 forget and 3,800 retain examples for f05. Prompt and padding tokens are masked out. NPO applies its nonlinear transform to response-summed NLL differences before averaging examples. Retain CE averages answer-token NLLs within each example, then averages examples equally.

B.4 MUSE-News and MUSE-Books: RMU Loss

We use `muse-bench/MUSE-News_target` and `muse-bench/MUSE-Books_target` as the initial and frozen reference models. Evaluation covers all 407 forget and 802 retain chunks for News, and all 553 forget and 105 retain chunks for Books. The retain chunks come from the `retain1` split. MSE is averaged over valid labeled tokens within each sequence and then across sequences.

B.5 Annotated Loss-Space Sweeps

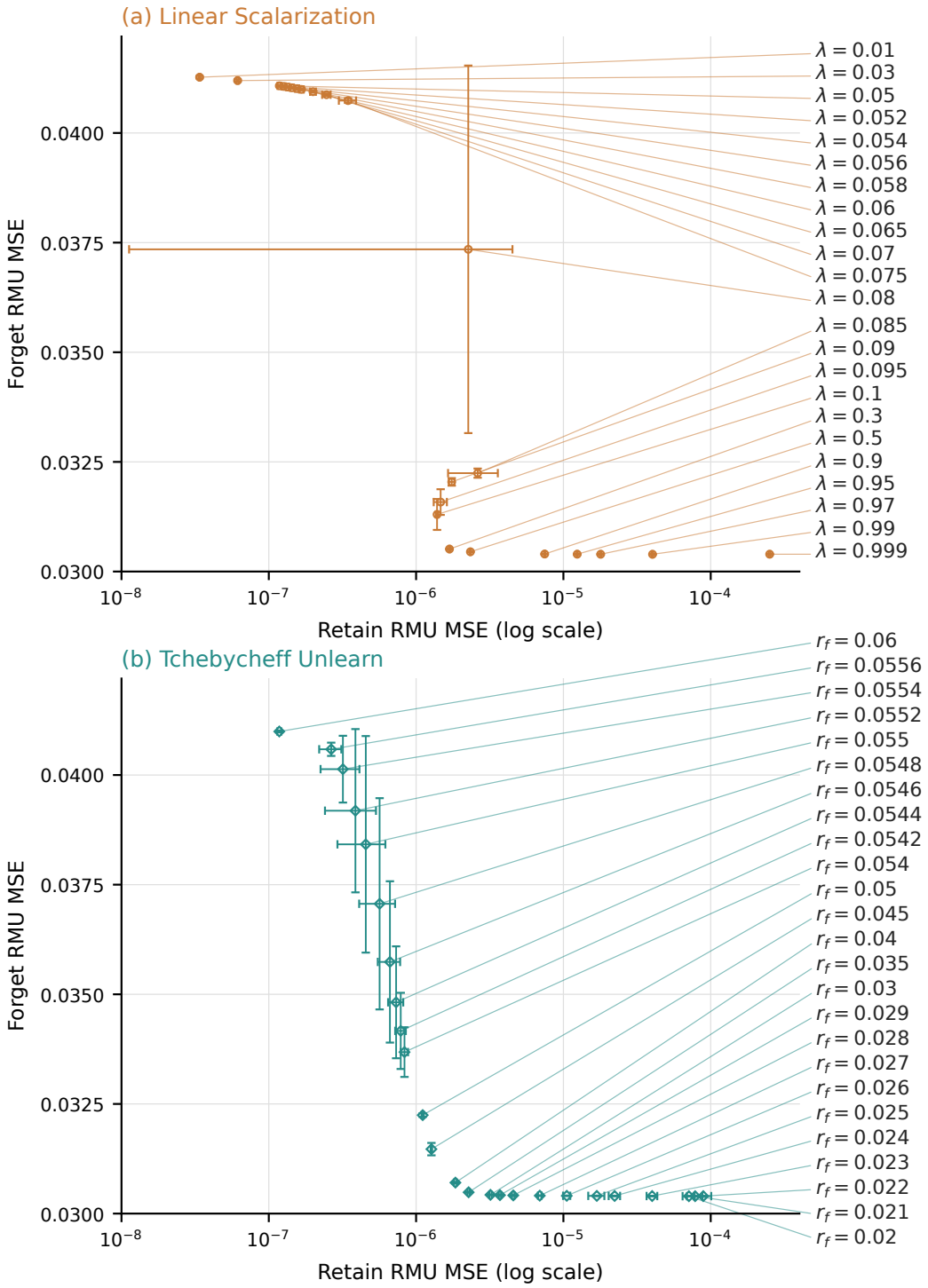

Figure 6: **Annotated WMDP sweep at 2,400 updates.** Detailed version of Figure 2, including the full loss range. Panels share axis limits; the retain axis uses a log scale. From top to bottom, labels list increasing λ for Linear Scalarization and decreasing r_f for Tchebycheff Unlearn.

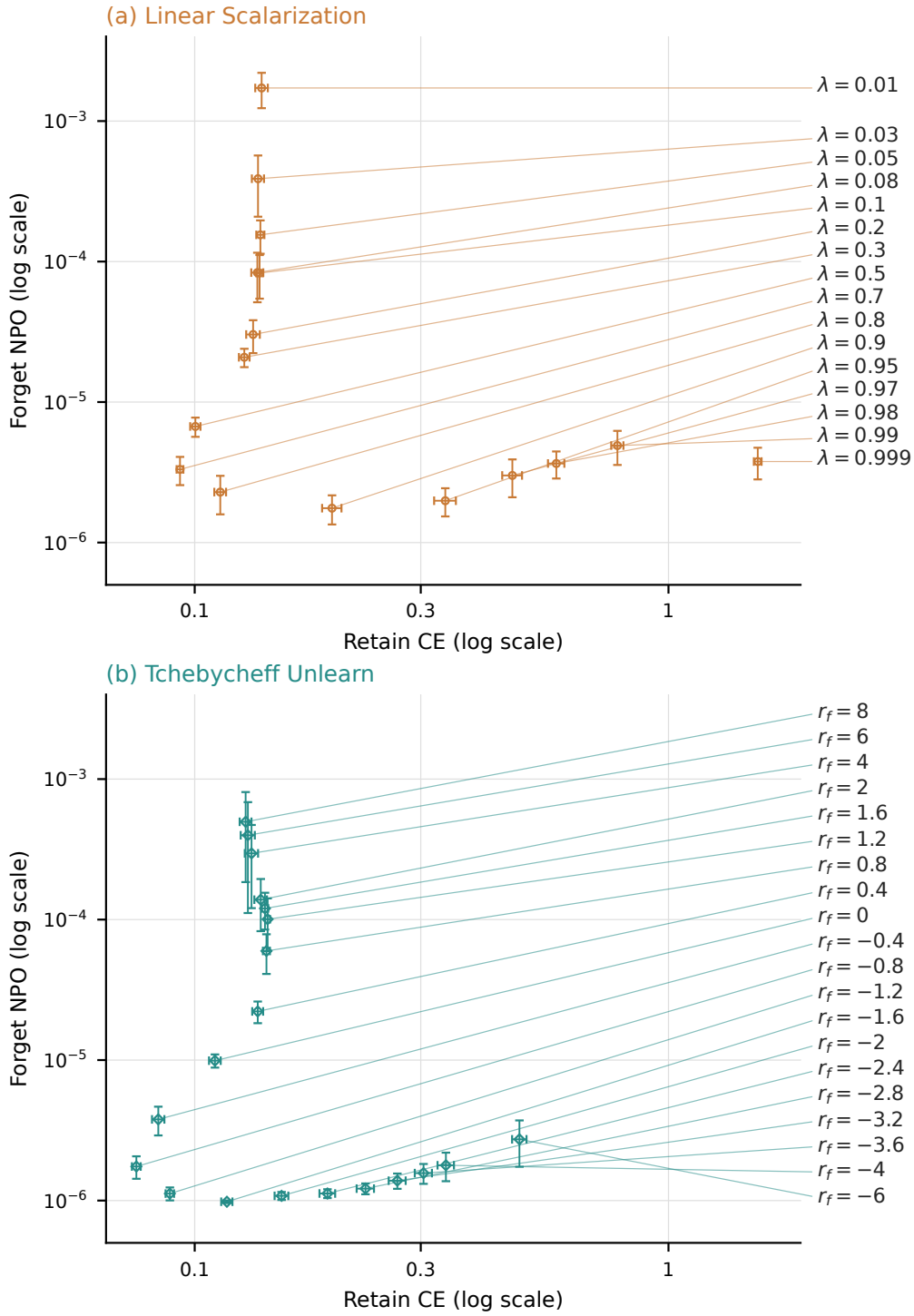

Figure 7: **Annotated TOFU-f01 sweep at 300 updates.** Detailed version of Figure 4(a). Panels share axis limits; both loss axes use a log scale. From top to bottom, labels list increasing λ for Linear Scalarization and decreasing r_f for Tchebycheff Unlearn.

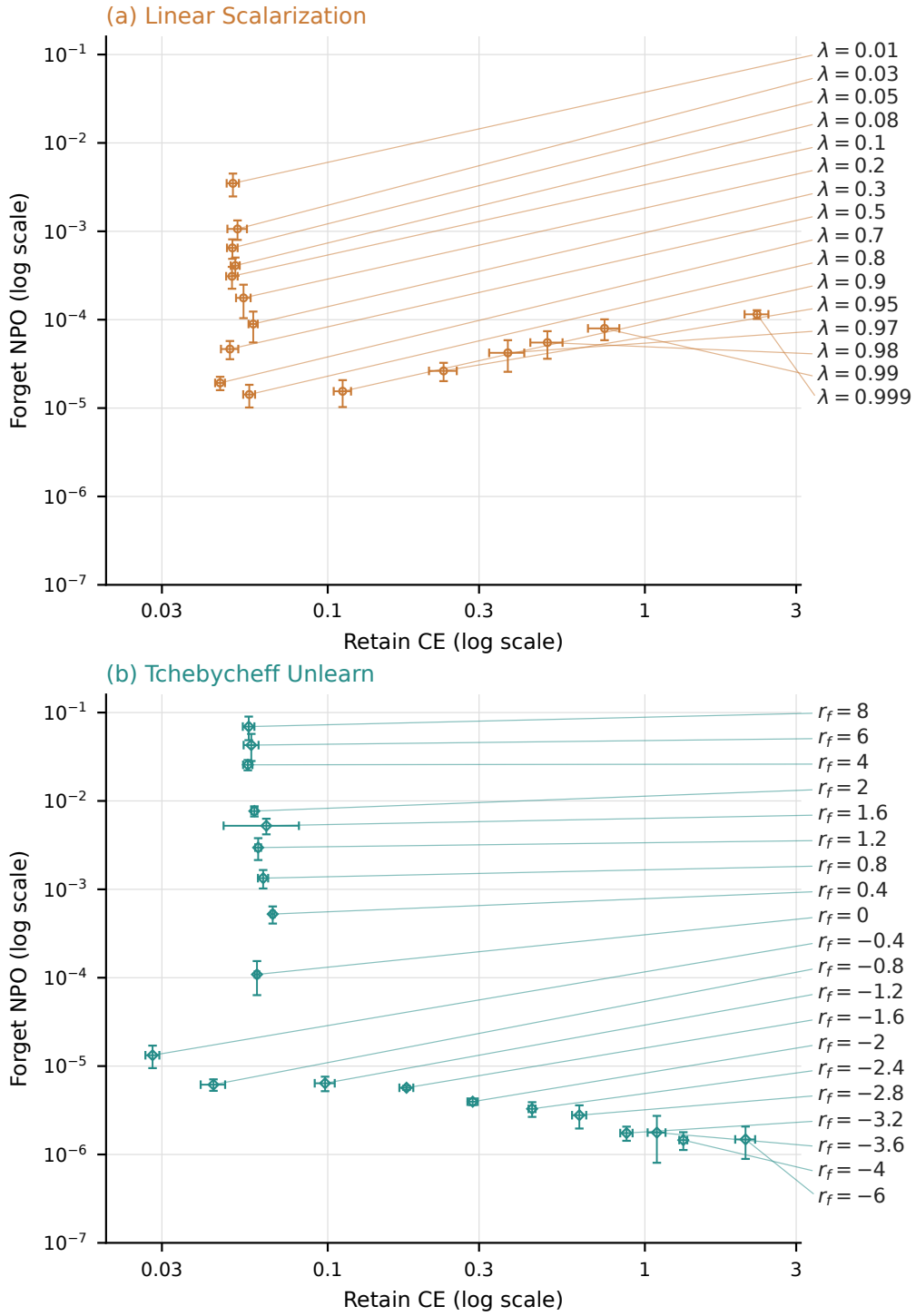

Figure 8: **Annotated TOFU-f05 sweep at 600 updates.** Detailed version of Figure 4(b). Panels share axis limits; both loss axes use a log scale. From top to bottom, labels list increasing λ for Linear Scalarization and decreasing r_f for Tchebycheff Unlearn.

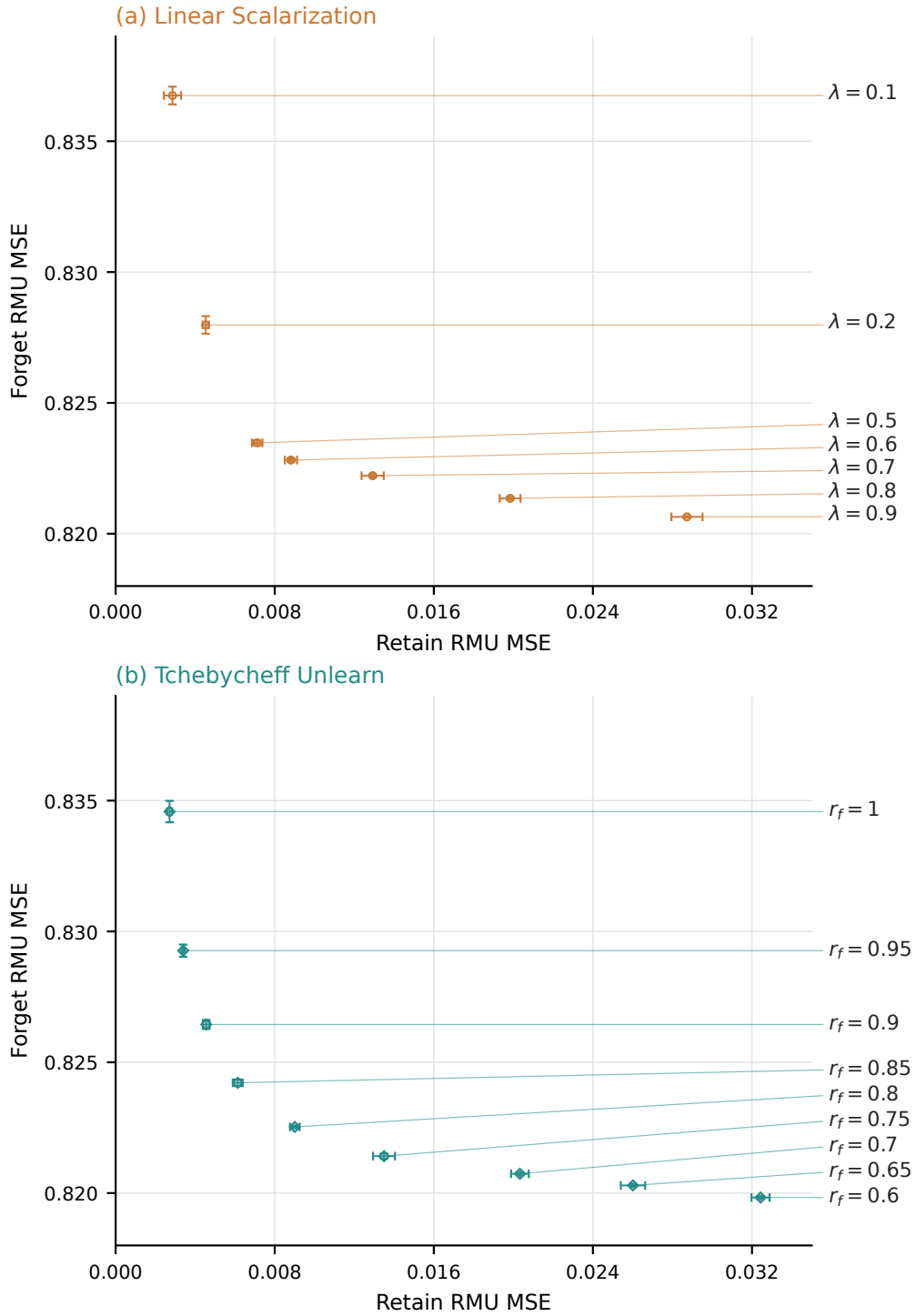

Figure 9: **Annotated MUSE-News sweep at 200 updates.** Detailed version of Figure 5(a). Panels share axis limits; both loss axes use a linear scale. From top to bottom, labels list increasing λ for Linear Scalarization and decreasing r_f for Tchebycheff Unlearn.

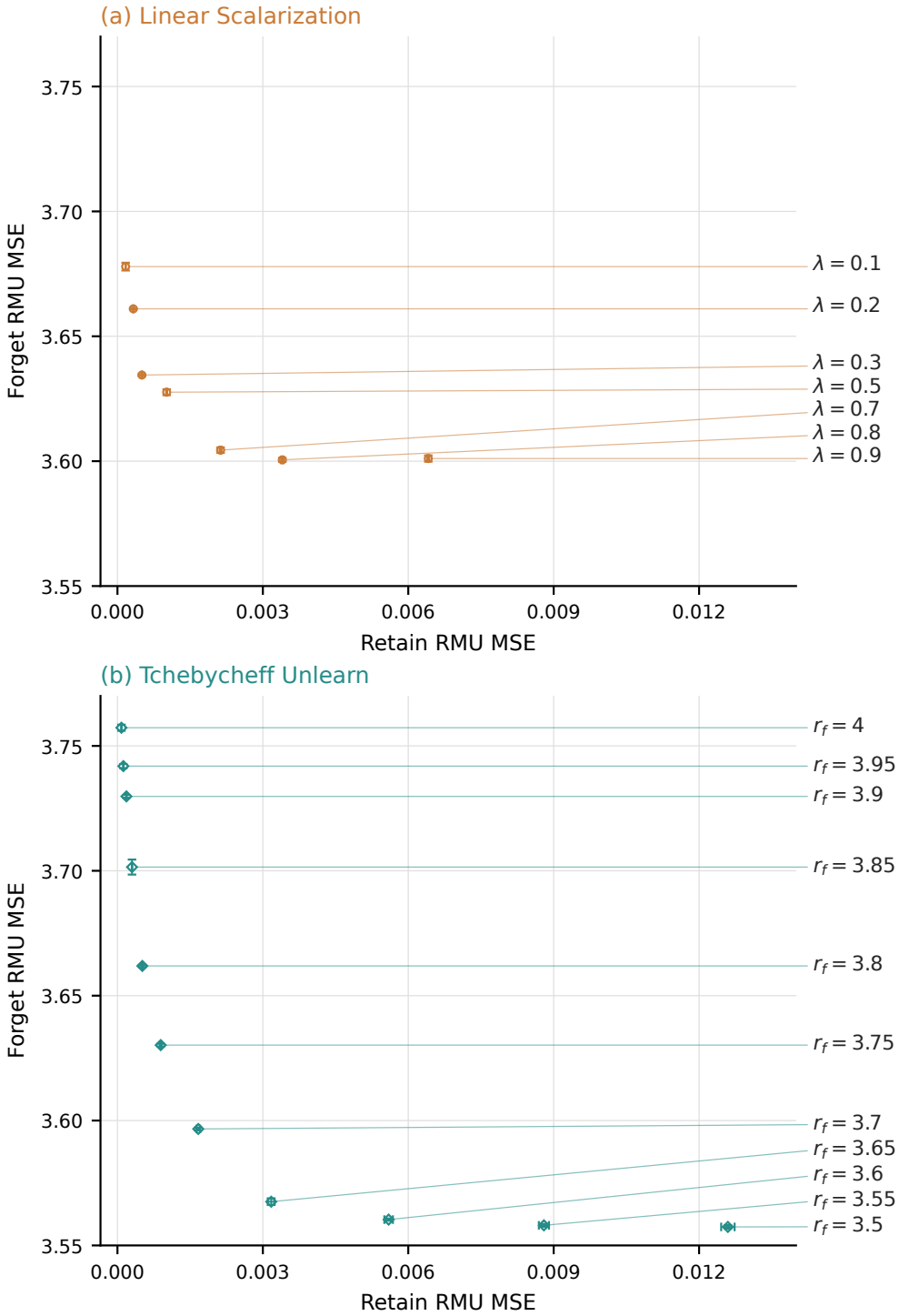

Figure 10: **Annotated MUSE-Books sweep at 200 updates.** Detailed version of Figure 5(b). Panels share axis limits; both loss axes use a linear scale. From top to bottom, labels list increasing λ for Linear Scalarization and decreasing r_f for Tchebycheff Unlearn.