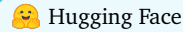


Towards Architectural-Level Safety Control: Relocating Hazardous Knowledge into Removable Experts in Mixture-of-Experts Models

Cheng Jin*, Zhongtian Ma*, Xiao Wang, Han Qi, Xiaoyu Wen, Xiang Wang, Youbang Sun, Xia Hu,
Bowen Zhou, Chaochao Lu^{†‡}, Qiaosheng Zhang^{‡#}

Shanghai AI Laboratory

* Equal Contribution. [†] Project Lead. [‡] Technical Lead. [#] Corresponding Authors.



Abstract | Large language models acquire broad knowledge from web-scale corpora during pretraining, including knowledge that could be hazardous if misused. Removing such knowledge from the model itself without degrading general capability is desirable but difficult, as it is deeply entangled with general capability within the same parameters. Existing safety alignment and unlearning methods typically adjust these shared parameters, tending to suppress the expression of hazardous knowledge rather than remove it. As a result, the knowledge remains in the model and is readily recoverable by subsequent fine-tuning. To address this limitation, we propose **REX** (Relocating into Removable **EX**perts), a knowledge relocation method that separates hazardous knowledge from general capability at the architectural level. Exploiting the modular structure of Mixture-of-Experts models, REX expands the model with a set of new, physically removable experts designated to hold hazardous knowledge. Through expert expansion, selective gradient masking and routing differentiation, it relocates hazardous knowledge out of the original experts and concentrates it in the new ones. Excising these new experts yields a model suitable for public release. This removal holds consistently across Intern-S2, Qwen3.5-35B-A3B and Gemma-4-26B-A4B; on Qwen3.5-35B-A3B, for instance, excision lowers WMDP-Bio accuracy by 48.4 points while MMLU changes by less than 3 points. The excised model matches the strongest of nine unlearning baselines in removing hazardous knowledge, and it is the only one that preserves this removal under subsequent fine-tuning. After 250M tokens of standard fine-tuning on Qwen3.5, the best baseline rebounds on WMDP-Bio from 44.1% to 78.3%, whereas REX rises only from 32.4% to 42.8%.

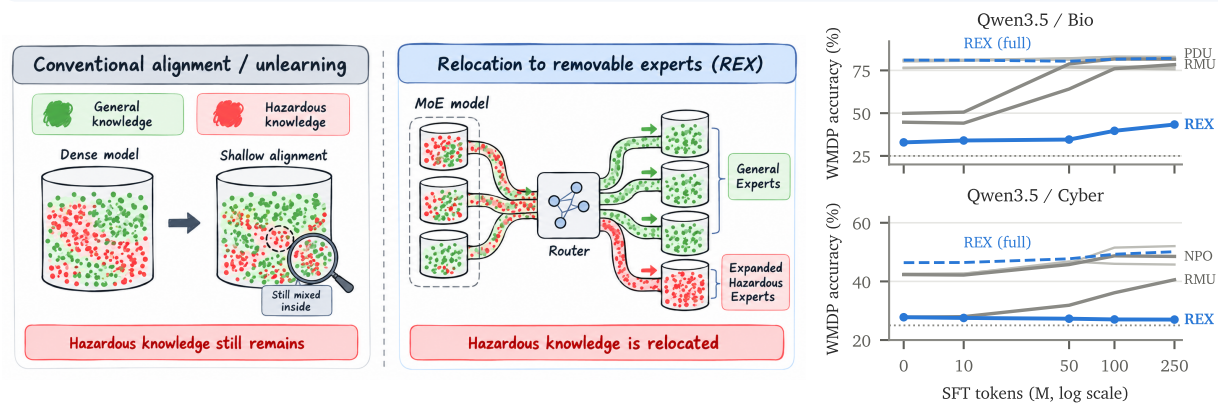


Figure 1 | Left: Conventional alignment and unlearning versus REX. In a dense model, general (green) and hazardous (red) knowledge are mixed, and alignment or unlearning changes only the surface behavior. REX routes hazardous knowledge into removable experts, which are excised before public release. Right: Qwen3.5 on WMDP-Bio and WMDP-Cyber during standard post-training (lower is better). Blue solid and dashed curves show the excised and full model. Gray curves show baselines.

Contents

1	Introduction	3
2	Related Work	4
3	Method	5
3.1	Preliminaries and notation	5
3.2	Expert expansion	6
3.3	Selective gradient masking	7
3.4	Routing differentiation	7
3.5	Excision and residual cleanup	8
3.6	Post-training and delivery	8
4	Experiments	9
4.1	Setup	9
4.2	Separating hazardous from general capability	10
4.3	Removing target knowledge from the base model	12
4.4	Persistence of hazardous knowledge removal under standard post-training	13
4.5	Ablations	14
5	Discussion	16
6	Conclusion	16
	References	17
A	Implementation details	21
A.1	Expansion and relocation	22
A.2	Residual cleanup	22
A.3	Standard post-training	22
A.4	Unlearning baselines	22
A.5	Evaluation	24
A.6	Routing diagnostic	25
B	Supplementary results	25
B.1	Separation at every stage	25
B.2	Random-expert excision	25
B.3	General capability	25

1. Introduction

Hazardous knowledge and general knowledge are deeply entangled in pretrained large language models. Pretraining inevitably teaches a model dual-use knowledge, such as synthesis routes in biology or attack techniques in cybersecurity, and this knowledge shares parameters with general knowledge, so the same weights encode legitimate academic competence and potentially dangerous capability. For open-weight models, external safeguards can be removed once the weights are released, so the hazardous knowledge itself should be addressed properly. An ideal removal approach should satisfy the following requirements:

1. Hazardous capability needs to be removed as thoroughly as possible, while doing minimal damage to general capability.
2. Removal should not come at the cost of destruction: developers still need to serve authorized research and defense users with the full capability, while releasing a version with controlled hazardous capability to the public.
3. The released model should remain safe after standard post-training, since open-weight models are routinely fine-tuned after release. Yet when existing methods only adjust original parameter values, standard supervised fine-tuning can restore or strengthen the removed knowledge, even on data unrelated to the hazardous domain [Doshi and Stickland, 2024].

Most existing approaches share a common limitation. They modify the original parameter set without changing the model architecture. They therefore fall short of the three requirements above in three related ways. First, behavioral interventions such as refusal training and inference-time filtering leave the hazardous knowledge encoded in the model unchanged and provide only shallow alignment [Qi et al., 2025a]. Unlearning methods such as Representation Misdirection for Unlearning (RMU) [Li et al., 2024], NPO [Zhang et al., 2024] and gradient ascent [Jang et al., 2023] act directly on the original parameters that also support general capability, so removing hazardous capability also affects general capability. Second, none of these approaches provides a single model architecture from which both a full model and a public model with a separately removable hazardous component can be delivered. Third, because current alignment and unlearning methods suppress or overwrite hazardous knowledge rather than removing it from the model, later post-training can make the same knowledge visible again (even when the new data is unrelated to the hazardous domain [Doshi and Stickland, 2024]). The released model therefore cannot be expected to remain safe after post-training. These limitations motivate a change at the architectural level. If we can augment a pretrained model with additional parameters, concentrate hazardous knowledge in them while preserving general capability in the original parameters, and remove these additional parameters before release, this could satisfy the three requirements above.

To realize this idea, we turn to Mixture-of-Experts (MoE) models [Shazeer et al., 2017; Fedus et al., 2022] and propose REX (Relocating into Removable EXperts), a method for relocating hazardous knowledge into newly added experts that can be removed before public release. MoE models are well suited to this approach because their feed-forward parameters are partitioned into independently addressable experts, the expert set can be expanded, a trainable router determines which experts process each token, and selected experts can be removed at delivery. REX expands a pretrained MoE model with dedicated target experts and relocates hazardous knowledge from the original experts into them, retaining general capability in the original experts and thereby separating hazardous capability from general capability at the architectural level.

The main challenge is to achieve this capability separation when hazardous and general knowledge are already mixed across the original experts. REX addresses this challenge through three components: expert expansion, selective gradient masking and routing differentiation. Expert expansion initializes

the target experts from the original experts that hazardous data relies on beyond general data, so relocation starts from experts already aligned with the target domain. Selective gradient masking lets target data update only the target experts, and on general data it excludes the target experts before top- K selection, so the original experts learn to maintain general capability without them while the hazardous knowledge they hold receives no training signal and fades. Routing differentiation directs target-related tokens toward the target experts and keeps general tokens away from them. After the target experts are excised, we use a light RMU to clean up the hazardous knowledge left in the retained parameters. Through these three components, REX separates hazardous capability from general capability at the architectural level. This separation allows general capability to improve during post-training without a substantial rebound in hazardous capability. Retaining or removing the target experts yields a full model for authorized users or a controlled model for public release.

An intuitive comparison is shown in Figure 1. The pretrained model can be viewed as a glass of water containing general knowledge (green) and hazardous knowledge (red) mixed together. Conventional alignment and unlearning operate directly inside this glass, trying to make the water appear green by suppressing hazardous outputs, while the underlying mixture remains unchanged. Our method uses the routing mechanism of an MoE model as an additional layer of separation. In this analogy, the router acts like a filter that directs the (red) hazardous signal to newly added target experts, while the (green) general signal remains in the original experts. As training continues on both data sources, routing and parameter updates reinforce this division. The model can then be separated into a full form that retains hazardous capability and an excised form with the target experts removed, providing a safer and more controllable model for release.

Our contributions are threefold.

- We propose REX, the first method to separate and remove hazardous capability in MoE models at the architectural level. REX moves the hazardous knowledge already present in a pretrained MoE model into expanded target experts while keeping general capability in the original experts. We measure this separation by comparing the full model with the excised model, obtained by removing the expanded experts, on hazardous and general benchmarks. In both domains and on three open-weight MoE models, the excised model scores substantially lower on hazardous capability while its general capability remains largely unchanged (Table 1).
- Although REX is designed for knowledge removal and capability separation rather than for benchmark forgetting alone, when applied to the base model it removes hazardous knowledge on the standard WMDP-Bio and WMDP-Cyber unlearning benchmarks on par with the strongest of 9 parameter-level unlearning baselines (Sections 4.3 and 4.5).
- We show that REX is the only method that preserves forgetting under post-training. In contrast to the baselines that reach a low starting point and then climb back towards the original model under the same standard fine-tuning, the delivered model of REX remains well below them in hazardous knowledge. On Qwen3.5, RMU rises from 44.1% to 78.3% on WMDP-Bio, close to the 83.2% of the original model, whereas REX ends at 42.8% and rises by only 10.5 points over the same trajectory (Section 4.4).

2. Related Work

Machine unlearning removes hazardous knowledge by changing parameter values in place, including in recent methods designed for Mixture-of-Experts models, and a growing body of work shows that such removal does not survive later training. RMU [Li et al., 2024] perturbs intermediate representations of forget data toward a random direction while preserving those of retain data, and it was introduced together with the WMDP benchmark that we use to measure hazardous knowledge.

Gradient ascent [Jang et al., 2023] maximizes the loss on forget data, gradient difference [Maini et al., 2024] adds a retain term to it, and NPO [Zhang et al., 2024] replaces its unbounded objective with a preference-style loss that avoids collapse. All of these methods act on parameters that general capability also uses. As a result, Doshi and Stickland [2024] show that ordinary fine-tuning on unrelated data restores the knowledge they remove, and Qi et al. [2025b] find that safeguards on open-weight models are routinely undone by later training. Tamper-resistant methods try to make the removal harder to reverse by shaping the loss landscape around the released weights, through adversarial meta-learning against simulated fine-tuning attacks in TAR [Tamirisa et al., 2025], an invariance objective in ILU [Wang et al., 2025a], or sharpness-aware minimization [Fan et al., 2025a], yet the suppressed knowledge still resides in parameters that remain in the model. For MoE models, recent work uses the expert structure to make unlearning more targeted. SEUF [Zhuang et al., 2024] attributes the target knowledge to the experts most engaged with it, confines unlearning to those experts and anchors the router so that they stay selected. Such methods use experts to decide where to unlearn, but they still overwrite parameter values inside experts that the delivered model keeps.

Knowledge localization takes a different approach and confines a capability to designated parameters so that it can be removed as a whole. Gradient routing [Cloud et al., 2024] masks gradients during training so that chosen data updates only designated subregions of a network, which can then be ablated. SGTm [Shilov et al., 2025] builds on it and applies selective gradient masking during pretraining so that target data updates only a designated subset of parameters, which are then removed, and GRAM [Roland et al., 2026] uses a modular architecture to enable access control over capabilities learned during pretraining. Because these methods prevent knowledge from entering the rest of the model while it is first learned, they require training from scratch and do not apply to a pretrained model in which the knowledge is already spread across the parameters.

3. Method

REX starts from a pretrained MoE model B and relocates the hazardous knowledge spread across its original experts into a set of new, physically removable experts, which we call target experts. The pipeline has three training stages and is illustrated in Figure 2. The first stage expands B into F_0 with the target experts and relocates the hazardous knowledge into them. Three components perform the relocation, namely expert expansion (Section 3.2), selective gradient masking (Section 3.3) and routing differentiation (Section 3.4). They turn the catastrophic forgetting of continued training into directed forgetting, and together they realize knowledge relocation and capability separation in the full model F_1 . The second stage excises the target experts of F_1 and their router rows, which we keep as H_1 , and applies a light RMU cleanup, which turns the excised model A_1 into the delivered base model A_2 (Section 3.5). The third stage recombines A_2 with H_1 into the full structure F_2 , runs standard post-training on F_2 to obtain the full model for authorized users F_3 , and excises the target experts at delivery to obtain the public model A_3 , which bounds the growth of hazardous knowledge during post-training (Section 3.6). We first fix the notation (Section 3.1).

3.1. Preliminaries and notation

We use two kinds of training data. Target data $\mathcal{D}_{\text{harm}}$ carries the target knowledge, such as a biosecurity or cybersecurity corpus, and general data $\mathcal{D}_{\text{std}}$ covers general capability. In an MoE layer, the router maps the hidden state x of a token to expert logits and routing probabilities

$$r(x) = W_r x, \quad P(x) = \text{softmax}(r(x)). \quad (1)$$

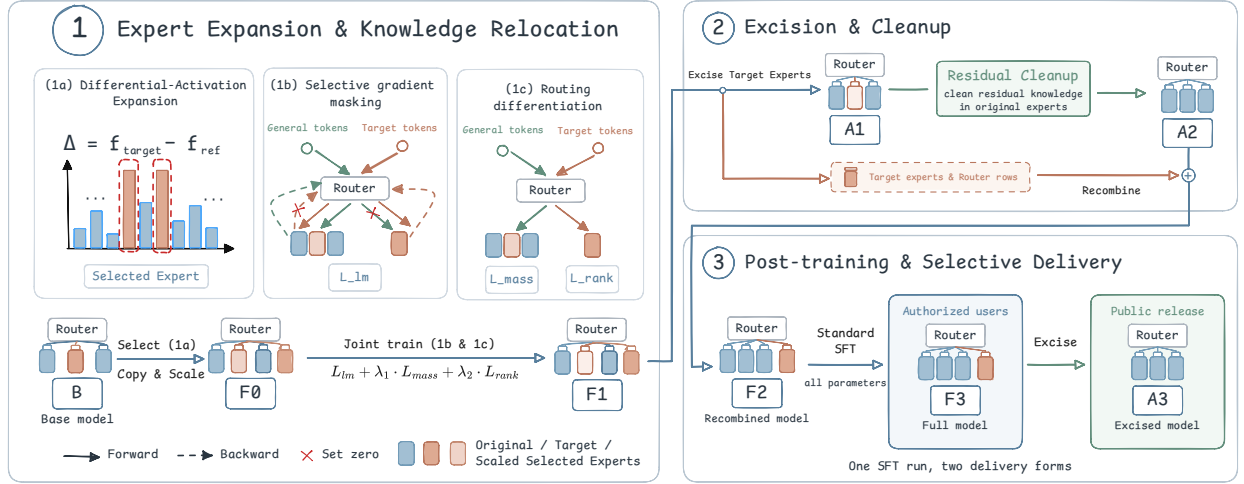


Figure 2 | Overview of the REX pipeline. The method first expands the pretrained MoE model and relocates hazardous knowledge into newly added experts through expert expansion, selective gradient masking and routing differentiation. A light RMU cleanup then cleans residual hazardous knowledge from the original parameters. Post-training runs on the recombined full model, and the target experts are excised at delivery to obtain the public model.

The layer output is the weighted sum of the top- K experts,

$$y = \sum_{i \in \text{TopK}(r(x))} g_i(x) E_i(x), \quad (2)$$

where $\text{TopK}(r(x))$ is the set of the K experts with the largest logits, E_i is the i -th expert and g_i are the normalized gate weights. We write θ_{harm} for the target experts and their router rows, θ_{std} for the original experts and their router rows, and θ_{shared} for the remaining parameters, including attention, embeddings, norms, the language-model head and any router parameters not tied to a single expert. Shared-expert MLPs belong to neither group of experts.

3.2. Expert expansion

Expert expansion adds the target experts and initializes them by differential-activation expansion. On the training corpora $\mathcal{D}_{\text{std}}$ and $\mathcal{D}_{\text{harm}}$, we count how often each expert i is selected into the top- K , which gives the frequencies f_i^{std} and f_i^{harm} , and compute $\Delta_i = f_i^{\text{harm}} - f_i^{\text{std}}$. In each layer, the M experts with the largest Δ_i form the source set S , where M is the number of target experts per layer. Selecting by the frequency on target data alone tends to pick general-purpose experts that are active on all data, and their copies would carry nothing specific to the target domain. Subtracting the frequency on general data keeps the experts that target data relies on beyond general data, so the target experts match the computation of the target domain from the start. The selection uses only routing statistics on the training data and never reads evaluation data.

For each expert in S , we copy its parameters and router row as a new target expert. We write H for the set of new target experts. Expanding the original model B in this way gives the expanded model F_0 . All original experts are kept, and the source experts are scaled by a fixed factor s to weaken the hazardous knowledge they hold before training. If the source experts were left unchanged, they would compete with the target experts for target tokens, making routing differentiation harder and greatly increasing the training budget that relocation needs. If they were set to zero, general capability would

suffer severe damage. Scaling weakens the hazardous knowledge in the source experts while keeping most of their general computation.

3.3. Selective gradient masking

Selective gradient masking decides where the updates of each data source are allowed to be written. During relocation the shared parameters stay frozen, including attention, embeddings, norms and the language-model head, together with the shared-expert MLPs, and only the routed experts and the routers are trained. For the expert parameters, gradients from target tokens update only the target experts, and gradients from general tokens update only the original experts. The routers are updated by both. In terms of parameter sets,

$$\text{updates} = \begin{cases} \theta_{\text{harm}} \cup \theta_{\text{router}} & \text{on } \mathcal{D}_{\text{harm}}, \\ \theta_{\text{std}} \cup \theta_{\text{router}} & \text{on } \mathcal{D}_{\text{std}}, \end{cases} \quad (3)$$

where θ_{router} collects all router rows. Forward computation is masked on general data only. A pre-top- K activation mask sets the logits of the target experts to $-\infty$ before the selection,

$$\tilde{r}_i(x) = \begin{cases} r_i(x) & i \notin H, \\ -\infty & i \in H, \end{cases} \quad x \in \mathcal{D}_{\text{std}}, \quad (4)$$

so the top- K is taken over the original experts and the gate weights are renormalized. Each token of general data is therefore dispatched without the target experts, and the model must perform well with the original experts alone.

The two rules reinforce each other. The gradient mask on target data makes the target experts the only place where new hazardous knowledge is written. The pre-top- K activation mask on general data trains the original experts to hold general capability without the target experts, so the hazardous knowledge they already carry loses its support and decays.

3.4. Routing differentiation

Routing differentiation steers target tokens toward the target experts, and together with selective gradient masking it accelerates capability separation. It adds two losses. The mass loss decides how much routing mass the target experts receive, and the rank loss decides where they sit in the top- K order. For token t in a given layer, let $z_t = \sum_{i \in H \cap \text{TopK}(r(x_t))} g_i(x_t)$ be the normalized top- K gate weight assigned to the target experts. z_t is computed before masking, while the actual dispatch uses the top- K selected again after masking. Writing $\bar{z}$ for the mean of z_t over the tokens of a batch and over MoE layers, the mass loss requires $\bar{z}$ to be no lower than a floor τ_h on target data and no higher than a ceiling τ_s on general data,

$$\mathcal{L}_{\text{mass}} = \begin{cases} \lambda_h \max(0, \tau_h - \bar{z}) & \text{on } \mathcal{D}_{\text{harm}}, \\ \lambda_s \max(0, \bar{z} - \tau_s) & \text{on } \mathcal{D}_{\text{std}} \end{cases}, \quad (5)$$

so it raises the mass on target tokens and holds it down on general tokens. The hinge form produces gradients only when routing violates this split.

The rank loss constrains the order of the top- K selection instead of the mass. A bound on the mass does not decide which experts are activated, because a target expert enters the activated set as soon as its logit crosses the top- K boundary, even when it then receives little mass. On $\mathcal{D}_{\text{std}}$ we therefore

write $h_{\max} = \max_{i \in H} r_i(x_t)$ for the largest target-expert logit and s_K for the K -th largest logit among the original experts, which is the top- K boundary, and require $h_{\max}$ to stay below s_K by a margin δ ,

$$\mathcal{L}_{\text{rank}} = \mathbb{E}_{t \in \mathcal{D}_{\text{std}}} [\max(0, \delta - (s_K - h_{\max}))], \quad (6)$$

so the target experts stay outside the activated set on general tokens. The pre-top- K activation mask keeps the target experts out of dispatch on general data only during relocation. At inference the router scores all experts, so the rank loss trains the unmasked logits themselves to keep the target experts below the top- K boundary on general tokens.

The two losses complement each other. The mass loss shapes how much routing mass the target experts receive, but it bounds only the total mass, so a satisfied bound does not prevent an expert from being wrongly selected. The rank loss acts on the top- K selection itself and keeps the target experts out of the activated set on general tokens. Because it constrains the selection so directly, however, its gradient can pull against the language-modeling loss. We therefore use the two losses together, with the rank loss correcting the selection and the mass loss shaping routing more gently. Thus, the final relocation objective for the first stage of REX, which turns F_0 into the full model F_1 , is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{mass}} + \lambda_r \mathcal{L}_{\text{rank}}, \quad (7)$$

where the language-modeling loss is computed and back-propagated under the masking rules of Section 3.3.

3.5. Excision and residual cleanup

Not all hazardous knowledge is held in the target experts. Part of it stays in the original experts, where directed forgetting is incomplete, and part of it is encoded in parameters that relocation never updates, such as attention and the shared-expert MLPs. Excision cannot remove this part, so the second stage of REX adds a parameter-level cleanup after the structural removal.

The second stage has two steps. It first excises the target experts of F_1 and their router rows, which we keep as H_1 for the third stage, and obtains the model A_1 . The residual cleanup then applies RMU [Li et al., 2024] to the expert MLPs that remain in A_1 . At a chosen monitor layer, the objective pushes the hidden states of target data toward a fixed random direction and keeps the hidden states of general data close to those of A_1 before cleanup,

$$\mathcal{L}_{\text{RMU}} = \mathbb{E}_{t \sim \mathcal{D}_{\text{harm}}} \|\phi_{\theta}(t) - cu\|_2^2 + \alpha \mathbb{E}_{t \sim \mathcal{D}_{\text{std}}} \|\phi_{\theta}(t) - \phi_{\theta_{\text{ref}}}(t)\|_2^2, \quad (8)$$

where $\phi_{\theta}(t)$ is the hidden state of token t at the monitor layer, u a fixed random unit vector, c the steering coefficient, α the weight of the retain term, and θ_{ref} the parameters of A_1 before cleanup. The trainable parameters are the routed-expert MLPs and the shared-expert MLPs of a few early layers, while the routers, attention, embeddings, norms and the language-model head stay frozen. The cleanup gives the delivered base model A_2 , which the third stage recombines with H_1 into F_2 .

3.6. Post-training and delivery

The third stage takes the full structure F_2 built in the previous stage and runs ordinary full-parameter fine-tuning S_{std} on it, which gives the full instruction model F_3 . Hazardous knowledge does not rebound during post-training because the router and the target experts work together. The router keeps sending target-related tokens to the target experts, so the updates those tokens produce are written mainly into the target experts, while the growth in general capability is written into the parameters that stay.

At delivery we excise the target experts, and the hazardous capability leaves with them, which gives the public model A_3 . The delivered model therefore improves on general capability without a matching rise in hazardous capability. The releasing party holds the target experts, so it can decide which artifact to expose. F_3 serves authorized users and keeps the hazardous capability, while A_3 is the public model without it.

4. Experiments

Our experiments follow the order of the three contributions. Section 4.2 measures the capability separation that REX builds, comparing the full model with the excised model both directly after REX and after standard post-training. Section 4.3 compares the excised base model with nine parameter-level unlearning baselines on WMDP before any post-training, and Section 4.4 compares the same routes after one shared standard fine-tuning procedure. Section 4.5 attributes each result to the step of the pipeline that produces it.

4.1. Setup

We study three MoE backbones with different routing designs and two hazardous domains. Intern-S2 [Bai et al., 2026] and Qwen3.5-35B-A3B [Qwen Team, 2026] have 40 layers with 256 routed experts, top-8 routing and a shared expert. Gemma-4-26B-A4B [Gemma Team, 2026] has 30 layers with 128 routed experts, top-8 routing, per-expert router scales and an always-on dense MLP. The target domains are biosecurity and cybersecurity, with the WMDP forget corpora [Li et al., 2024] as $\mathcal{D}_{\text{harm}}$ and FineWeb-Edu [Penedo et al., 2024] as $\mathcal{D}_{\text{std}}$. Expansion adds $M = 8$ target experts to every MoE layer, and the full recipe of relocation and residual cleanup is given in Section A. Model names follow Section 3. B is the original model and F_0 its expansion, F_1 and A_1 are the full and excised models after relocation, A_2 is the delivered base model after residual cleanup and F_2 its recombination with the target experts, and post-training turns F_2 into the full model for authorized users F_3 and, after excision, the public model A_3 .

We compare against nine parameter-level unlearning baselines: RMU [Li et al., 2024], gradient ascent (GA) [Jang et al., 2023], gradient difference (GD) [Maini et al., 2024], NPO [Zhang et al., 2024], SimNPO [Fan et al., 2025b], CEU [Yang, 2025], PDU [Entesari et al., 2025], SatImp [Yang et al., 2025] and WGA [Wang et al., 2025b]. Every baseline is trained on the same forget and retain data as REX, tuned with a hyperparameter search that follows its original paper (Section A.4) and evaluated in its standard form without excision, so RMU in the tables denotes this external baseline rather than the matched control R of Section 4.5. A method has collapsed if it scores below half of the original model on any of the four general-capability benchmarks, or, along the post-training trajectory, more than 10 MMLU points below the original model. Collapsed methods are still reported but take no part in the comparisons of Sections 4.3 and 4.4, and since no baseline usable on the base model collapses during post-training, the comparison set is the same at both stages.

Post-training uses one fixed standard fine-tuning procedure S_{std} for every route (Section A.3). It is full-parameter supervised fine-tuning on a text-only instruction mixture with the life-science subsets removed, run along one nominal 250M-token cosine trajectory at learning rate 2×10^{-6} with the Muon optimizer [Jordan et al., 2024] and the same data order and seed, and WMDP and MMLU are evaluated at 0, 10M, 50M, 100M and 250M tokens. Hazardous knowledge is accuracy on WMDP-Bio or WMDP-Cyber [Li et al., 2024], a four-way multiple-choice test with chance accuracy at 25%. General capability is measured by MMLU [Hendrycks et al., 2021], ARC-C [Clark et al., 2018], HellaSwag [Zellers et al., 2019] and GSM8K [Cobbe et al., 2021], and instruction following by IFEval prompt-level strict accuracy [Zhou et al., 2023], all run with lm-eval [Biderman et al., 2024] and

vLLM [Kwon et al., 2023] under the protocol of Section A.5. Along the trajectory, the excised side of REX is evaluated with the pre-top- K activation mask on the target experts, which agrees with physical excision within one standard error where both were measured (Table 9).

4.2. Separating hazardous from general capability

Excising the target experts takes away the target knowledge and leaves general capability almost unchanged, so the same model takes two distinct capability profiles depending on whether these experts are kept (Table 1). For every backbone and domain, Table 1 compares F_2 with A_2 at the base-model stage and F_3 with A_3 after standard fine-tuning. Within each pair the two models share every parameter except the target experts, and Table 10 lists the pair at every stage of the pipeline. At the base-model stage, excision lowers WMDP by 9.5–48.4 points, while across the 48 general-capability comparisons of the two stages the median change is 0.6 points and 34 of them stay below one point. The pattern holds after post-training as well. After 250M tokens of full-parameter fine-tuning, the target experts still carry the target knowledge, and excising them removes 11.1–38.9 points of WMDP at the same mild cost to general capability.

The full model retains the hazardous capability, so the removal performed by REX is separation rather than destruction. After post-training, the full model for authorized users F_3 is 2.3–6.0 points of WMDP below the original model fine-tuned with the same procedure, $S_{\text{std}}(B)$, across the six settings, while A_3 , excised from the same run, is the public model. The base-model pair gives more direct evidence. The parameters of F_2 outside the target experts have already been cleaned and are identical to those of A_2 , yet adding back the target experts alone raises WMDP by up to 48.4 points, from 32.4% to 80.8% on Qwen3.5 / Bio. The target knowledge is therefore stored in these experts rather than wiped out beyond recovery, and both forms of the model come from one fine-tuning run.

The drop in target knowledge comes from the target experts, not from excising any parameters of the same size (Table 11). On the Intern-S2 / Bio model F_1 , we remove the same number of original experts as there are target experts in every layer, keep the added experts, and repeat with five random seeds. After random excision WMDP-Bio is $78.9 \pm 0.9\%$, essentially the 79.4% of F_1 , whereas excising the target experts lowers it to 64.3%. MMLU stays at $82.4 \pm 0.2\%$ and 82.3% under the two excisions. Removing structure of the same size does not by itself take the target knowledge away. What takes it away is the set of experts that REX selected and relocated.

Relocation makes the target experts the inference path for target tokens, while general tokens stay on the original experts (Figure 3). On target tokens, the routing mass assigned to the target experts rises in all six settings, from 17–27% to 65–76% on Bio and from 15–22% to 35–47% on Cyber, starting from a share inherited from the source experts that expansion copies. On Bio the target experts hold more than half of the selected mass at most layers of all three backbones, and on Cyber, whose smaller forget corpus gives a shorter relocation budget, the share approaches or passes half in the last layers. On general tokens the mass stays far below this level in every setting and is highest on Intern-S2 / Bio at 20.2%, where excision still lowers MMLU only from 83.2% to 82.3%. Because target tokens take the target experts at inference time, excision carries the target knowledge away, and post-training gradients on target-related data flow into these experts (Section 4.4).

Table 1 | Capability separation in the same model (%). Full is the model with the target experts and Excised the same model with them removed; Δ excision (shaded row) is Full minus Excised, i.e. the accuracy taken away by the excision, with the target-knowledge entries in blue. Grey text marks the references: the original model B for the base model, and the same standard SFT (250M tokens) applied to it, $S_{\text{std}}(B)$, after SFT.

		Qwen3.5					Intern-S2					Gemma 4				
		WMDP↓	MMLU	ARC-C	Hella.	GSM8K	WMDP↓	MMLU	ARC-C	Hella.	GSM8K	WMDP↓	MMLU	ARC-C	Hella.	GSM8K
Bio																
Base	Original B	83.2	82.4	73.5	85.6	85.8	83.4	85.0	74.3	85.8	87.1	78.9	74.3	73.2	85.6	71.4
	Full F_2	80.8	80.1	70.1	83.7	83.4	61.9	82.4	70.7	84.8	84.8	71.0	72.6	71.0	85.0	69.1
	Excised A_2	32.4	77.2	68.8	83.6	83.3	29.6	81.0	70.7	84.9	83.8	33.6	68.7	66.7	84.6	66.0
	Δ excision	48.4	2.9	1.4	0.1	0.1	32.3	1.4	0.0	-0.1	1.0	37.4	3.9	4.3	0.4	3.1
SFT	$S_{\text{std}}(B)$	84.0	82.2	73.6	85.4	85.1	84.5	84.8	74.4	85.8	86.9	78.8	74.9	73.7	85.5	72.2
	Full F_3	81.7	80.7	70.7	84.2	85.9	78.6	84.1	72.0	85.3	85.2	76.4	73.9	71.8	85.2	69.2
	Excised A_3	42.8	78.7	70.3	84.1	85.7	55.0	82.5	71.3	85.1	86.2	41.3	70.5	67.9	84.8	68.2
	Δ excision	38.9	2.0	0.4	0.1	0.2	23.6	1.5	0.7	0.1	-1.0	35.1	3.4	3.9	0.4	1.1
CYBER																
Base	Original B	57.6	82.4	73.5	85.6	85.8	58.9	85.0	74.3	85.8	87.1	50.8	74.3	73.2	85.6	71.4
	Full F_2	46.5	81.4	72.9	82.9	84.4	52.8	84.4	73.3	82.3	84.0	38.9	72.5	72.6	84.6	63.1
	Excised A_2	28.2	80.7	72.3	82.4	83.2	26.3	84.1	73.6	81.7	83.9	29.4	71.9	72.0	83.9	53.6
	Δ excision	18.4	0.7	0.6	0.4	1.1	26.6	0.3	-0.3	0.7	0.1	9.5	0.6	0.6	0.7	9.5
SFT	$S_{\text{std}}(B)$	55.3	82.2	73.6	85.4	85.1	59.1	84.8	74.4	85.8	86.9	51.1	74.9	73.7	85.5	72.2
	Full F_3	50.3	82.0	72.5	84.3	85.6	55.8	84.0	73.7	84.7	86.5	45.1	74.0	72.8	85.4	72.1
	Excised A_3	27.3	81.3	73.0	83.9	85.4	29.8	84.0	73.7	84.3	85.6	34.0	73.3	72.7	85.2	69.9
	Δ excision	23.0	0.8	-0.4	0.4	0.2	26.0	0.0	0.0	0.4	0.9	11.1	0.6	0.1	0.2	2.2

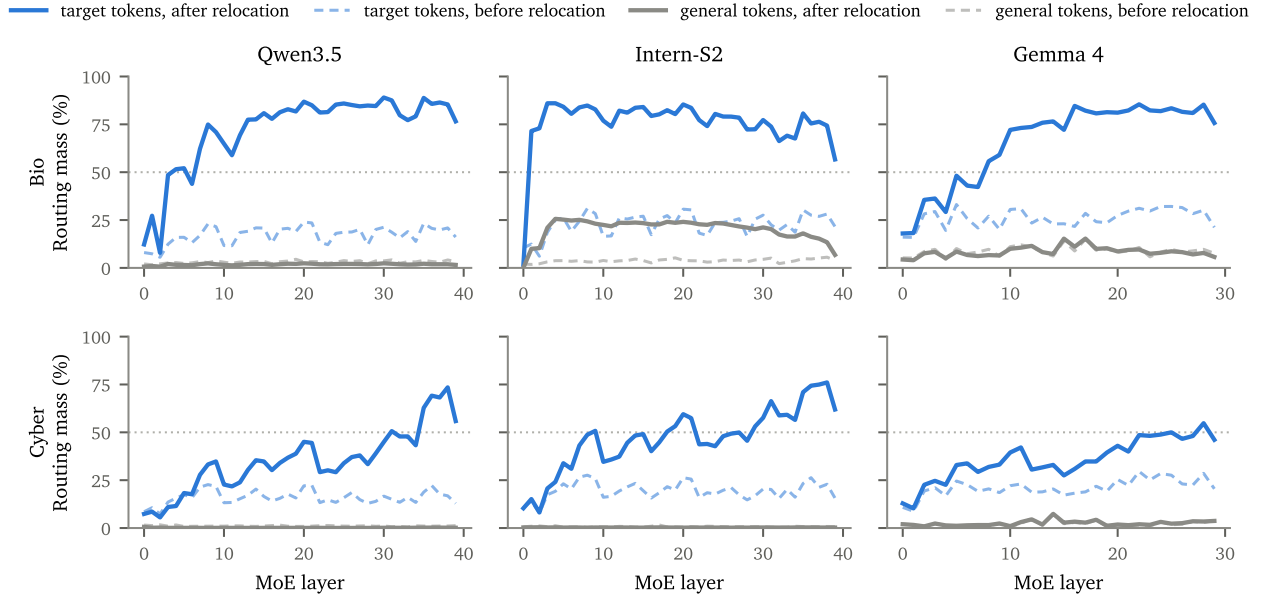


Figure 3 | Routing mass on the target experts at every MoE layer. Each curve is the mean normalized top- K mass that valid tokens assign to the target experts, so values above the dashed 50% line mean the target experts take most of the selected mass. Solid lines are after relocation (F_1) and faint dashed lines before it (F_0). Target and general tokens use separate fixed samples shared between F_0 and F_1 , and all panels share one vertical axis.

Table 2 | Base-model unlearning before post-training (%). REX (shaded row) is the delivered base A_2 . Deeper blue marks a larger share of the above-chance WMDP accuracy of B removed, $(W_B - W)/(W_B - 25)$; general capability is a constraint and is not ranked. A dagger marks a collapsed method, with a score below 50% of B on any general-capability benchmark; it is muted and excluded from the comparison.

	Qwen3.5					Intern-S2					Gemma 4				
	WMDP↓	MMLU	ARC-C	Hella.	GSM8K	WMDP↓	MMLU	ARC-C	Hella.	GSM8K	WMDP↓	MMLU	ARC-C	Hella.	GSM8K
Bio															
Original (B)	83.2	82.4	73.5	85.6	85.8	83.4	85.0	74.3	85.8	87.1	78.9	74.3	73.2	85.6	71.4
REX (ours)	32.4	77.2	68.8	83.6	83.3	29.6	81.0	70.7	84.9	83.8	33.6	68.7	66.7	84.6	66.0
CEU	81.2†	81.0†	40.1†	39.3†	3.8†	70.2†	75.7†	29.0†	30.6†	0.0†	75.1†	72.0†	34.5†	33.5†	0.0†
GA	25.0†	28.1†	26.0†	26.1†	0.0†	24.7†	23.0†	25.6†	25.5†	0.0†	24.7†	22.9†	27.2†	26.6†	0.0†
GD	81.0	81.2	72.0	85.2	74.7	66.9	82.5	73.5	85.0	85.1	28.6	70.7	65.5	79.2	40.5
NPO	76.4	77.4	71.5	83.4	83.6	76.9	78.7	67.4	83.2	82.2	65.0	69.2	68.4	84.1	57.1
PDU	47.8	78.2	72.5	84.7	67.7	54.8	83.8	75.3	86.2	87.1	69.4	73.3	71.9	84.4	68.5
RMU	44.1	82.4	73.9	85.4	84.7	37.4	84.3	74.6	85.4	86.1	29.2	70.0	68.0	78.6	68.4
SatImp	80.4	80.6	43.3	44.1	70.3	73.8†	77.6†	35.1†	38.0†	0.3†	61.6†	57.1†	35.1†	33.4†	0.0†
SimNPO	76.4	77.2	71.0	83.7	82.6	74.6	77.8	69.4	83.5	81.5	64.3	68.4	69.2	84.1	58.5
WGA	80.9	80.7	43.5	44.2	71.2	74.0†	76.9†	35.0†	37.0†	0.5†	66.3†	61.0†	35.5†	34.3†	0.0†
CYBER															
Original (B)	57.6	82.4	73.5	85.6	85.8	58.9	85.0	74.3	85.8	87.1	50.8	74.3	73.2	85.6	71.4
REX (ours)	28.2	80.7	72.3	82.4	83.2	26.3	84.1	73.6	81.7	83.9	29.4	71.9	72.0	83.9	53.6
CEU	37.8†	75.4†	35.5†	33.4†	0.0†	44.8†	77.8†	34.1†	36.0†	0.0†	26.6†	32.3†	30.4†	30.1†	0.0†
GA	34.7†	74.4†	30.4†	26.9†	0.0†	24.6†	35.5†	26.3†	26.7†	0.0†	43.6†	72.4†	28.2†	29.0†	0.0†
GD	36.9†	78.6†	74.7†	84.3†	0.0†	31.0†	81.8†	74.3†	84.2†	0.4†	27.3†	66.6†	60.2†	73.9†	0.0†
NPO	41.9	81.3	74.2	87.6	85.3	41.4	83.2	72.3	86.5	87.6	44.1	70.9	67.9	84.1	56.2
PDU	42.4	81.9	73.2	85.4	76.0	27.8	84.7	74.5	86.4	85.8	25.1	73.7	71.3	84.5	69.8
RMU	27.5	81.8	72.6	84.9	76.9	35.7	84.8	74.3	83.4	85.0	26.8	72.6	67.9	78.7	65.4
SatImp	32.8†	74.4†	39.1†	37.2†	0.0†	37.6†	78.8†	48.5†	54.2†	0.0†	28.2†	40.4†	30.5†	34.1†	0.0†
SimNPO	42.5	81.1	74.9	88.5	74.3	47.5	81.6	72.7	82.4	86.0	26.8	67.9	66.1	82.2	48.7
WGA	32.2†	72.9†	37.5†	35.7†	0.0†	37.6†	78.7†	48.4†	53.4†	0.0†	26.3†	34.1†	27.5†	30.6†	0.0†

4.3. Removing target knowledge from the base model

Before any post-training, the base model A_2 delivered by REX removes target knowledge to a degree comparable with the strongest external parameter-level unlearning methods (Table 2). Once collapsed methods are set aside, the usable baselines in every setting still span at least 15 points of WMDP accuracy. Most of them leave target accuracy far above chance, and only a few reach the low region. A_2 lies in this low region in all six settings and is never more than 5.0 points above the lowest usable baseline. In three settings A_2 is itself the lowest, by 11.6 points below RMU on Qwen3.5 / Bio, 7.8 points below RMU on Intern-S2 / Bio and 1.6 points below PDU on Intern-S2 / Cyber, and on Qwen3.5 / Cyber it is within 0.6 points of RMU. The strongest baseline is not the same method across settings, alternating among RMU, GD and PDU, whereas A_2 reaches this group in every setting with one pipeline. This level is reached with a light RMU cleanup because excision has already taken away the knowledge carried by the target experts, so the cleanup acts only on what remains in the retained parameters. Section 4.5 shows that the same cleanup applied to the original model leaves more target knowledge in every setting. On Gemma 4 / Bio, GD and RMU start below A_2 at the base-model stage. Section 4.4 shows that this starting point does not hold under the same standard post-training.

This removal does not come at the expense of general capability, which A_2 retains at the same level as the usable external methods. In all six settings, the MMLU, ARC-C and HellaSwag scores of A_2 stay above 91% of those of the original model B , the same level of retention as usable methods such as RMU and PDU (Table 2). The more visible MMLU decrease on Bio is concentrated in the subjects adjacent to the target domain: the ten biology and health subjects make up 15% of MMLU questions but account for 57–75% of the decrease, while the remaining subjects stay within 1.2–2.8 points

Table 3 | Hazardous knowledge under the same standard post-training (%). Every route runs the same 250M-token S_{std} ; Before and After give WMDP at its input and output, and MMLU is measured after. REX (shaded row) is the delivered model A_3 , fine-tuned as the recombined full model and excised at delivery. Deeper blue marks a larger share of the above-chance WMDP accuracy of B still removed after fine-tuning. Only baselines that did not collapse on the base model (Table 2) are reported; a dash marks a collapsed setting.

	Qwen3.5			Intern-S2			Gemma 4		
	Before	After↓	MMLU	Before	After↓	MMLU	Before	After↓	MMLU
Bio									
Original + S_{std}	83.2	84.0	82.2	83.4	84.5	84.8	78.9	78.8	74.9
REX (ours)	32.4	42.8	78.7	29.6	55.0	82.5	33.6	41.3	70.5
GD	81.0	82.2	81.5	66.9	83.0	84.4	28.6	77.4	74.8
NPO	76.4	76.0	78.3	76.9	77.6	80.7	65.0	67.8	70.3
PDU	47.8	81.8	81.4	54.8	82.7	84.8	69.4	78.5	74.7
RMU	44.1	78.3	82.0	37.4	82.0	84.6	29.2	60.8	73.2
SatImp	80.4	82.7	82.1	–	–	–	–	–	–
SimNPO	76.4	77.5	78.3	74.6	75.6	79.9	64.3	66.9	69.0
WGA	80.9	82.7	82.0	–	–	–	–	–	–
CYBER									
Original + S_{std}	57.6	55.3	82.2	58.9	59.1	84.8	50.8	51.1	74.9
REX (ours)	28.2	27.3	81.3	26.3	29.8	84.0	29.4	34.0	73.3
NPO	41.9	49.4	81.9	41.4	50.6	84.5	44.1	46.5	72.6
PDU	42.4	45.5	82.3	27.8	40.1	84.8	25.1	35.5	74.5
RMU	27.5	40.6	82.2	35.7	51.2	84.3	26.8	35.7	73.8
SimNPO	42.5	52.5	81.9	47.5	52.1	82.8	26.8	38.0	71.4

of B (Table 13). What leaves together with the target domain is therefore mainly knowledge in the adjacent subjects rather than general capability at large. On Gemma 4 / Bio, RMU, which removes target knowledge to a similar depth, falls to the same level on these subjects (60.4% against 59.8% for A_2). Section 4.4 further shows that standard post-training raises the MMLU of the delivered model.

4.4. Persistence of hazardous knowledge removal under standard post-training

Under the same standard fine-tuning, the model delivered by REX retains the least target knowledge among all methods that are usable on the base model (Table 3 and Figure 4). After 250M tokens, A_3 has the lowest WMDP in 6 of 6 settings, 19.5–33.2 points below the lowest baseline on Bio and 1.5–13.3 points below it on Cyber. The baselines that matched A_2 in Section 4.3 return close to the fine-tuned original model. On Intern-S2 / Bio, RMU climbs from 37.4% to 82.0%, against 84.5% for the original model under the same S_{std} , while A_3 ends at 55.0%. On Gemma 4 / Bio, GD and RMU start below A_2 (28.6% and 29.2% against 33.6%) yet end at 77.4% and 60.8%, above the 41.3% of A_3 . A low starting point before post-training therefore does not bound what post-training restores.

The advantage of A_3 is not inherited from a lower starting point. It keeps most of its base-model removal, whereas the baselines lose most of theirs. We define the kept share of a route as $(W_B - W_{250M}) / (W_B - W_0)$, where W_B , W_0 and W_{250M} are the WMDP accuracy of B , of the route before post-training and of the route after 250M tokens, so it measures how much of the removal before post-training is still present. On Bio, A_3 keeps at least 52%, while the two strongest baselines of each setting keep at most 37%. On Cyber the figures are at least 78% against at most 63%. The removal of every baseline resides in parameter values, and standard fine-tuning modifies those same values. REX instead fine-tunes the recombined full model. The router sends target-related tokens to the target experts, the fine-tuning gradients related to target knowledge follow them there, and the resulting updates leave with those experts at delivery. F_3 from the same run serves as the full

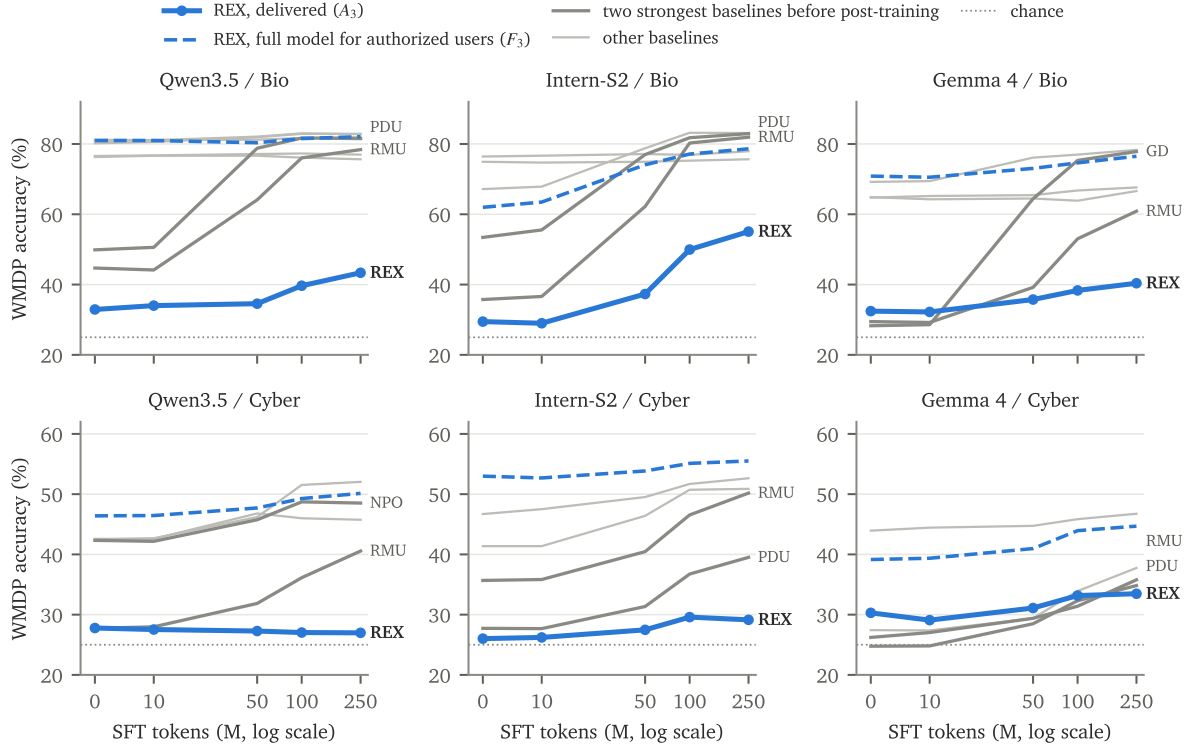


Figure 4 | Target knowledge along one shared 250M-token standard SFT trajectory (same data, order and seed for every route, with the 0 point the input checkpoint). Blue curves are REX, solid for the delivered model A_3 and dashed for the full model for authorized users F_3 from the same run. Grey curves are the external unlearning baselines that did not collapse on the base model, with the two lowest in WMDP before post-training drawn darker and labelled.

model for authorized users. The direct-SFT control in Section 4.5 starts from the same A_2 without recombination and ends 9.6–20.2 points higher on Bio, which attributes the smaller rise of A_3 to fine-tuning on the full structure and excising at delivery rather than to the low starting point of A_2 .

The general-capability gains of post-training still reach the delivered model. On Bio, the MMLU of A_3 rises by 1.5–1.8 points over fine-tuning, and its IFEval is comparable or even better than the original model under the same S_{std} (Table 12). Standard post-training thus improves the delivered model as it improves the original one, while the hazardous capability of the delivered model stays low. In REX, capability growth and risk growth are decoupled.

4.5. Ablations

The two mechanisms that REX adds to selective gradient masking together leave the least target knowledge in the retained parameters (Table 4). Each variant is measured by the WMDP of its excised model A_1 before residual cleanup. Gradient masking alone, the localization rule of SGTm [Shilov et al., 2025] applied inside our expanded model, leaves A_1 at 68.4%. The rule keeps target data from being written into the original experts, but it does not act on the target knowledge they already hold, which is the gap between localization during pretraining and relocation in a pretrained model (Section 2). With the pre-top- K activation mask on general data and the routing-differentiation losses together, the full method lowers A_1 to 64.3%. Removing either mechanism leaves more target knowledge, 65.9% without the pre-top- K activation mask and 70.2% without routing differentiation, so directed

Table 4 | Relocation-mechanism ablation on Intern-S2 / Bio (%). Each row enables the marked mechanisms: selective gradient masking on target data (Grad.), the pre-top- K activation mask on general data (Act.) and the routing-differentiation losses (Route). The Grad. only row keeps the masking rule that REX shares with SGTm [Shilov et al., 2025], applied to the same expanded model and recipe. A_1 is the relocated full model F_1 with the target experts excised, before residual cleanup, and its WMDP is the target knowledge that stays in the retained parameters; $\Delta\text{WMDP} = F_1 - A_1$ is the part that the excision takes away. All rows share one starting model, token budget, data order and relocation recipe, so the REX row is the model in the other tables.

	Grad.	Act.	Route	A_1 WMDP↓	A_1 MMLU	ΔWMDP
REX	✓	✓	✓	64.3	82.3	15.1
w/o Act.	✓	–	✓	65.9	82.2	14.8
w/o Route	✓	✓	–	70.2	83.6	13.8
Grad. only	✓	–	–	68.4	83.3	15.8

Table 5 | Pipeline ablation (%). Base: removing either step of base-model delivery; R applies the RMU operator, hyperparameters and forget budget of the residual cleanup directly to B . SFT: the same 250M-token S_{std} applied to R , to the excised base A_2 , and to the recombined full model F_2 with the target experts excised at delivery. Shaded rows are the full pipeline; deeper blue marks a larger share of the above-chance WMDP accuracy of B removed, as in Table 2.

		Qwen3.5		Intern-S2		Gemma 4	
		WMDP↓	MMLU	WMDP↓	MMLU	WMDP↓	MMLU
Bio							
Base	Excision only (A_1)	62.9	78.6	64.3	82.3	70.2	72.1
	RMU only (R)	41.6	81.8	33.3	84.5	48.5	72.2
	Excision + RMU (A_2)	32.4	77.2	29.6	81.0	33.6	68.7
SFT	SFT on R (R -SFT)	78.8	82.0	81.0	84.6	71.5	74.1
	SFT on A_2 (A_3 -direct)	63.0	79.4	64.6	83.0	55.7	71.7
	SFT on F_2, excise (A_3)	42.8	78.7	55.0	82.5	41.3	70.5
CYBER							
Base	Excision only (A_1)	43.2	81.0	45.1	84.2	48.1	73.6
	RMU only (R)	31.8	82.3	30.9	84.9	30.9	72.1
	Excision + RMU (A_2)	28.2	80.7	26.3	84.1	29.4	71.9
SFT	SFT on R (R -SFT)	43.5	82.3	44.7	84.2	37.6	73.7
	SFT on A_2 (A_3 -direct)	35.9	81.6	35.9	84.2	35.7	73.3
	SFT on F_2, excise (A_3)	27.3	81.3	29.8	84.0	34.0	73.3

forgetting in the original experts and the routing path into the target experts both contribute to moving existing knowledge out.

On the base model, excision and residual cleanup are complementary. In all six settings, A_2 has lower WMDP than both excision alone (A_1) and the same cleanup applied without excision (R) (Table 5). Excision without cleanup leaves 30.5–36.6 points of WMDP-Bio in the retained parameters, which the residual cleanup then removes. R applies the same RMU operator, hyperparameters and forget budget directly to the original model, so its only difference from A_2 is the starting point of the cleanup. It stays 3.7–14.9 points above A_2 on Bio and 1.5–4.7 points above it on Cyber, and this gap is what relocation and excision contribute to the base model. The higher MMLU of R lies almost entirely in the subjects adjacent to the target domain: on Intern-S2 / Bio, R and A_2 score 85.8% and 68.2% on the adjacent subjects but 84.2% and 83.2% on all others (Table 13). Cleanup alone thus suppresses answers to the target benchmark while knowledge in the adjacent subjects stays, whereas excision removes it together with the target domain, as discussed in Section 4.3.

Under the same standard post-training, the three routes form the same ordering in every setting,

R -SFT above A_3 -direct above A_3 , and each gap in this ordering isolates one part of the pipeline (Table 5, SFT rows). Excision is the difference between R -SFT and A_3 -direct, which fine-tune the cleaned original model R and the excised and cleaned base A_2 , respectively. It accounts for 15.8–16.4 points on Bio and 1.9–8.8 points on Cyber. Fine-tuning the recombined full model and excising at delivery is the difference between A_3 -direct and A_3 , which start from the same A_2 . It accounts for a further 9.6–20.2 points on Bio and 1.7–8.6 points on Cyber.

5. Discussion

Our results suggest that methods for removing hazardous knowledge should be compared on the model as it is delivered, not only on the score right after unlearning. Open-weight models are post-trained both before and after release, and prior work found that unlearning implemented as in-place modification of parameter values is undone by ordinary fine-tuning [Doshi and Stickland, 2024]. A score taken right after unlearning therefore reflects how deeply the knowledge has been suppressed, but not whether it will return under later training, and the risk of a public model is decided by its state at delivery, after post-training. Judged by this state, the comparison between methods differs from the one on the base model: methods that are comparable on the base model separate clearly after the same standard fine-tuning (Sections 4.3 and 4.4). Whether the public model of REX carries the hazardous capability depends on which parameters are delivered, a decision made only after post-training, so its effect corresponds to this evaluation point by construction.

Taken together, the results of Section 4 show that a change at the architectural level can relocate the hazardous knowledge already present in a pretrained model into removable experts and decouple hazardous from general capability in the structure of the model, which opens a direction for controlling hazardous capability beyond alignment and unlearning. Behavioral alignment changes how the model responds and parameter-level unlearning changes parameter values, but both work within a fixed set of parameters, so the hazardous knowledge always remains in the delivered model. REX instead changes the structure of the model itself. The hazardous capability is concentrated in a set of experts that can be removed as a whole, the public model no longer contains these experts, and it keeps most of its removal under standard post-training (Sections 4.2 and 4.4). This direction is aligned with knowledge localization during pretraining [Cloud et al., 2024; Shilov et al., 2025], but it does not require training from scratch and applies to the models that releasing parties already hold, whose pretraining is complete. Its only premise is that the model contains units of parameters that can be removed as a whole, and the experts and router that MoE models introduce for computational efficiency provide exactly such units. For MoE models, controlling which capabilities are released thus becomes an architectural choice rather than a matter of repeatedly adjusting parameter values.

6. Conclusion

Relocating hazardous knowledge into removable target experts changes where that knowledge is stored in an MoE model, and thereby turns its removal from an in-place modification of parameter values into a structural operation. Through expert expansion, selective gradient masking and routing differentiation, one model takes two forms, a full model that keeps the hazardous capability for authorized users and an excised model for public release, with almost the same general capability. On the base model, excision followed by a light RMU cleanup removes target knowledge to a degree comparable with the strongest parameter-level unlearning methods. Post-training on the full structure and excising at delivery gives the delivered model the lowest WMDP among usable methods after standard fine-tuning, while its general capability keeps improving with post-training. What to release thus becomes a choice of which parameters to deliver from a single trained model.

References

- [1] Jai Doshi and Asa Cooper Stickland. Does unlearning truly unlearn? a black box evaluation of LLM unlearning methods, 2024. URL <https://arxiv.org/abs/2411.12103>.
- [2] 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*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/88be023075a5a3ff3dc3b5d26623fa22-Abstract-Conference.html.
- [3] 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, Bhruhu 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28525–28550. PMLR, 2024. URL <https://proceedings.mlr.press/v235/li24bc.html>.
- [4] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Proceedings of the First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=MXLBXjQkmb>.
- [5] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.805. URL <https://aclanthology.org/2023.acl-long.805/>.
- [6] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarsz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=B1ckMDqlg>.
- [7] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL <https://www.jmlr.org/papers/v23/21-0998.html>.
- [8] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Proceedings of the First Conference on Language Modeling*, 2024. URL <https://openreview.net/pdf?id=B41hNBowLo>.
- [9] Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight LLMs. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9d3a4cdf6f70559e8c6fe02170fba568-Abstract-Conference.html.

- [10] Rishub Tamirisa, Bhruhu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fc49a629d33bc2461ed7a715ce44da68-Abstract-Conference.html.
- [11] Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 65464–65479. PMLR, 2025a. URL <https://proceedings.mlr.press/v267/wang25en.html>.
- [12] Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 15762–15778. PMLR, 2025a. URL <https://proceedings.mlr.press/v267/fan25e.html>.
- [13] Haomin Zhuang, Yihua Zhang, Kehan Guo, Jinghan Jia, Gaowen Liu, Sijia Liu, and Xiangliang Zhang. SEUF: Is unlearning one expert enough for mixture-of-experts LLMs?, 2024. URL <https://arxiv.org/abs/2411.18797>.
- [14] Alex Cloud, Jacob Goldman-Wetzler, Evžen Wybitul, Joseph Miller, and Alexander Matt Turner. Gradient routing: Masking gradients to localize computation in neural networks, 2024. URL <https://arxiv.org/abs/2410.04332>.
- [15] Igor Shilov, Alex Cloud, Aryo Pradipta Gema, Jacob Goldman-Wetzler, Nina Panickssery, Henry Sleight, Erik Jones, and Cem Anil. Beyond data filtering: Knowledge localization for capability removal in LLMs, 2025. URL <https://arxiv.org/abs/2512.05648>.
- [16] Ethan Roland, Murat Cubuktepe, Erick Martinez, Stijn Servaes, Keenan Pepper, Michael Vaiana, Diogo S. de Lucena, Judd Rosenblatt, Addie Foote, Cem Anil, and Alex Cloud. Modular pretraining enables access control. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 105584–105617. PMLR, 2026. URL <https://proceedings.mlr.press/v306/roland26a.html>.
- [17] Lei Bai, Jiaqi Cao, Chiyu Chen, Guanzhou Chen, Kai Chen, Guangran Cheng, Erfei Cui, Xuanlang Dai, Shengyuan Ding, Shangheng Du, Yanhui Duan, Yue Fan, Youqing Fang, Quan Gan, Yuanyuan Gao, Jiaye Ge, Lixin Gu, Yuzhe Gu, Qipeng Guo, Junjun He, Xin Hong, Ming Hu, Zhouqi Hua, Haian Huang, Junhao Huang, Zixian Huang, Minxi Jin, Ling kai Kong, Alexander Lam, Zehao Li, Zonglin Li, Tianhao Liang, Dahua Lin, Junyao Lin, Tianyang Lin, Zhouhan Lin, Jiangning Liu, Jin Liu, Kuikun Liu, Wenran Liu, Yifei Liu, Yuhong Liu, Yuhong Liu, Zhoumianze Liu, Ziyang Liu, Ziyu Liu, Haijun Lv, Han Lv, Chengqi Lyu, Le Ma, Ningsheng Ma, Zerun Ma, Haoyang Peng, Runyu Peng, Jifei Shan, Zixin Shang, Kou Shi, Xiang Shi, Qisheng Su, Xuerui Su, Hao Sun, Xiao Sun, Yanan Sun, Yu Sun, Huanze Tang, Yinghao Tang, Wenhui Tian, Zhongbo Tian, Bingli Wang, Haomin Wang, Jiarui Wang, Jingzhi Wang, Rui Wang, Xiquan Wang, Yi Wang, Zhecan Wang, Ziyi Wang, Zun Wang, Rubin Wei, Lianyi Wu, Wen Wu, Yue Wu, Yuhan Wu, Zhenyu Wu, Zijian Wu, Shuhao Xing, Jun Xu, Xingle Xu, Xuenan Xu, Xiangchao Yan, Ziang Yan, Bowen Yang, Danni Yang, Lin Yang, Zhiqi Yang, Qian Yao, Haochen Ye, Peng Ye, Jinhui Yin, Jiashuo Yu, Dingbo Yuan, Fei Yuan, Yuhang Zang, Bo Zhang, Chao Zhang, Chen Zhang, Hongjie Zhang, Junming Zhang, Wenlong Zhang, Wenwei Zhang, Yiming Zhang, Zhuo Zhang, Ziyang Zhang, Haiteng Zhao, Penghao Zhao, Yibo Zhao, Zhonghan Zhao, Zhihang Zhong, Bowen Zhou, Peiheng Zhou, Xin Zhou, Xinyu Zhou,

- Yunhua Zhou, Dongsheng Zhu, and Yicheng Zou. Intern-s2-preview: Scientific agentic foundation model, 2026. URL <https://arxiv.org/abs/2608.13505>.
- [18] Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL <https://qwen.ai/blog?id=qwen3.5>.
- [19] Gemma Team. Gemma 4 technical report, 2026. URL <https://arxiv.org/abs/2607.02770>.
- [20] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024. URL <https://arxiv.org/abs/2406.17557>.
- [21] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://papers.neurips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.
- [22] Bo Yang. CE-U: Cross entropy unlearning, 2025. URL <https://arxiv.org/abs/2503.01224>.
- [23] Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5549. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/f346824d37ce8d0ab8f5920514934b8f-Abstract-Conference.html.
- [24] Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance LLM unlearning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 71318–71357. PMLR, 2025. URL <https://proceedings.mlr.press/v267/yang25ak.html>.
- [25] Qizhou Wang, Jin Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9ad307ab459fb93163248a0a56959307-Abstract-Conference.html.
- [26] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL <https://kellerjordan.github.io/posts/muon/>.
- [27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/pdf?id=d7KBjmI3GmQ>.
- [28] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge, 2018. URL <https://arxiv.org/abs/1803.05457>.
- [29] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. URL <https://aclanthology.org/P19-1472/>.

- [30] 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>.
- [31] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL <https://arxiv.org/abs/2311.07911>.
- [32] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models, 2024. URL <https://arxiv.org/abs/2405.14782>.
- [33] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023. URL <https://doi.org/10.1145/3600006.3613165>.

A. Implementation details

This section gives the implementation details of REX in the order of the pipeline, followed by the training of the external baselines (Section A.4). Table 6 lists the hyperparameters of REX shared by all six settings and Table 7 those that vary across settings. Both tables are generated directly from the training configuration of each setting. Every training stage uses the same random seed, which fixes the data order and the remaining sources of randomness.

Table 6 | Hyperparameters shared by all six settings. The learning rates, routing losses and the schedule and budget of residual cleanup depend on the setting and are listed in Table 7.

<i>Expert expansion</i>		<i>Standard post-training S_{std}</i>	
Target experts per layer M	8	Tokens / per step	250M / 500K
Noise on cloned experts	10^{-5}	Packed length	31,250 (Gemma 4: 15,625)
Statistics chunk	2048 tokens	Learning rate	2×10^{-6}
General rows for statistics	first 300,000	Schedule	10% warmup, cosine to 10^{-7}
<i>Relocation</i>		<i>Evaluation</i>	
Optimizer (β_1, β_2)	AdamW (0.9, 0.999)	Weight decay / clipping	0.05 / 1
Weight decay / clipping	0 / 1	Muon momentum	0.95, Nesterov
Schedule	no warmup, cosine to 0	Muon update scale	spectral, factor 0.2
Packed length	4096	Non-matrix parameters	Adam
		Evaluation points	0M, 10M, 50M, 100M, 250M
<i>Residual cleanup</i>		Seed	42
Optimizer (β_1, β_2)	AdamW (0.9, 0.999)	Libraries	lm-eval 0.4.12, vLLM 0.19.1
Weight decay / clipping	0 / 1	Precision / max length	BF16 / 8,192
Packed length	4096	Chat template	IFEval only, no thinking

Table 7 | Hyperparameters that differ across settings. Token pairs give target / general data, multiplier pairs give original / target experts, and the BOS pair gives relocation / residual cleanup. Multipliers scale the relocation learning rate; a target-expert multiplier of 0 keeps the cloned experts fixed. The cleanup budget is one pass over the forget corpus or a fixed number of forget tokens. Hyperparameters shared by all settings are listed in Table 6.

	Qwen3.5 Bio	Intern-S2 Bio	Gemma 4 Bio	Qwen3.5 Cyber	Intern-S2 Cyber	Gemma 4 Cyber
Source-expert scale s	0.5	0.5	0.7	0.3	0.5	0.7
Relocation tokens	164M / 109M	164M / 109M	157M / 52M	7M / 5M	7M / 5M	7M / 2M
Learning rate	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Router multipliers	40 / 60	40 / 60	40 / 60	40 / 60	40 / 60	40 / 60
Expert multipliers	1 / 0	1 / 40	1 / 20	20 / 0	10 / 0	1 / 0
Mass loss τ_h (weight)	0.75 (6)	0.75 (6)	0.65 (4)	0.75 (6)	0.75 (6)	0.65 (4)
Mass loss τ_s (weight)	0.02 (1)	0.02 (1)	0.02 (1)	0.02 (1)	0.02 (1)	0.02 (1)
Rank loss δ (weight)	0.1 (30)	0.1 (30)	0.1 (30)	0.1 (30)	0.1 (30)	0.1 (30)
RMU monitor / updated layers	9 / 7-9	9 / 7-9	7 / 5-7	9 / 7-9	9 / 7-9	9 / 7-9
RMU steering coefficient	300	300	1200	300	300	1200
RMU retain weight	350	350	4.45	350	350	4.45
RMU learning rate	10^{-5}	10^{-5}	10^{-6}	10^{-5}	10^{-5}	2×10^{-4}
RMU schedule	constant	constant	warmup + cosine	constant	constant	constant
RMU forget budget	1 pass	1 pass	1 pass	13.6M tokens	13.6M tokens	1 pass
RMU master weights	BF16	BF16	FP32	BF16	BF16	BF16
RMU parameter scope (MLPs)	routed + shared	routed + shared	routed + dense	routed + shared	routed + shared	routed down
BOS per packed sequence	no / no	no / no	yes / yes	no / no	no / no	yes / no

A.1. Expansion and relocation

The routing statistics for expansion are collected on the original model B using training corpora only. The target side uses the full forget corpus and the general side a prefix of the shuffled FineWeb-Edu corpus, with long documents split into fixed-length chunks that are counted separately. Differential-activation expansion selects $M = 8$ source experts in every MoE layer, the cloned target experts receive small Gaussian noise, and the source experts are scaled by the factor s of each setting.

Relocation trains the routed experts and the routers with AdamW, while θ_{shared} and the shared-expert MLPs stay frozen. The learning rate has no warmup and decays to zero on a cosine schedule, and the original experts, the target experts and the router rows of each bank each carry their own learning-rate multiplier. Training text is packed into fixed-length sequences, with EOS appended to every document and no BOS, except that on Gemma 4 each packed sequence starts with BOS. Target and general data are mixed under the token budgets of Table 7. The Cyber forget corpus is much smaller than the Bio corpus, so the Cyber relocation budgets are correspondingly smaller.

A.2. Residual cleanup

Residual cleanup applies RMU to the expert MLPs that remain in A_1 . It monitors one early layer and updates the routed-expert MLPs and the shared-expert MLPs (the always-on dense MLP on Gemma 4) of the three layers up to it, and on Gemma 4 / Cyber it updates only the down projections of the routed experts. Routers, attention, embeddings, norms and the language-model head remain frozen. The random direction u is drawn from a fixed seed and kept for the whole cleanup, and both the forget and the retain terms are computed on labelled tokens only and averaged per sequence. The learning rate, schedule and forget budget vary across settings and are listed in Table 7. The matched control R of the pipeline ablation uses exactly the same configuration, and its only difference is that it starts from the original model B instead of A_1 .

A.3. Standard post-training

S_{std} is full-parameter supervised fine-tuning on the Megatron backend of ms-swift. The data is a text-only instruction mixture with the life-science subsets removed, converted with the chat template and thinking format of each backbone and packed into fixed-length sequences, and samples longer than the packed length are dropped. Matrix parameters are trained with Muon and all other parameters with Adam. The same recipe produces every post-training trajectory in the paper, namely $F_2 \rightarrow F_3$, $A_2 \rightarrow A_3$ -direct, $R \rightarrow R$ -SFT and the post-training of every baseline. These trajectories share the data, data order and seed, and differ only in their starting checkpoint.

A.4. Unlearning baselines

The nine baselines are trained in all six settings under one shared configuration, with the same forget corpus $\mathcal{D}_{\text{harm}}$ and the same FineWeb-Edu retain corpus $\mathcal{D}_{\text{std}}$ as REX. Every run uses a Megatron-Core backend on 8 GPUs with expert parallelism 8, tensor, pipeline and context parallelism 1, BF16 precision and seed 42. The corpora are tokenized for each backbone and packed into 4096-token sequences, and on Gemma 4 each packed sequence starts with BOS, as in relocation. All methods use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) without weight decay, gradient clipping at 1, a global batch of 32 sequences and a cosine schedule with a linear warmup over the first 5% of steps. They train all parameters, including attention, experts and routers, except RMU, which updates only layers 7–9. The forget term carries the weight $\gamma = 1$ in every method, and the retain term, where present, the weight $\alpha = 1$ unless stated otherwise. Table 8 lists the remaining hyperparameters.

Table 8 | Hyperparameters of the nine unlearning baselines. Left: method-specific hyperparameters and base learning rate on Bio / Cyber; PDU on Intern-S2 / Bio uses 10^{-6} . All methods share the configuration described in the text, including the forget weight $\gamma = 1$. Right: values that depend on the backbone, with the RMU values shared by both domains, and the Cyber epoch (23 steps each) selected by the largest $\text{Acc}_{\text{MMLU}} - \text{Acc}_{\text{WMDP}}$; Bio uses the final checkpoint.

Method	Hyperparameters	LR (Bio / Cyber)		Qwen3.5	Intern-S2	Gemma 4
CEU	–	$10^{-6} / 5 \times 10^{-6}$	<i>Backbone-specific</i>			
GA	–	$10^{-6} / 5 \times 10^{-6}$		PDU ϵ , Bio	2.51	1.98
GD	$\alpha = 1$	$10^{-6} / 5 \times 10^{-6}$		PDU ϵ , Cyber	2.51	2.05
NPO	$\beta = 0.1, \alpha = 1$, reference B	$5 \times 10^{-5} / 5 \times 10^{-5}$		RMU c	300	300
PDU	$\lambda_0 = 1, \eta = 0.1, \epsilon$ per backbone	$4 \times 10^{-6} / 5 \times 10^{-6}$		RMU α	634	350
RMU	layers 7–9 (monitor 9), c, α per backbone	$10^{-6} / 5 \times 10^{-6}$	<i>Selected Cyber epoch</i>			
SatImp	$\beta_1 = 2, \beta_2 = 0.1, \alpha = 0$	$10^{-6} / 5 \times 10^{-6}$		CEU	8	4
SimNPO	$\beta = 0.1, \delta = 0, \alpha = 1$	$5 \times 10^{-5} / 5 \times 10^{-5}$		GA	2	2
WGA	$\beta = 2, \alpha = 0$	$10^{-6} / 5 \times 10^{-6}$		GD	3	6
				NPO	3	3
				PDU	2	3
				RMU	4	2
				SatImp	2	1
				SimNPO	2	1
				WGA	8	1

The objectives below share one notation, and every formula is a loss to be minimized. For token t of a sequence x , let $o_t \in \mathbb{R}^{|V|}$ be the vocabulary logits of the trained model π_θ , $\ell_t = -\log \pi_\theta(x_t | x_{<t})$ the token negative log-likelihood and $p_t = \exp(-\ell_t)$ the probability of the correct token. We write $\bar{\ell}(x) = \frac{1}{|x|} \sum_t \ell_t$ for the length-normalized NLL of a sequence, $\mathbb{E}_{\text{harm}}[\cdot]$ for the token average over $\mathcal{D}_{\text{harm}}$, and $\mathcal{L}_{\text{std}} = \mathbb{E}_{\text{std}}[\ell_t]$ for the mean NLL on $\mathcal{D}_{\text{std}}$. The reference model π_{ref} is the frozen original model B . The symbols $\alpha, \beta, \delta, \lambda$ and η are local to this subsection.

Five baselines act on the token-level predictions of the target data. Four of them operate directly on the token NLL and differ in how they weight it:

$$\begin{aligned}
 \mathcal{L}_{\text{GA}} &= -\gamma \mathbb{E}_{\text{harm}}[\ell_t], \\
 \mathcal{L}_{\text{GD}} &= -\gamma \mathbb{E}_{\text{harm}}[\ell_t] + \alpha \mathcal{L}_{\text{std}}, \\
 \mathcal{L}_{\text{WGA}} &= -\mathbb{E}_{\text{harm}}[\text{stopgrad}(p_t^\beta) \ell_t], \\
 \mathcal{L}_{\text{SatImp}} &= -\mathbb{E}_{\text{harm}}[\text{stopgrad}(p_t^{\beta_1} (1 - p_t)^{\beta_2}) \ell_t].
 \end{aligned}$$

Here `stopgrad` preserves the forward value but blocks its gradient, so the probability weights are recomputed at each step and treated as constants during backpropagation. GA [Jang et al., 2023] raises the NLL of the target data, and GD [Maini et al., 2024] adds standard language modelling on the retain data. WGA [Wang et al., 2025b] weights each token by a power of its current probability, so the tokens that the model still predicts confidently dominate the update and the tokens that are already forgotten contribute little. SatImp [Yang et al., 2025] multiplies this saturation weight by the importance factor $(1 - p_t)^{\beta_2}$. WGA and SatImp are run without the retain term ($\alpha = 0$), the form in which they produce effective forgetting in our setting. The fifth, CEU [Yang, 2025], replaces ascent with cross-entropy towards a target distribution that excludes the correct token. At each position it renormalizes the model’s own distribution over the remaining vocabulary,

$$q_t(v) = \frac{\exp(o_t(v))}{\sum_{v' \neq x_t} \exp(o_t(v'))} \quad (v \neq x_t), \quad q_t(x_t) = 0,$$

treats q_t as a fixed target and minimizes $\mathcal{L}_{\text{CEU}} = -\mathbb{E}_{\text{harm}} \left[\sum_{\nu} q_t(\nu) \log \pi_{\theta}(\nu \mid x_{<t}) \right]$, skipping the first token of each sequence.

NPO [Zhang et al., 2024] and SimNPO cast forgetting as a preference against the target sequences and operate on sequence-level likelihoods,

$$\mathcal{L}_{\text{NPO}} = -\frac{2\gamma}{\beta} \mathbb{E}_{x \sim \mathcal{D}_{\text{harm}}} \left[\log \sigma(\beta(\bar{\ell}_{\theta}(x) - \bar{\ell}_{\text{ref}}(x))) \right] + \alpha \mathcal{L}_{\text{std}},$$

where σ is the sigmoid. SimNPO [Fan et al., 2025b] is the reference-free variant. It replaces $\bar{\ell}_{\text{ref}}(x)$ with a constant margin δ , set to 0, and therefore needs no forward pass through a frozen reference.

PDU [Entesari et al., 2025] flattens the output distribution on target data and enforces retain performance as an explicit constraint. Its forget loss $\mathcal{L}_{\text{flat}} = \mathbb{E}_{\text{harm}} \left[(\max_{\nu} o_t(\nu) - \bar{o}_t)^2 \right]$, with $\bar{o}_t$ the mean logit over the vocabulary, drives the output towards uniform. The constraint $\mathcal{L}_{\text{std}} \leq \varepsilon$ enters through a Lagrangian, $\mathcal{L}_{\text{PDU}} = \gamma \mathcal{L}_{\text{flat}} + \lambda(\mathcal{L}_{\text{std}} - \varepsilon)$, whose dual variable starts at λ_0 and is updated after every step as $\lambda \leftarrow \max(0, \lambda + \eta(\mathcal{L}_{\text{std}} - \varepsilon))$, so the balance between forgetting and retention is adjusted automatically. The threshold ε is set per backbone.

The external RMU baseline [Li et al., 2024] uses the objective of Equation (8) with the original model B as reference, monitor layer 9 and all parameters of layers 7–9 trainable. It thus differs from the residual cleanup of REX in both its starting point and its parameter scope, since the cleanup starts from A_1 and updates only expert MLPs. Its steering coefficient c and retain weight α are set per backbone and shared by both domains.

The two domains differ in schedule and checkpoint selection because the Bio forget corpus is more than an order of magnitude larger than the Cyber one. Bio runs train for 3 epochs (2142 steps) with the learning rate decaying to 0 and are evaluated at the final checkpoint. Cyber runs train for 10 epochs (230 steps) with the learning rate decaying to a floor of 10^{-6} , and are evaluated after every epoch with WMDP-Cyber and 0-shot MMLU. The epoch with the largest $\text{Acc}_{\text{MMLU}} - \text{Acc}_{\text{WMDP}}$ is kept, which gives each baseline its most favourable trade-off between forgetting and retention. The selected checkpoint is the one reported in Table 2 and the starting point of the baseline’s standard post-training (Section A.3).

A.5. Evaluation

Endpoint evaluation uses lm-eval with vLLM at BF16 precision and without a chat template. WMDP and MMLU report 0-shot accuracy, ARC-C 25-shot normalized accuracy, HellaSwag 10-shot normalized accuracy and GSM8K 5-shot strict-match exact match. IFEval reports prompt-level strict accuracy with the chat template applied and thinking disabled. The Gemma 4 checkpoints carry no chat template, so for evaluation only a copy of the tokenizer receives a single-turn, text-only, non-thinking template equivalent to the ms-swift template used in training, and the checkpoint itself is unchanged. The endpoint evaluation of A_3 reloads the endpoint checkpoint and excises the target experts physically.

Evaluation along the post-training trajectory runs on the resident training model, without export or reloading. At each evaluation point the trajectory of REX is scored once with the full model, which gives F_3 , and once with the pre-top- K activation mask on the target experts, which gives A_3 . Table 9 compares this mode with physical excision of the same delivered base under the same evaluator, in the two settings that were evaluated both ways, and the two agree to within one standard error on WMDP and MMLU.

Table 9 | Pre-top- K activation mask versus physical excision of the same delivered base A_2 (%). Both columns use the evaluator of the resident trajectory evaluation, which differs from the endpoint evaluator of Table 2, so the physical values differ slightly from those reported there. The two settings shown are those in which A_2 was evaluated both ways under this evaluator.

Setting	WMDP			MMLU		
	Masked	Physical	Δ	Masked	Physical	Δ
Qwen3.5 / Cyber	27.8	27.9	-0.1	80.6	80.7	-0.1
Intern-S2 / Cyber	26.0	26.5	-0.4	84.0	84.1	-0.0

A.6. Routing diagnostic

The routing diagnostic uses only training text. With seed 42, it samples up to 512 nonoverlapping windows of at most 2048 tokens per corpus by reservoir sampling, over the full forget corpus and over the first 4096 retain documents in sorted file order. Text is encoded without BOS and with EOS appended, and each F_0/F_1 pair is measured on the same windows. The measurement hooks read the routing logits without changing them, and the model parameters are verified to be unchanged by the collection. The mass is computed from the normalized top- K gate weights without masking and before the per-expert scaling of Gemma 4, and each layer mean weights every valid token equally.

B. Supplementary results

B.1. Separation at every stage

Table 10 lists the full-versus-excised pair at every stage. Each pair shares every parameter except the target experts, so the difference between the two members of a pair is the target knowledge that those experts carry at that point in the pipeline.

B.2. Random-expert excision

Table 11 gives the per-seed results behind the five-seed summary reported in Section 4.2.

B.3. General capability

Table 12 lists the general capability of every route at the 250M-token endpoint. The IFEval level differs across backbones under the shared S_{std} for all routes alike, so it is compared within a backbone rather than across backbones. Table 13 splits MMLU into the subjects adjacent to the target domain and all other subjects.

Table 10 | Full model versus removal of the target experts at every stage (%). Δ is full minus excised. B is the original model: WMDP Qwen3.5 / Bio 83.2, Intern-S2 / Bio 83.4, Gemma 4 / Bio 78.9, Qwen3.5 / Cyber 57.6, Intern-S2 / Cyber 58.9, Gemma 4 / Cyber 50.8.

Setting	Stage	WMDP			MMLU		
		Full	Excised	$\Delta \uparrow$	Full	Excised	$\Delta \downarrow$
Qwen3.5 / Bio	Relocated (F_1 / A_1)	81.8	62.9	+18.9	80.4	78.6	+1.8
	Delivered base (F_2 / A_2)	80.8	32.4	+48.4	80.1	77.2	+2.9
	After SFT (F_3 / A_3 , 250M)	81.7	42.8	+38.9	80.7	78.7	+2.0
Intern-S2 / Bio	Relocated (F_1 / A_1)	79.4	64.3	+15.1	83.2	82.3	+1.0
	Delivered base (F_2 / A_2)	61.9	29.6	+32.3	82.4	81.0	+1.4
	After SFT (F_3 / A_3 , 250M)	78.6	55.0	+23.6	84.1	82.5	+1.5
Gemma 4 / Bio	Relocated (F_1 / A_1)	76.2	70.2	+6.0	74.0	72.1	+1.9
	Delivered base (F_2 / A_2)	71.0	33.6	+37.4	72.6	68.7	+3.9
	After SFT (F_3 / A_3 , 250M)	76.4	41.3	+35.1	73.9	70.5	+3.4
Qwen3.5 / Cyber	Relocated (F_1 / A_1)	57.6	43.2	+14.4	81.4	81.0	+0.4
	Delivered base (F_2 / A_2)	46.5	28.2	+18.4	81.4	80.7	+0.7
	After SFT (F_3 / A_3 , 250M)	50.3	27.3	+23.0	82.0	81.3	+0.8
Intern-S2 / Cyber	Relocated (F_1 / A_1)	57.9	45.1	+12.8	84.5	84.2	+0.2
	Delivered base (F_2 / A_2)	52.8	26.3	+26.6	84.4	84.1	+0.3
	After SFT (F_3 / A_3 , 250M)	55.8	29.8	+26.0	84.0	84.0	+0.0
Gemma 4 / Cyber	Relocated (F_1 / A_1)	49.6	48.1	+1.5	74.0	73.6	+0.4
	Delivered base (F_2 / A_2)	38.9	29.4	+9.5	72.5	71.9	+0.6
	After SFT (F_3 / A_3 , 250M)	45.1	34.0	+11.1	74.0	73.3	+0.6

Table 11 | Random-expert excision control on Intern-S2 / Bio (%). Each seed removes the same number of original experts in each layer of F_1 as target-expert excision, while retaining the added experts. Δ is F_1 minus the excised model; SD is the sample standard deviation across five seeds, not an evaluation standard error.

Excision	WMDP	MMLU	Δ WMDP	Δ MMLU
Seed 0	80.1	82.6	-0.7	+0.7
Seed 1	78.2	82.7	+1.3	+0.6
Seed 2	79.3	82.2	+0.1	+1.0
Seed 3	77.9	82.4	+1.5	+0.9
Seed 4	79.2	82.3	+0.2	+1.0
Mean $\pm$ seed SD	78.9 $\pm$ 0.9	82.4 $\pm$ 0.2	+0.5	+0.8
Target experts (A_1)	64.3	82.3	+15.1	+1.0

Table 12 | General capability at the 250M-token SFT endpoint (%): MMLU and IFEval (prompt-level strict, chat template, thinking disabled). For REX A_3 , IFEval reloads the endpoint and physically excises the target experts. A dagger marks a baseline that collapsed on the base model (Table 2); it is muted and takes no part in the comparisons.

Method	Bio						Cyber					
	Qwen3.5		Intern-S2		Gemma 4		Qwen3.5		Intern-S2		Gemma 4	
	MMLU	IFEval	MMLU	IFEval	MMLU	IFEval	MMLU	IFEval	MMLU	IFEval	MMLU	IFEval
Original + S_{std}	82.2	43.1	84.8	78.9	74.9	26.8	82.2	43.1	84.8	78.9	74.9	26.8
CEU	82.1 [†]	50.1 [†]	84.7 [†]	79.7 [†]	74.5 [†]	26.2 [†]	81.5 [†]	54.2 [†]	83.4 [†]	78.9 [†]	66.3 [†]	10.5 [†]
GA	80.5 [†]	51.9 [†]	46.9 [†]	79.3 [†]	27.3 [†]	16.1 [†]	80.8 [†]	46.6 [†]	81.1 [†]	81.9 [†]	73.7 [†]	17.2 [†]
GD	81.5	38.8	84.4	78.7	74.8	27.5	81.3 [†]	43.8 [†]	83.2 [†]	78.4 [†]	73.1 [†]	26.6 [†]
NPO	78.3	37.9	80.7	70.4	70.3	22.4	81.9	37.7	84.5	75.8	72.6	23.8
PDU	81.4	50.3	84.8	78.4	74.7	26.2	82.3	48.2	84.8	77.8	74.5	25.7
RMU	82.0	40.7	84.6	79.3	73.2	27.7	82.2	38.5	84.3	77.5	73.8	27.4
SatImp	82.1	48.2	84.4 [†]	79.5 [†]	74.2 [†]	25.7 [†]	81.1 [†]	50.5 [†]	83.9 [†]	80.8 [†]	71.0 [†]	15.5 [†]
SimNPO	78.3	42.5	79.9	65.8	69.0	24.4	81.9	40.7	82.8	75.8	71.4	20.1
WGA	82.0	49.2	84.6 [†]	78.9 [†]	74.3 [†]	26.4 [†]	81.0 [†]	51.8 [†]	83.7 [†]	82.6 [†]	66.9 [†]	13.3 [†]
Matched R -SFT	82.0	45.7	84.6	78.6	74.1	28.8	82.3	39.6	84.2	78.0	73.7	27.0
A_3 -direct	79.4	45.3	83.0	79.3	71.7	27.7	81.6	41.0	84.2	78.0	73.3	26.8
REX full (F_3)	80.7	43.2	84.1	78.6	73.9	26.1	82.0	39.2	84.0	79.1	74.0	27.5
REX (A_3)	78.7	44.7	82.5	78.6	70.5	28.1	81.3	38.1	84.0	76.2	73.3	26.2

Table 13 | MMLU split into target-adjacent subjects and all other subjects (accuracy, %, weighted by question count). Bio-adjacent: the ten biology and health subjects; Cyber-adjacent: the four computer-science subjects. *Share*: fraction of the questions that are adjacent. *Adj. share of drop*: fraction of the $B \rightarrow A_2$ MMLU drop that comes from adjacent subjects.

Setting	Share	Adjacent			Other subjects			Adj. share of drop
		B	A_2	A_3	B	A_2	A_3	
Qwen3.5 / Bio	15	86.9	66.9	72.8	81.7	79.0	79.7	57.1
Intern-S2 / Bio	15	88.7	68.2	76.8	84.4	83.2	83.5	75.0
Gemma 4 / Bio	15	81.5	59.8	66.5	73.1	70.3	71.2	57.6
Qwen3.5 / Cyber	3	81.8	58.5	65.3	82.5	81.4	81.8	39.5
Intern-S2 / Cyber	3	87.4	72.1	77.4	85.0	84.4	84.2	45.7
Gemma 4 / Cyber	3	69.7	47.3	59.2	74.5	72.7	73.8	27.3

A_3 is the 250M SFT endpoint.