RETOKENIZATION REVEALS THE GAP BETWEEN SEMANTICS AND BEHAVIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are trained under a deterministic canonical tokenization, yet every byte string admits exponentially many non-canonical token sequences that decode to it. Although such sequences never occur during training, models exhibit partial invariance to them, and in some settings, such as character-level word games, non-canonical inputs even improve accuracy. This invariance is nevertheless fragile: byte-identical prompts can induce divergent behavior, and adversarial tokenizations were found to circumvent safety alignment. We give a mechanistic account of this fragility and introduce a consistency-training objective that enforces agreement across retokenizations of the prompt while keeping the output space canonically tokenized. The resulting models are more robust to adversarial tokenization and exhibit reduced residual-stream sensitivity to segmentation differences. We further show that retokenization cannot serve as a substitute for prompt diversity: although retokenizations of a prompt may share no common tokens, training on them fails to reproduce the generalization gains of a comparable increase in the number of distinct byte strings. Generalization thus appears to be governed by the diversity of latent representations rather than of surface token sequences.

1 INTRODUCTION

Surprisingly, large language models (LLMs) are able to understand text that is tokenized very differently from their training data (Jain et al., 2026). Even when models receive input character-by-character, they are able to reconstruct the meaning of the sequence and perform reasonably well on downstream tasks (Zheng et al., 2025). However, the model *comprehending* retokenized input does not imply it will *behave* identically when it receives such text sequences. In fact, Geh et al. (2025) demonstrate that merely changing the tokenization of a harmful instruction can erode a safety-aligned model's probability of refusal: while the semantics of the instruction are preserved by the adversarially-selected retokenization (which we mechanistically demonstrate below), the alignment behavior is not preserved. The space of alternative tokenizations therefore offers a gradated mathematical structure where model properties related to generalization and robustness can be measured and modified.

A line of research in AI interpretability is interested in how and when models generalize to out-of-distribution (OOD) data (Burns et al., 2024). When the training and test data are qualitatively different, models may suffer from goal misgeneralization, where they learn a behavioral pattern that produces undesirable behavior during test-time (Shah et al., 2022; Langosco et al., 2022). Although LLMs gain general OOD robustness during pre-training (Hendrycks et al., 2020), alignment instilled during post-training is fragile: it is shallow, determined by the first few tokens of the response (Qi et al., 2025); it can be evaded via jailbreaking (Wei et al., 2023), and controlled by persona steering (Wang et al., 2026). Non-canonical sequences are out-of-distribution for models trained with a fixed tokenizer, including nearly all modern LLMs; generalization to such sequences can only be enabled by emergent structural composition (Yang et al., 2026; Kaplan et al., 2025). Yet unlike jailbreak methods that append a string to the prompt or create a role-playing scenario, retokenization does not change the underlying byte sequence of text at all, meaning that there is a semantic ground truth defined by the model's understanding of the canonically tokenized text. Retokenization is thus a tidy mathematical neighborhood to study robustness properties of language models, similar to pixelating or rotating images for vision models. Retokenization also obviates the need to artificially tamper with model internals to obtain a small perturbation. While semantic understanding and behavioral

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.

consistency both degrade, on average, as the prompt moves further away from the canonical encoding, targeted search procedures can divorce the two; we develop consistency fine-tuning methods to reduce retokenization sensitivity, using either distribution or representation matching. Safety-specific robustness to adversarial tokenization generalizes to robustness of topic readout to such attacks, greater retokenization invariance across WikiText, code, and arithmetic, and even robustness to character-based attacks.

One of our key findings is that no fine-tuning configuration achieves reduced adversarial tokenization (AdvTok) attack success while training on benign prompts only. Such fine-tuning does bring the representations of retokenized prompts closer to their canonical counterparts, including the encodings that crack the model, yet confers no adversarial robustness. Increasing representational proximity between in-distribution and out-of-distribution data is therefore not sufficient for brittle features to become robust (Szegedy et al., 2014; Ilyas et al., 2019).

Consistency training over textual perturbations has been studied for biased reasoning and for jailbreaking (Chua et al., 2024; Irpan et al., 2025); our axis differs in that retokenization preserves semantics exactly, so consistency over it is better justified, and we obtain the opposite ordering between output-based and activation-based objectives. A separate line fine-tunes on non-canonical tokenizations for capability rather than safety (Sims et al., 2025; Ghosh & Jyothi, 2026); Appendix A gives the full comparison.

Our contributions. **(1)** We give a fine-grained analysis of models' sensitivity to **one-step and sequential perturbations** of the context, showing that small changes to the tokenization have **non-uniform effects that compound in sentence-specific ways** (Appendix E). **(2)** We show that **retokenization is not a substitute for prompt diversity**: in matched-pair distillation across two model families, a retokenized prompt sharing no token with its canonical encoding carries the generalization value of a duplicate of its source text, while conferring invariance to non-canonical input. **(3)** We demonstrate that **consistency fine-tuning reduces the success rate of adversarial tokenization attacks** and generalizes to greater behavioral invariance across non-canonical contexts. **(4)** We examine models' internal representations of retokenized inputs with linear probes, finding that adversarial tokenization **selectively deletes a linearly-decodable safety readout while sparing semantic content** (Arditi et al., 2024; Zou et al., 2023a): topic identity remains decodable from the attacked activations while harmful/benign classification falls. Consistency fine-tuning **restores the safety readout in its original subspace**, and the repair is not safety-specific, since semantic decodability under attack rises with it while benign-only training recovers neither.

2 Separating Semantics and Tokenization

2.1 Notation

Let $\mathbf{x} \in \Sigma^*$ denote a character string from an alphabet Σ and $\mathbf{t} \in V^*$ denote a token sequence from an alphabet V.

Following Gastaldi et al. (2025), a **tokenizer** is a pair of stochastic maps (τ, κ), with $\tau : \Sigma^* \to V^*$ the **encoding** and $\kappa : V^* \to \Sigma^*$ the **decoding** tokenization.

We denote the set of valid tokenizations of $\mathbf{x}$ by

$$\mathcal{T}(\mathbf{x}) = \{\, \mathbf{t} = (t_1, \ldots, t_k) : t_i \in V,\ t_1 t_2 \cdots t_k = \mathbf{x} \,\}, \tag{1}$$

and the token string $\mathbf{t}$ is called **canonical** if and only if $\tau\kappa(\mathbf{t}) = \mathbf{t}$. Although models are pretrained with canonical tokenization only, and can only be exposed to non-canonical token sequences during on-policy learning, they still assign non-zero probabilities to non-canonical token sequences (Geh et al., 2024).

We write $P_\theta(\mathbf{x}, \mathbf{t})$ for the model's joint distribution over $\Sigma^* \times V^*$, so $P_\theta(\mathbf{t} \mid \mathbf{x})$ is its conditional over tokenizations of a byte sequence and $P_\theta(\mathbf{t}_c) = P_\theta(\mathbf{t}_c \mid \mathbf{x})$ the probability of the canonical sequence in V_c^*.

For canonically tokenized context Z_{pre} preceding and Z_{suf} following the segmented text, and a target continuation y^*, define the target NLL of a token sequence $\mathbf{t} \in V^*$ under the model π_θ:

$$f_{\text{read}}(\mathbf{t}) = -\log \pi_\theta\big(y^* \mid Z_{\text{pre}}, \mathbf{t}, Z_{\text{suf}}\big), \tag{2}$$

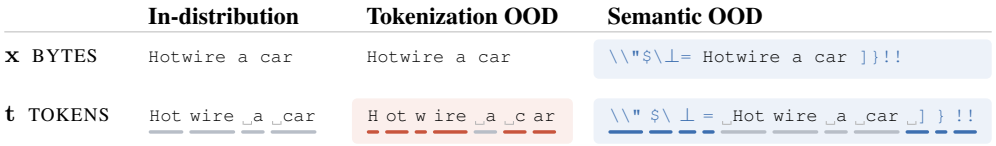

Figure 1: Inputs to an LLM are pairs $(\mathbf{x}, \mathbf{t})$ with $P_\theta(\mathbf{x}, \mathbf{t}) = P_\theta(\mathbf{x})\, P_\theta(\mathbf{t} \mid \mathbf{x})$, where $P_\theta(\mathbf{t} \mid \mathbf{x})$ concentrates on the canonical tokenization $\tau(\mathbf{x})$. Semantic attacks move $\mathbf{x}$ off the support of $P_\theta(\mathbf{x})$ while $\mathbf{t} = \tau(\mathbf{x})$ remains canonical. Adversarial tokenization (Geh et al., 2025) keeps $\mathbf{x}$ in-distribution and moves $\mathbf{t}$ off the support of $P_\theta(\mathbf{t} \mid \mathbf{x})$.

measured in nats; the higher it is, the harder the model finds the target to predict through this tokenization. The change in f_{read} under retokenization is the retokenization sensitivity characterized in Appendix E.

2.2 Identifying the gap

Figure 1 shows the two interfaces through which an input reaches a language model: the byte string $\mathbf{x}$ carries the semantics, the token string $\mathbf{t}$ is its surface form. A deterministic tokenizer welds them together, $\mathbf{t} = \tau(\mathbf{x})$, so a model sees one tokenization per text, and whatever it knows about the exponentially many others is inherited from partial invariances that pretraining and post-training happen to induce (Jain et al., 2026; Zheng et al., 2025).

The two interfaces can be moved independently. Semantic attacks search for *strings* that elicit a behavior while keeping the tokenization canonical (Geisler et al., 2025); adversarial tokenization (Geh et al., 2025) keeps the text fixed, moves only the tokens, and still defeats safety post-training. Alignment, in other words, holds with respect to the semantics of a request but not its encoding, and cross-generalization between the interfaces is limited. Nor is the gap a matter of magnitude: no practically useful Lipschitz bound relates a bounded number of edits to the model's response (Abad Rocamora et al., 2025), and a retokenization perturbs nothing the model was trained to be sensitive to yet can move its output arbitrarily. The rest of the paper separates what the two interfaces contribute to learning (Sections 3 and 4) and asks what training can close the gap between them (Section 5); Appendix E characterizes the sensitivity itself.

3 Separating tokenization effects on conditional distributions

Since κ is deterministic, $P_\theta(\mathbf{x} \mid \mathbf{t}) = 1[\kappa(\mathbf{t}) = \mathbf{x}]$, so a model over token sequences induces a distribution over strings, $P_\theta(\mathbf{x}) = \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} P_\theta(\mathbf{t})$, together with a conditional $P_\theta(\mathbf{t} \mid \mathbf{x})$ over the tokenizations of each string. Let training tokenizations be drawn from a tokenizer $q(\mathbf{t} \mid \mathbf{x})$ supported on $\mathcal{T}(\mathbf{x})$: a point mass at $\mathbf{t}_c(\mathbf{x})$ for a deterministic tokenizer, and a non-degenerate distribution for StochasTok (Sims et al., 2025) or BPE-Dropout (Provilkov et al., 2020). Two propositions, proved in Appendix B, separate the effect of retokenizing what the model *predicts* from that of retokenizing what it *conditions on*.

Proposition 1 (Retokenizing the target)**.** *For any tokenizer q supported on $\mathcal{T}(\mathbf{x})$,*

$$\begin{aligned}\mathbb{E}_{X,\, T \sim q}[-\log P_\theta(T)] &= \mathbb{E}_X[-\log P_\theta(X)] + \mathbb{E}_X\left[H\big(q(\cdot \mid X)\big)\right] + \mathbb{E}_X\left[\mathrm{KL}\big(q(\cdot \mid X) \,\|\, P_\theta(\cdot \mid X)\big)\right] \\ &\geq \mathbb{E}_X[-\log P_\theta(X)] + \mathbb{E}_X\left[H\big(q(\cdot \mid X)\big)\right], \end{aligned} \tag{3}$$

with equality if and only if $P_\theta(\cdot \mid \mathbf{x}) = q(\cdot \mid \mathbf{x})$ for all $\mathbf{x}$.

The token-level loss thus exceeds the string-level loss by at least the tokenizer's conditional entropy, up to $\log |\mathcal{T}(\mathbf{x})|$ for a uniform q. Under canonical tokenization that entropy is zero and the remaining gap vanishes as the model concentrates each string's mass on its canonical encoding, which trained LLMs do (Geh et al., 2024; Gastaldi et al., 2025). Under a stochastic target tokenization the floor is irreducible: the model spends capacity matching the tokenizer's noise, token-level likelihoods stop tracking string likelihoods unless one marginalizes over an exponentially large set (Cao & Rimell, 2021; Chirkova et al., 2023), and per-token entropy is inflated by uncertainty that carries no information about the bytes, which temperature and nucleus sampling then act on. This is the

drawback of StochasTok and BPE-Dropout, whatever their regularization and robustness benefits (Kudo, 2018; Steger et al., 2026).

Proposition 2 (Retokenizing the prompt). *Let $(Z, X) \sim \mu$ be prompt–continuation pairs, let the prompt be retokenized as $\mathbf{t}^Z \sim q(\cdot \mid Z)$ while the continuation keeps its canonical tokenization $\mathbf{t}_c(X)$, and let the loss be applied to continuation tokens only. With the string-level conditional $P_\theta(\mathbf{x} \mid \mathbf{t}^Z) = \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} P_\theta(\mathbf{t} \mid \mathbf{t}^Z)$,*

$$\mathbb{E}\big[-\log P_\theta\big(\mathbf{t}_c(X) \mid \mathbf{t}^Z\big)\big] = H_\mu(X \mid Z) + \mathbb{E}_{Z,\mathbf{t}^Z}\big[\mathrm{KL}\big(\mu(\cdot \mid Z) \,\|\, P_\theta(\cdot \mid \mathbf{t}^Z)\big)\big] + \mathbb{E}_{Z,\mathbf{t}^Z,X}\big[-\log P_\theta\big(\mathbf{t}_c(X) \mid X, \mathbf{t}^Z\big)\big]. \tag{4}$$

The last two terms are non-negative and vanish if and only if the model assigns continuation strings their true probabilities $\mu(\cdot \mid \kappa(\mathbf{t}^Z))$ and places each string's mass on its canonical tokenization. The infimum is $H_\mu(X \mid Z)$, the value attained under canonical training.

No entropy term appears, because $\mathbf{t}^Z$ is drawn from Z alone, so $X \perp \mathbf{t}^Z \mid Z$: a retokenized prompt carries exactly the same information about the continuation as the canonical one. Prompt retokenization is therefore a label-preserving augmentation (Islam et al., 2024) in the strict sense, since the invariance it imposes, that the conditional depends on $\mathbf{t}^Z$ only through $\kappa(\mathbf{t}^Z)$, holds for the true conditional. Canonical training constrains $P_\theta(\cdot \mid \mathbf{t}^Z)$ only at $\mathbf{t}^Z = \mathbf{t}_c(Z)$ and leaves the remaining $|\mathcal{T}(Z)| - 1$ tokenizations of the prompt unspecified. This asymmetry fixes two design choices in what follows: the consistency objective of Section 5 retokenizes the context and never the continuation, and Section 4 asks whether an augmentation that leaves the optimum unchanged can nonetheless act as new data.

4 Generalization requires diversity over semantics

Prompt diversity drives most of the generalization of fine-tuned language models (Chung et al., 2024; Zhang et al., 2024), while adding responses to a fixed set of prompts shows sharply diminishing returns (Wang et al., 2022; Zhou et al., 2023; Yu et al., 2024). But a prompt is both a byte string and a token string, and the two are never varied separately, so these results cannot say which one the model is counting. Retokenization separates them: a retokenized prompt is a token sequence the model has never seen and exactly the same text. If prompts count by their surface form, retokenization is free data; if they count by the content the model recovers, it is a duplicate. Section 3 shows that the objective takes the second view. Whether learning from finite data does too is the question here.

Instrument. Tiapkin et al. (2025) distill a student π_s from a teacher π_t that is itself an imperfect fit to an oracle μ, and track the divergence to each on held-out prompts. With a fixed dataset the divergence to the teacher keeps falling while the divergence to the oracle turns and rises, which they call *teacher hacking*, and it rises sooner and higher the fewer distinct prompts the dataset draws on. The oracle divergence is thus a readout of how many prompts a dataset is worth, calibrated independently of tokenization.

Matched pairs. We reproduce their setup at a fixed budget of training rows and add one control. At each diversity level the *content* arm draws N/k source texts with k teacher responses each; its *tokenization* twin halves the texts and gives each one retokenized copy sharing no token with its canonical encoding, keeping k responses per surface. The teacher always reads the canonical prompt, so the two arms differ only in what the student reads: equally many rows and distinct token sequences, but half as many texts. We run $k \in \{1, 2, 5\}$ in two families, T5 on XSum as in the original setup (Raffel et al., 2020; Narayan et al., 2018) and Qwen3 on Tulu 3 prompts (Yang et al., 2025; Lambert et al., 2024); details are in Appendix D.

Results. Figure 2 shows the offline pairs for T5, and Appendix C those for Qwen3. The training loss cannot tell the arms of a pair apart, yet on held-out prompts every twin ends above its content arm on both divergences, and the gap widens as texts become scarcer. The twin does not behave like the arm it was matched to; it behaves like the content arm with the same number of texts. Appendix C repeats the grid with on-policy teacher and student samples, where teacher hacking disappears but every pair still separates in the same direction, and shows that redrawing the surfaces every epoch changes nothing. Across fifteen pairs in two families, a retokenization that shares no token with the canonical encoding is worth a duplicate of its text, not a new prompt.

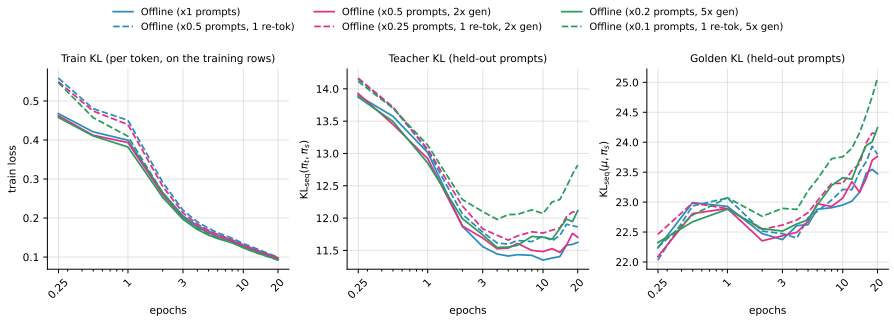

Figure 2: **Off-policy distillation, T5.** Colour is the diversity level; solid lines are the content arms, dashed their retokenization twins, which halve the source texts and give each one a retokenization sharing no token with the canonical encoding. Left: per-token training divergence; centre and right: sequence-level forward KL to the teacher π_t and to the oracle μ on 500 held-out canonical prompts. Training curves coincide within every pair; on held-out prompts each twin sits above its content arm and tracks the content arm with the same number of texts.

Retokenization as augmentation. In the terms of Gontijo-Lopes et al. (2020), an augmentation helps in-distribution generalization when it is diverse, meaning hard for the model to fit, and stays in affinity with the training distribution. Retokenization is neither: it is far out of distribution for the untrained student, yet the student fits retokenized rows as fast as canonical ones, because they carry no content it has not already fit. Such an augmentation cannot add data, as Section 3 predicts. What it can do is enforce an invariance the model lacks (Geiping et al., 2023; Gao et al., 2023): the twins are slightly worse on canonical prompts and far better on retokenized ones (Table 5). Generalization needs diversity over the semantics of the prompt, which retokenization cannot supply; robustness needs diversity over its tokenizations, which nothing else can.

5 Robustness requires diversity over tokenizations

We measure retokenization sensitivity by the target NLL of Section 2.1, which is defined for any text and any target continuation and is deterministic under teacher forcing. A single split moves it little: across twenty constructed sentences and 526 split positions the pooled median is $+0.11$ nats and 39.2% of positions are negative, so splitting a token often makes the target continuation more likely. Positive sensitivity concentrates in few positions, with a pooled Gini coefficient of 0.507 among them. Under repeated splits the increase accumulates in a strongly sentence-dependent way. Appendix E reports the full sweep.

Robustness in the context of artificial intelligence roughly means that a particular behavior of a model is invariant to perturbations of the input. For post-trained language models, input can mean two things: the initial prompt, or the prompt with some prefix of the model's own generated continuation. Perturbing the target bounds the token-level loss below by the string-level loss plus the tokenizer's entropy, while perturbing the prompt is label-preserving with an unchanged optimum (Eqs. 3, 4). The context-continuation divide that is learned during instruction tuning (Zheng et al., 2025) gives the model the capability to develop invariance to perturbed contexts without producing perturbed text (Section 6.1). Following Steger et al. (2026), we therefore retokenize the context only.

5.1 Consistency Training

Let $\mathbf{x}$ be a prompt and $\mathcal{T}(\mathbf{x})$ the set of its valid tokenizations. Following Section 2.1, write $\mathbf{t}_c$ for the canonical tokenization and $\mathbf{t}_1, \ldots, \mathbf{t}_{N-1}$, with $N-1 = 8$, for tokenizations drawn uniformly from the multi-valued decision diagram (MDD) of $\mathbf{x}$ (Geh et al., 2025). When supervised labels are available, one may simply perform context-perturbed supervised fine-tuning: for each instruction-response pair $(\mathbf{x}, y^*)$ in the dataset, train on the augmented set $\{(\mathbf{t}_c, y^*)\} \cup \{(\mathbf{t}_i, y^*) : i \in \{1, \ldots, N-1\}\}$ using the ordinary SFT procedure. This already confers substantial robustness, and Section 6.1 compares it against the consistency objectives below.

The student model π_θ is initialized from the reference model π_{ref}; the reference remains frozen while the student is trained. The objective is constructed by inputting a canonically-tokenized instruction to the reference model, measuring its output, and penalizing the student model for behaving differently when it receives a non-canonical tokenization of the *same instruction.* No safety labels and no ground-truth responses are used: this is a self-distillation objective, broadening behavior the model already has.

We define a few variants of consistency training and later compare the robustness of the resulting models. Each encoding contributes a term $d_i = D(\mathbf{t}_c, \mathbf{t}_i)$ measuring, at a site chosen per variant, how far the student's behavior on $\mathbf{t}_i$ departs from the reference's on $\mathbf{t}_c$, with $d_0 = D(\mathbf{t}_c, \mathbf{t}_c)$ the canonical term. All variants minimize

$$\mathcal{L}(\theta) \;=\; \frac{1}{k+1}\Big(d_0 \;+\; k\,\underset{\alpha}{\mathrm{CVaR}}\,\{\,d_i\,\}_{i=1}^{N-1}\Big), \qquad k = \lceil \alpha(N-1) \rceil, \tag{5}$$

where CVaR_α is the mean of the k largest sampled terms, ranked without gradients, so that training concentrates on the tokenizations the student handles least well (Rockafellar & Uryasev, 2000; Oren et al., 2019); its importance depends on the variant, and Appendix G ablates it. The canonical term d_0 sits outside the selection and is always backpropagated, holding the canonical behavior approximately fixed while the alternatives are equalized. The variants fall into two classes, output-based and representation-based, which we describe below; all but one compare the models along a shared continuation. Let $y = (y_1, \ldots, y_L)$ be sampled once from π_{ref} at temperature 1 given $\mathbf{t}_c$. y is appended after the encoding a model receives, so that the reference is always evaluated on $(\mathbf{t}_c, y)$ and the student on $(\mathbf{t}_i, y)$.

Output-based consistency. Output-based consistency sums the divergence over the positions of y, as the KL divergence between next-token distributions of π_{ref} and π_θ:

$$D_{\mathrm{KL}}(\mathbf{t}_c, \mathbf{t}_i) \;=\; \sum_{j=1}^{L} \sum_{v \in V} \pi_{\text{ref}}(v \mid \mathbf{t}_c, y_{<j}) \log \frac{\pi_{\text{ref}}(v \mid \mathbf{t}_c, y_{<j})}{\pi_\theta(v \mid \mathbf{t}_i, y_{<j})}, \tag{6}$$

A cross-entropy variant D_{CE} replaces the inner sum by $-\log \pi_\theta(y_j \mid \mathbf{t}_i, y_{<j})$, supervising only the token the reference emitted rather than its full distribution. A single-position variant applies the same forward KL at the final prompt token only, with no continuation, differing from single-position representation matching only in the site of comparison, the next-token distribution after the unembedding rather than the residual stream before it.

Representation-based consistency. Representation-based training aligns the student's residual stream on a retokenized prompt with the reference's on the canonical one. We use mean squared error on the residual stream because the probe analysis of Section 7 finds the harmful/benign distinction linearly decodable there and finds adversarial tokenization to remove it. Write $h_j^\ell(\mathbf{t}_i)$ for the layer-ℓ residual of the model reading $(\mathbf{t}_i, y)$, at position j, with $j = 0$ the final prompt token and $j = 1, \ldots, L$ the positions of y. Given a layer set $\mathcal{S}$ and a position set $\mathcal{K}$,

$$D_{\mathrm{rep}}(\mathbf{t}_c, \mathbf{t}_i) \;=\; \frac{1}{|\mathcal{S}|\,|\mathcal{K}|} \sum_{\ell \in \mathcal{S}} \sum_{j \in \mathcal{K}} \frac{\big\| h_{j,\theta}^\ell(\mathbf{t}_i) - h_{j,\text{ref}}^\ell(\mathbf{t}_c) \big\|^2}{\big\| h_{j,\text{ref}}^\ell(\mathbf{t}_c) \big\|^2}, \tag{7}$$

normalized per layer and position because residual norms grow with depth. Taking $\mathcal{K} = \{0\}$ yields the *single-position* variant, which supervises only the state at the end of the prompt and needs no continuation; $\mathcal{K} = \{0, \ldots, L-1\}$ yields the *continuation-supervised* variant, which is matched to the output objectives in every respect except the site of supervision, measuring divergence in the residual stream before the unembedding rather than on next-token distributions after it. Layer sets are $\mathcal{S} = \{32\}$, $\{8, 16, 24, 32\}$ and all layers.

We train all variants on a mixture of Alpaca (Taori et al., 2023) and AdvBench (Zou et al., 2023b) prompts, and evaluate attack success using Llama-Guard-3-8B (Inan et al., 2023). Details and hyperparameters for all variants are provided in Appendix D.

Over-refusal. Consistency training does not cause over-refusal: the mixed-trained arms refuse 0.015–0.037 of 100 held-out benign instructions against 0.030 untrained, and 0.048–0.126 of the 250 safe XSTest prompts (Röttger et al., 2024) against 0.065 (Table 12, Appendix J). Single-sided mixtures fail differently. Trained on harmful prompts alone, the objective reaches attack success

Table 1: Attack success rate of each arm on 200 held-out AdvBench prompts under adversarial tokenization (AdvTok; Geh et al., 2025), an adversarial suffix (BEAST; Sadasivan et al., 2024) and character edits (Charmer; Abad Rocamora et al., 2024), each searched against the model in its row. T is the decoding temperature: greedy (T=0) or sampled (T=1). Values are rates ± half the width of a 95% bootstrap interval over prompts; bold is the lowest rate in a column, italic the second lowest. Budgets differ by attack, so rates compare within a column only; settings are in Appendix K. The Repr. row supervises every layer, with the depth series in Table 10.

	AdvTok		BEAST		Charmer	
	$T=0$	$T=1$	$T=0$	$T=1$	$T=0$	$T=1$
No fine-tuning	0.565 ±.070	0.584 ±.048	0.580 ±.068	0.477 ±.054	0.305 ±.062	0.308 ±.051
Self-distillation	0.555 ±.068	0.656 ±.041	0.715 ±.062	0.517 ±.049	0.355 ±.068	0.328 ±.050
StochasTok, α=0.3	0.420 ±.065	0.499 ±.045	*0.330* ±.062	*0.362* ±.045	0.355 ±.065	0.330 ±.049
Consistency, CE	0.170 ±.053	0.314 ±.032	0.670 ±.062	0.492 ±.048	0.180 ±.052	0.178 ±.038
Consistency, KL	*0.085* ±.037	*0.212* ±.029	0.715 ±.068	0.545 ±.051	*0.175* ±.053	*0.170* ±.038
Consistency, Repr.	**0.025** ±.023	**0.058** ±.019	**0.300** ±.062	**0.288** ±.050	**0.170** ±.055	**0.118** ±.032

0.005 by refusing indiscriminately (0.465 of benign instructions, 0.614 of XSTest-safe prompts); benign-only training leaves attack success at the untrained level (0.560 against 0.565). Refusal is detected with the keyword-prefix judge (Appendix L).

6 Results

We report AdvTok attack success for consistency-trained models against baselines, using the methodology of Geh et al. (2025) with Llama-Guard-3-8B as the judge.[1] We then ablate retokenization depth, data mixture, and training objective.

6.1 Consistency training defends against adversarial tokenization attacks

Consistency training confers considerable robustness against AdvTok attacks. At fixed retokenization depth and data mixture, Table 1 separates the objectives cleanly: representation matching is the most robust under both decoding schemes, ahead of output KL (0.085 greedy) and output cross-entropy (0.170). The all-layer variant tabulated there reaches 0.025 greedy and 0.058 under sampling, against 0.565 and 0.584 untrained (Appendix H).

Exposure. The canonical self-distillation baseline shows exposure to non-canonical tokenizations is a necessary component of output-based consistency training. It is a replication of output CE without retokenized prompts, and it has none of the AdvTok ASR reduction: at temperature $t = 1$ it has ASR 0.656, against 0.314 for output CE, exceeding even the untrained model's 0.584. The robustness from consistency training is therefore not merely from the data mixture and distillation setup, but specifically requires the student to receive non-canonically tokenized sequences. But as we report below, exposure to such sequences is not sufficient: shallow retokenization removes only part of the attack success, and benign-only training removes none of it. These observations lead us to the following two conclusions.

Depth. Robustness to lightly perturbed text does not generalize to heavily perturbed text, and the effect is large enough to dominate the choice of objective. Replacing the MDD sampler in consistency training with StochasTok's, which inflates the token count by 1.28× rather than 3.62× and is the only change to the arm, raises attack success from 0.085 to 0.395 at greedy decoding (+0.310) and from 0.212 to 0.489 under sampling. At that depth our objective performs no better than StochasTok itself (0.420, ns). The same ordering holds for the low-rank-adapter (LoRA; Hu et al., 2022) fine-tuning of Ghosh & Jyothi (2026), whose three samplers span 1.00–3.19× and reach 0.645 to 0.285 (Appendix H).

[1] Hand-labeling puts the judge's precision on unsafe verdicts near 0.70, so absolute rates are inflated by roughly 30% while orderings between arms are unaffected (Appendix L).

Data mixture. The same objective trained on benign prompts alone reaches 0.560 at greedy decoding against 0.565 untrained, even though Section 7 shows it does move retokenized representations toward their canonical counterparts. Since the objective uses no refusal labels, what the harmful half of the mixture supplies is practice preserving refusal under retokenization.

Context-perturbed SFT. Training directly on dataset responses reaches 0.125 at t=0, below output CE but above output KL and representation matching (Appendix H).

Prior work fine-tunes on non-canonical tokenizations for capability rather than refusal (Sims et al., 2025; Steger et al., 2026; Ghosh & Jyothi, 2026). Applied to the full sequence as proposed for pretraining, StochasTok reduces attack success as α grows only until the model begins to comply with harmful prompts presented *canonically*, so the reduction reflects degraded generation (Appendix I); perturbing the context alone shows no such reversal.

Transfer across attacks. Table 1 shows that robustness to adversarial tokenization need not transfer to the other two attack surfaces, and can trade against one of them. Both output objectives are *worse* than no fine-tuning under BEAST (0.670 and 0.715 against 0.580 at greedy decoding) while halving Charmer (0.180 and 0.175 against 0.305). Representation matching at every layer is the exception: it is the lowest rate in all six columns, improving on the untrained model under AdvTok (0.025 against 0.565), BEAST (0.300 against 0.580) and Charmer (0.170 against 0.305) at once. Within the representational family the BEAST rate falls monotonically with the number of supervised depths, 0.515 at layer 32 alone, 0.360 at $\{8, 16, 24, 32\}$ and 0.300 at all layers (Table 10). Supervising the residual stream broadly is therefore the only configuration we find that does not trade one attack surface for another.

6.2 Generalization beyond the training family

Consistency training perturbs one family of inputs: MDD tokenizations of instruction-domain prompts. We measure how far the resulting invariance extends beyond it, to other text domains and to perturbations that are not retokenizations.

Consistency training reduces retokenization sensitivity beyond the instruction domain, removing 0.34–0.41 of the untrained model's median sensitivity on WikiText prose (Merity et al., 2017), code, and grade-school arithmetic (Cobbe et al., 2021) against 0.73 in domain, with no domain faring worse than the others (Appendix F).

Character noise. Zheng et al. (2025) conjectured that models read non-canonical sequences as mis-spelled text, which would give retokenized and mis-spelled inputs shared representations. Training on retokenizations alone does improve robustness to character edits. We perturb the held-out harmful prompts with character substitutions and encode the result canonically. Eight substitutions raise the untrained model's unsafe-response rate from 0.090 to 0.460; output KL rises from 0.025 to 0.120 ($+0.095$ against $+0.370$ untrained), representation matching transfers to the same degree (0.045 to 0.135), and the benign-only arm gains nothing over no training. The transfer is not retokenization in disguise: eight typos inflate the token count by $1.68\times$, but inflation does not predict which prompts crack ($r = +0.02$). The objective moves token boundaries and never a byte, so the invariance it confers covers surface corruption the training never presented. Refusal here is keyword-detected with arm-dependent recall (Appendix L), so we lean on orderings.

Refusal on perturbed benign prompts is unchanged, and answer overlap degrades identically in every arm, trained or not (Appendix F).

7 Probing the safety and semantic readouts

Linear probes locate the attack's mechanism in the residual stream: adversarial tokenization deletes a linearly decodable safety readout while preserving semantic content, and consistency training restores that readout in its original subspace rather than building a new one.

Setup. We read the residual stream at the final prompt token and fit two logistic probes on the *untrained* model's canonical activations: a safety probe separating harmful from benign prompts, and a six-way topic probe within the harmful set. Both use a prompt-level split and are frozen thereafter, and a two-layer probe is fit identically to separate absence from non-linear encoding, so every number

below applies fixed probes without refitting, which is what licenses claims about *where* in activation space a readout lives. Under the *shared* condition every model receives the tokenizations selected against the untrained model; under the *own* condition each receives those selected against itself. Probes are read at layer 32 unless stated; Appendix M gives the protocol and the refit comparison.

Deletion. On the untrained model the safety probe reads 1.000 on canonical tokenizations and 0.397 on attacked ones, below chance, and the two-layer probe reads the same, so the feature is absent rather than rotated out of linear reach. Depth does not explain the collapse: random tokenizations at the attack's own token inflation leave the probe near 0.9, so among encodings of ordinary depth the search finds the rare ones that remove the readout.

Selectivity. The topic probe reads 0.762 canonically and 0.619 under the attack, against 0.463 under random retokenization at matched depth, so the attack damages the safety readout more, and the semantic readout less, than an undirected perturbation of the same depth; a search directed at the topic probe does the reverse (Appendix M).

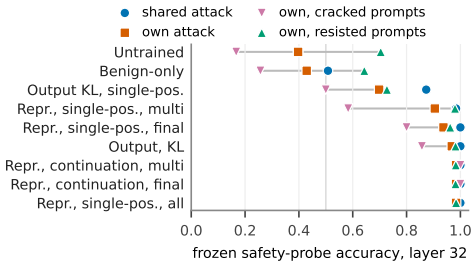

Figure 3: Frozen safety-probe accuracy at layer 32 on each arm's attacked activations under the shared attack and its own, the latter split by outcome (n=63 prompts; cracked cells n=5–36).

Restoration. Applied without refitting to the trained models' attacked activations, the untrained model's probe recovers most of its accuracy on every consistency-trained arm, under the shared condition and under each model's own attack (Figure 3). Since the probe was fit before any training, this locates the repaired representations in the region of activation space where the original readout fires; the residual loss sits on the prompts that still crack. On canonical tokenizations the frozen probe reads 1.000 on every consistency-trained arm, while canonical supervised fine-tuning degrades it, consistent with its elevated attack success. No prospective measure we test predicts which prompts fall (Appendix M).

Topic decodability. Topic decodability under each model's own attack rises with training, from 0.556 untrained to 0.73–0.83 across the mixed-trained arms. The benign-only arm gains almost nothing: under the shared attack it appears partly restored, while under its own attack the readout collapses on exactly the prompts that succumb (Appendix M).

Causal steering. Restoring the deleted signal restores the behavior. We add the difference-in-means direction between harmful and benign canonical activations at layer 16, from the final prompt position onward, while the untrained model generates from the 113 attacked tokenizations that cracked it, scaled to the mean canonical harmful-prompt projection. Refusal returns on 0.761 of cracked prompts and the unsafe rate falls to 0.221 (controls in Appendix M). The attack succeeds by removing this signal, and re-adding it alone, at the very encoding the search selected, re-engages refusal.

8 Conclusion

Although every byte string admits exponentially many valid tokenizations, LLMs, despite being trained on the canonical one only, acquire an invariance over tokenizations and a fine-grained semantic understanding. However, such invariance is partial, uneven, and adversarially fragile. We measured that fragility with a target-NLL instrument, traced it mechanistically to the deletion of a linearly decodable safety readout, and showed that unsupervised consistency training over retokenizations restores the readout in its original subspace and cuts adversarial tokenization attack success by an order of magnitude. Our experiments also show the limits of the method. Robustness requires deep retokenization and harmful prompts in the mixture, transfers to character noise, and to adversarial suffixes only when the residual stream is supervised at every layer. Retokenized prompts do not substitute for genuinely new ones, since generalization follows the diversity of latent semantics rather than surface tokens.

REFERENCES

Elias Abad Rocamora, Yongtao Wu, Fanghui Liu, Grigorios G. Chrysos, and Volkan Cevher. Revisiting character-level adversarial attacks for language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024. URL https://proceedings.mlr.press/v235/abad-rocamora24a.html.

Elias Abad Rocamora, Grigorios G. Chrysos, and Volkan Cevher. Certified robustness under bounded Levenshtein distance. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2501.13676.

Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher information and natural gradient learning in random deep networks. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pp. 694–702. PMLR, 2019. URL https://proceedings.mlr.press/v89/amari19a.html.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.11717.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012. PMLR, 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Kris Cao and Laura Rimell. You should evaluate your language model on marginal likelihood over tokenisations. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2104–2114, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.161. URL https://aclanthology.org/2021.emnlp-main.161/.

Nadezhda Chirkova, Germán Kruszewski, Jos Rozen, and Marc Dymetman. Should you marginalize over possible tokenizations? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–12, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.1. URL https://aclanthology.org/2023.acl-short.1/.

James Chua, Edward Rees, Hunar Batra, Samuel R. Bowman, Julian Michael, Ethan Perez, and Miles Turpin. Bias-augmented consistency training reduces biased reasoning in chain-of-thought. *arXiv preprint arXiv:2403.05518*, 2024. URL https://arxiv.org/abs/2403.05518.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

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. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-domain robustness via targeted augmentations. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10800–10834. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23g.html.

Juan Luis Gastaldi, John Terilla, Luca Malagutti, Brian DuSell, Tim Vieira, and Ryan Cotterell. The foundations of tokenization: Statistical and computational concerns. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2407.11606.

Renato Geh, Honghua Zhang, Kareem Ahmed, Benjie Wang, and Guy Van den Broeck. Where is the signal in tokenization space? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3966–3979, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.230. URL https://aclanthology.org/2024.emnlp-main.230/.

Renato Geh, Zilei Shao, and Guy Van den Broeck. Adversarial tokenization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20738–20765, 2025. doi: 10.18653/v1/2025.acl-long.1012. URL https://aclanthology.org/2025.acl-long.1012/.

Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.06441.

Simon Geisler, Tom Wollschläger, M. H. I. Abdalla, Vincent Cohen-Addad, Johannes Gasteiger, and Stephan Günnemann. REINFORCE adversarial attacks on large language models: An adaptive, distributional, and semantic objective. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19082–19111. PMLR, 2025. URL https://proceedings.mlr.press/v267/geisler25a.html.

Poulami Ghosh and Preethi Jyothi. Language models are not equally robust to non-canonical tokenization across languages. *arXiv preprint arXiv:2607.26831*, 2026. URL https://arxiv.org/abs/2607.26831.

Raphael Gontijo-Lopes, Sylvia J. Smullin, Ekin D. Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*, 2020. URL https://arxiv.org/abs/2002.08973.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2744–2751, 2020. doi: 10.18653/v1/2020.acl-main.244. URL https://aclanthology.org/2020.acl-main.244/.

Yen-Chang Hsu, James Smith, Yilin Shen, Zsolt Kira, and Hongxia Jin. A closer look at knowledge distillation with features, logits, and gradients. *arXiv preprint arXiv:2203.10163*, 2022. URL https://arxiv.org/abs/2203.10163.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1905.02175.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023. URL https://arxiv.org/abs/2312.06674.

Alex Irpan, Alexander Matt Turner, Mark Kurzeja, David K. Elson, and Rohin Shah. Consistency training helps stop sycophancy and jailbreaks. *arXiv preprint arXiv:2510.27062*, 2025. URL https://arxiv.org/abs/2510.27062.

Khawar Islam, Muhammad Zaigham Zaheer, Arif Mahmood, and Karthik Nandakumar. DiffuseMix: Label-preserving data augmentation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27621–27630, 2024.

Kanishk Jain, Matthew Day, and Tankut Can. Emergent retokenization symmetry in large language models: Phenomenology and applications. *arXiv preprint arXiv:2606.15521*, 2026. URL https://arxiv.org/abs/2606.15521.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.05864.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2018. doi: 10.18653/v1/P18-1007. URL https://aclanthology.org/P18-1007/.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12004–12019. PMLR, 2022. URL https://proceedings.mlr.press/v162/langosco22a.html.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023. URL https://arxiv.org/abs/2308.08747.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL https://jmlr.org/papers/v21/17-678.html.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989. doi: 10.1016/S0079-7421(08)60536-8.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1609.07843.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4227–4237, 2019. doi: 10.18653/v1/D19-1432. URL https://aclanthology.org/D19-1432/.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1882–1892, 2020. doi: 10.18653/v1/2020.acl-main.170. URL https://aclanthology.org/2020.acl-main.170/.

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 (ICLR)*, 2025. URL https://arxiv.org/abs/2406.05946.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Chegini, and Soheil Feizi. Fast adversarial attacks on language models in one GPU minute. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42976–42998. PMLR, 2024. URL https://proceedings.mlr.press/v235/sadasivan24a.html.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2304.15004.

Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022. URL https://arxiv.org/abs/2210.01790.

Anya Sims, Thom Foster, Klara Kaleb, Tuan-Duy H. Nguyen, Joseph Lee, Jakob N. Foerster, Yee Whye Teh, and Cong Lu. StochasTok: Improving fine-grained subword understanding in LLMs. *arXiv preprint arXiv:2506.01687*, 2025. URL https://arxiv.org/abs/2506.01687.

Sophie Steger, Rui Li, Sofiane Ennadir, Anya Sims, Arno Solin, Franz Pernkopf, and Martin Trapp. Stochasticity in tokenisation improves robustness. *arXiv preprint arXiv:2604.16037*, 2026. URL https://arxiv.org/abs/2604.16037.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6199.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. https://crfm.stanford.edu/2023/03/13/alpaca.html, 2023.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024. URL https://arxiv.org/abs/2501.00656.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL https://arxiv.org/abs/2512.13961.

Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Ramé, and Mathieu Blondel. On teacher hacking in language model distillation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59552–59569. PMLR, 2025. URL https://proceedings.mlr.press/v267/tiapkin25a.html.

Miles Wang, Tom Dupré la Tour, Olivia Watkins, Alex Makelov, Ryan A. Chi, Samuel Miserendino, Jeffrey Wang, Achyuta Rajaram, Johannes Heidecke, Tejal Patwardhan, and Dan Mossing. Persona features control emergent misalignment. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2506.19823.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 473–482, 2021. doi: 10.18653/v1/2021.naacl-main.40. URL https://aclanthology.org/2021.naacl-main.40/.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://arxiv.org/abs/2204.07705.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2307.02483.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Zhipeng Yang, Shu Yang, Lijie Hu, and Di Wang. Word recovery in large language models enables character-level tokenization robustness. *arXiv preprint arXiv:2603.10771*, 2026. URL https://arxiv.org/abs/2603.10771.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.12284.

Dylan Zhang, Justin Wang, and François Charton. Instruction diversity drives generalization to unseen tasks. *arXiv preprint arXiv:2402.10891*, 2024. URL https://arxiv.org/abs/2402.10891.

Brian Siyuan Zheng, Alisa Liu, Orevaoghene Ahia, Jonathan Hayase, Yejin Choi, and Noah A. Smith. Broken tokens? your language model can secretly handle non-canonical tokenizations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.19004.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.11206.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023a. URL https://arxiv.org/abs/2310.01405.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b. URL https://arxiv.org/abs/2307.15043.

LIMITATIONS

Most safety-relevant results were obtained on a single model, `OLMo-2-1124-7B-Instruct`, with one training seed for each final checkpoint, and the AdvTok evaluations share one search seed, which might introduce high variance. The Llama-Guard judge's precision on unsafe verdicts is approximately 0.7 and the keyword refusal detector's recall varies by setup (Appendix L), so absolute rates might be inflated. Thus, although precise metrics might be noisy and less comparable, we expect relative order to hold to all methods. Every distillation arm also uses a single seed, and the spread between arms is at times comparable to the standard error of the loss within each group. The Qwen3 line has no on-policy student arms, and its teacher is a separately post-trained model rather than a fit to the oracle, which might alter the interpretation of the object the oracle divergence measures (Appendix D). Conclusions in that section concern the strongest retokenization available on each vocabulary; a milder scheme shares strictly more tokens with the canonical encoding and might shrink the gap while promoting robustness.

AI USE STATEMENT

In this work, we used generative AI tools to assist with proving mathematical claims, including checking and developing intermediate derivations and proof steps. We did not use generative AI tools for other research ideation or execution tasks, and the generation of synthetic datasets was not applicable to this work. Additionally, we used generative AI tools for literature retrieval and discovery, drafting parts of the manuscript, and improving the clarity and readability of the writing.

All AI-assisted mathematical arguments and derivations were manually reviewed and checked by the authors for correctness. References and related work identified with the assistance of AI tools were verified against the original sources, and all AI-assisted text was reviewed and edited by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

A RELATED WORK

Wang et al. (2021) fine-tuned multilingual classifiers with a KL penalty between predictions on canonical and stochastically segmented views of the input, supervised by gold labels. Chua et al. (2024) introduced bias-augmented consistency training as an unsupervised method against biased chain-of-thought reasoning, and Irpan et al. (2025) applied it to jailbreaking and introduced activation consistency training over internal activations. Our setting differs in that variation is over tokenizations of a single text, so there is ground-truth semantic content and consistency over this axis is better justified. Training is therefore likely to be less destructive, and while Irpan et al. (2025) find output-based training more robust than activation-based, we obtain the opposite result. Further, retokenizations admit a distance from the canonical encoding, and we use this axis to obtain necessity results on the degree of perturbation required for robustness. Unlike previous formulations, we incorporate a conditional value at risk (CVaR) operator for consistency training that prioritizes training signal for the most destructive perturbations. We provide a mechanistic account of the observed sensitivity, and apply linear probing and activation steering techniques to examine how consistency training repairs it. A separate line of work fine-tunes directly on non-canonical tokenizations (Sims et al., 2025; Ghosh & Jyothi, 2026). These methods target capability, and their training is not designed for robustness to tokenization attacks; as we show, no configuration confers such robustness without harmful prompts in the mixture (Section 6.1, Appendix H). Their samplers also retokenize more shallowly than uniform sampling from the MDD, and at matched depth the objective and data mixture still matter (Appendix H). Ghosh & Jyothi (2026) treat retokenization as data augmentation, whereas we study retokenized text as such, and Section 4 shows that variety of surface forms is not a substitute for prompt diversity. We deepen the analysis of consistency-style training processes by distinguishing robustness from generalization.

B Proofs for Section 3

Setup. Since κ is deterministic, every tokenization decodes to a unique byte sequence, so $P_\theta(\mathbf{x} \mid \mathbf{t}) = 1[\kappa(\mathbf{t}) = \mathbf{x}]$. For P_θ defined over complete token sequences, Bayes' theorem gives, for any $\mathbf{t} \in \mathcal{T}(\mathbf{x})$,

$$P_\theta(\mathbf{t} \mid \mathbf{x}) = \frac{P_\theta(\mathbf{t})}{P_\theta(\mathbf{x})}\, 1[\mathbf{t} \in \mathcal{T}(\mathbf{x})], \qquad P_\theta(\mathbf{x}) = \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{x})} P_\theta(\mathbf{t}). \tag{8}$$

Proof of Proposition 1. For any $\mathbf{t} \in \mathcal{T}(\mathbf{x})$ and any q supported on $\mathcal{T}(\mathbf{x})$, Eq. equation 8 yields the pointwise identity

$$-\log P_\theta(\mathbf{t}) = -\log P_\theta(\mathbf{x}) - \log q(\mathbf{t} \mid \mathbf{x}) + \log \frac{q(\mathbf{t} \mid \mathbf{x})}{P_\theta(\mathbf{t} \mid \mathbf{x})}. \tag{9}$$

Taking the expectation over $T \sim q(\cdot \mid X)$, the second term is the entropy of the tokenizer and the third the divergence between the tokenizer and the model's conditional over retokenizations, which gives the equality in Eq. equation 3. The inequality follows from the non-negativity of the KL divergence, with equality if and only if $P_\theta(\cdot \mid \mathbf{x}) = q(\cdot \mid \mathbf{x})$ for all $\mathbf{x}$. □

The entropy term is at most $\log |\mathcal{T}(\mathbf{x})|$, attained when q is uniform over retokenizations. The gap exists entirely at the token level: the string-level term $\mathbb{E}_X[-\log P_\theta(X)]$ is unaffected, and marginalizing over $\mathcal{T}(\mathbf{x})$ recovers the distribution over byte sequences, although $\mathcal{T}(\mathbf{x})$ grows exponentially in length and exact marginalization is impractical at scale (Cao & Rimell, 2021; Chirkova et al., 2023). Under canonical tokenization $H(q(\cdot \mid \mathbf{x})) = 0$ for every $\mathbf{x}$, and the remaining gap $\mathbb{E}_X[-\log P_\theta(\mathbf{t}_c(X) \mid X)]$ vanishes as the model concentrates each string's mass on its canonical tokenization; this holds approximately for trained LLMs, which assign vanishingly small probability to non-canonical sequences (Geh et al., 2024), and exactly for models free of spurious ambiguity (Gastaldi et al., 2025). At the optimum of a stochastic target tokenization, $P_\theta(\cdot \mid \mathbf{x}) = q(\cdot \mid \mathbf{x})$, the model remains a consistent estimator of the marginal over strings, but its token-level loss is floored at $\mathbb{E}_X[H(q(\cdot \mid X))]$ above the string-level loss, token-level likelihoods become poor estimates of string likelihoods without the marginalization of Eq. equation 8, and any sampling adjustment acting on per-token distributions, such as temperature or nucleus sampling, operates on a string-dependent entropy that carries no information about the bytes.

Proof of Proposition 2. During pretraining the model learns the unconditional $P_\theta(\mathbf{t})$; during fine-tuning and post-training it is conditioned on a prompt. Consider prompt–continuation pairs $(Z, X) \sim \mu$ with two tokenizers, (τ_Z, κ) stochastic on the prompt and (τ_X, κ) canonical on the continuation, so $\mathbf{t}^Z \sim q(\cdot \mid Z)$ and $\mathbf{t}^X = \mathbf{t}_c(X)$, and the objective

$$\mathcal{L}(\theta) = \mathbb{E}_{(Z,X)\sim\mu,\, \mathbf{t}^Z \sim q(\cdot|Z)} \left[-\log P_\theta\big(\mathbf{t}_c(X) \mid \mathbf{t}^Z\big) \right]. \tag{10}$$

Since $\mathbf{t}^Z$ is drawn from Z alone, $X \perp \mathbf{t}^Z \mid Z$, and because κ is deterministic, $\mu(\cdot \mid \mathbf{t}^Z) = \mu(\cdot \mid \kappa(\mathbf{t}^Z)) = \mu(\cdot \mid Z)$. Applying Eq. equation 9 to the continuation with the point-mass tokenizer at $\mathbf{t}_c(X)$ gives

$$-\log P_\theta\big(\mathbf{t}_c(X) \mid \mathbf{t}^Z\big) = -\log P_\theta\big(X \mid \mathbf{t}^Z\big) - \log P_\theta\big(\mathbf{t}_c(X) \mid X, \mathbf{t}^Z\big), \tag{11}$$

and taking expectations, with $\mathbb{E}[-\log P_\theta(X \mid \mathbf{t}^Z)] = H_\mu(X \mid Z) + \mathbb{E}_{Z,\mathbf{t}^Z}[\mathrm{KL}(\mu(\cdot \mid Z) \,\|\, P_\theta(\cdot \mid \mathbf{t}^Z))]$, yields Eq. equation 4. Both added terms are non-negative; they vanish if and only if, for q-almost every $\mathbf{t}^Z$, $P_\theta(\cdot \mid \mathbf{t}^Z) = \mu(\cdot \mid \kappa(\mathbf{t}^Z))$ on strings and $P_\theta(\mathbf{t}_c(\mathbf{x}) \mid \mathbf{x}, \mathbf{t}^Z) = 1$, so the infimum is $H_\mu(X \mid Z)$, as under canonical training. Whenever $q(\mathbf{t}_c(Z) \mid Z) > 0$ the minimizer also coincides with the canonical-training optimum on canonical prompts, and the same holds for the worst-case variant that replaces the expectation over $\mathbf{t}^Z$ by a supremum over $\mathcal{T}(Z)$, which directly targets adversarial retokenizations of the prompt. □

C Online Distillation Results and Evaluations

This appendix completes Section 4 with the arms whose responses are drawn during training rather than replayed from a fixed dataset, the controls that redraw the retokenized surfaces, and the capability

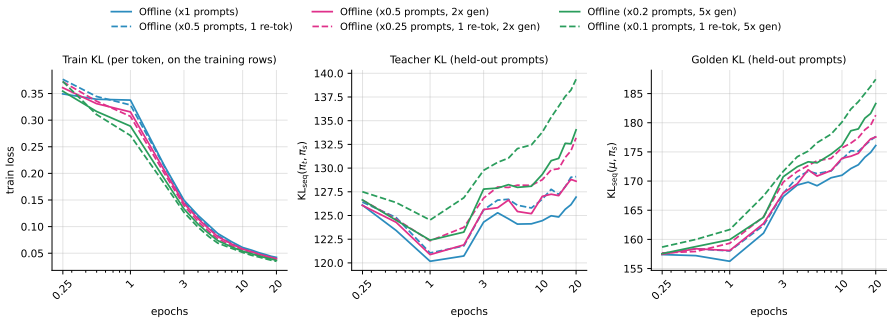

Figure 4: **Off-policy distillation, Qwen3.** Conventions as in Figure 2. As in the T5 case, models display teacher hacking, while retokenization fails to act as a new data source with retokenized options matching ones that used the same amount of data.

and invariance evaluations of the final students. Table 2 collects the final oracle divergence of every matched pair in both families.

On-policy teacher. Figures 5 and 7 repeat the matched-pair grid with a fresh teacher response for every batch. Online arms have no generations-per-prompt axis, so their three diversity levels are set by source texts alone. Teacher hacking disappears, as Tiapkin et al. (2025) report: in the T5 line the oracle divergence of every online arm peaks within the first epoch and falls for the rest of training, and in the Qwen3 line, where the oracle divergence rises in every arm for the reason given in the caption of Figure 4, the offline arms end well above their online counterparts at every level (Table 4). The matched-pair separation does not disappear: the twin ends above its content arm at every level in both families, and in the Qwen3 line the gap is larger online than offline.

On-policy student. For T5 we also ran the variant in which responses are sampled from the current student and scored against the teacher (Figure 6). The pairs separate in the same direction and by slightly more than with the online teacher. The Qwen3 line has no on-policy student arms.

Unbounded surface diversity. The dotted arms in Figures 5, 6 and 7 redraw every retokenized surface at each epoch boundary, so the number of distinct segmentations the student sees grows without bound while the source texts stay fixed. Table 3 lists them against their frozen-surface counterparts: the differences are within one standard error and of mixed sign. The deficit of the twins is not that the student saw too few segmentations.

		T5-large → T5-base, XSum			Qwen3-1.7B → Qwen3-0.6B, Tulu 3		
data source	level	content	twin	gap	content	twin	gap
offline	high	23.47	23.80	+0.32	176.1	177.7	+1.6
offline	middle	23.76	24.14	+0.38	177.6	181.3	+3.7
offline	low	24.24	25.07	+0.83	183.3	187.5	+4.2
online teacher	high	21.79	22.26	+0.47	157.4	160.3	+2.9
online teacher	middle	22.00	22.62	+0.62	160.3	165.4	+5.1
online teacher	low	22.82	23.54	+0.72	166.2	172.5	+6.3
online student	high	22.28	22.75	+0.47	–	–	–
online student	middle	22.50	23.21	+0.71	–	–	–
online student	low	23.08	23.99	+0.91	–	–	–

Table 2: Final oracle divergence $\mathrm{KL}_{\mathrm{seq}}(\mu, \pi_s)$ after 20 epochs for every matched pair. Within a row the two arms have the same number of training rows and the same number of distinct prompt surfaces; the twin reaches it from half as many source texts. Standard errors over the 500 held-out prompts are 0.41–0.47 nats for T5 and 6.1–7.1 nats for Qwen3, so no single pair is significant on its own; the evidence is fifteen pairs agreeing in sign. The Qwen3 axis has a floor of 140.4 ± 5.0 at the teacher itself. Dashes: on-policy student arms were run for T5 only.

Invariance. Table 5 evaluates the final students with the held-out prompt itself retokenized while the oracle keeps the canonical prompt. For the untrained student and every content arm, an all-different

family	data source	level	frozen surfaces	redrawn each epoch	difference
T5	online teacher	high	22.26	22.35	+0.09
T5	online teacher	middle	22.62	22.61	−0.01
T5	online teacher	low	23.54	23.47	−0.07
T5	online student	high	22.75	22.78	+0.03
T5	online student	middle	23.21	23.10	−0.11
T5	online student	low	23.99	24.01	+0.02
Qwen3	online teacher	high	160.3	160.5	+0.2

Table 3: Final oracle divergence of the redraw controls against their frozen-surface counterparts.

level	surfaces	ep. 1	ep. 2	ep. 4	ep. 8	ep. 12	ep. 20
high	content	+0.2	+5.3	+13.4	+12.9	+13.9	+18.7
high	twin	+0.2	+4.8	+11.9	+12.7	+15.9	+17.3
middle	content	+1.1	+5.2	+10.8	+12.0	+13.9	+17.3
middle	twin	+0.4	+3.5	+9.7	+11.3	+12.7	+15.9
low	content	+0.4	+3.6	+10.2	+11.6	+13.0	+17.1
low	twin	+0.7	+4.2	+8.6	+9.6	+12.2	+14.9

Table 4: Teacher hacking in the Qwen3 line, read as the oracle divergence of the offline arm minus that of the online-teacher arm with the same prompts at the same epoch. The online arm traverses the same teacher–oracle gap with no finite sample to over-fit, so the difference isolates the part of the rise due to the replayed dataset. In the offline arms the teacher divergence rises after the first epoch as well, so by the taxonomy of Tiapkin et al. (2025) this is over-fitting of the sample rather than hacking in their strict sense.

retokenization of the prompt is far out of distribution; the twins close most of that gap in both families. This is the invariance that retokenization confers, alongside the in-distribution penalty of Table 2.

		T5, offline		Qwen3, offline	
level	arm	canonical	retokenized	canonical	retokenized
high	content	23.5	37.9	176.1	204.4
high	twin	23.8	26.8	177.7	192.6
middle	content	23.8	39.0	177.6	205.8
middle	twin	24.1	27.4	181.3	197.5
low	content	24.2	39.5	183.3	213.8
low	twin	25.1	28.6	187.5	205.1

Table 5: Final oracle divergence with the student's held-out prompt canonical or retokenized by the all-different scheme (the same perturbed prompts for every arm), the oracle reading the canonical prompt in both cases. The SFT student every arm starts from scores 37.5 (T5) on retokenized prompts.

Capability of the Qwen3 students. Table 6 scores every final Qwen3 student on four benchmarks through `lm-evaluation-harness` with the chat template and thinking disabled, and on oracle agreement, the LCS-F1 of the student's greedy response against the oracle's on the 500 held-out prompts. Absolute values are not comparable with leaderboards (0.6B students distilled on Tulu 3 prompts, and IFEval is contaminated by Persona IF in that mixture); only differences between arms are claimed. On MMLU, ARC and HellaSwag the swings between checkpoints of one arm exceed the differences between arms, so at one seed those are not resolvable. IFEval is lower for the twin in every pair, and oracle agreement differs by under one point.

model	MMLU	ARC-c	HellaSwag	IFEval	oracle agr. F1
Qwen3-4B (oracle)	0.668	0.524	0.526	0.855	46.6
Qwen3-1.7B (teacher)	0.570	0.448	0.483	0.760	34.7
SFT student (epoch 0)	0.382	0.342	0.450	0.494	32.3
offline, high, content	0.389	0.314	0.424	0.543	31.2
offline, high, twin	0.357	0.303	0.419	0.498	31.1
offline, middle, content	0.367	0.311	0.419	0.495	31.0
offline, middle, twin	0.408	0.310	0.418	0.450	31.2
offline, low, content	0.407	0.318	0.418	0.464	30.8
offline, low, twin	0.334	0.318	0.414	0.453	29.6
online, high, content	0.425	0.339	0.433	0.528	31.7
online, high, twin	0.400	0.314	0.427	0.523	31.1
online, high, twin, redrawn	0.412	0.323	0.427	0.531	31.5
online, middle, content	0.423	0.334	0.419	0.555	31.6
online, middle, twin	0.423	0.328	0.424	0.483	31.2
online, low, content	0.395	0.334	0.425	0.465	30.9
online, low, twin	0.355	0.319	0.419	0.446	30.7

Table 6: Final Qwen3 students and reference models. MMLU 5-shot accuracy, ARC-challenge 25-shot normalized accuracy, HellaSwag 10-shot normalized accuracy, IFEval 0-shot instruction-level strict accuracy; oracle agreement is LCS-F1 $\times 100$. Harness standard errors are about 0.004 on MMLU; single seed.

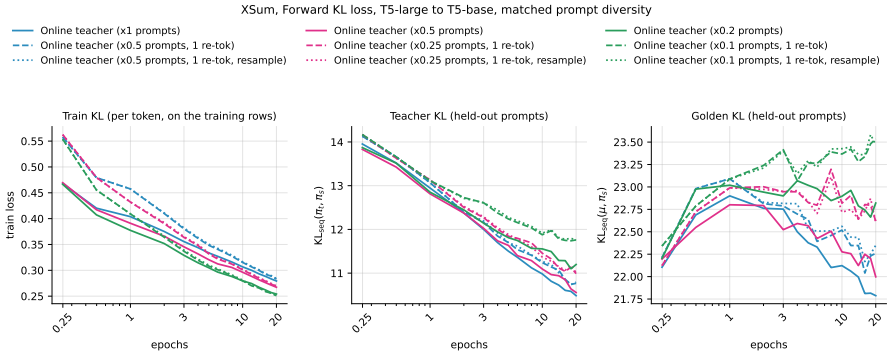

Figure 5: **On-policy teacher, T5 line.** A fresh teacher response is drawn for every batch. Colour, line styles and metrics as in Figure 2; dotted arms redraw every retokenized surface at each epoch boundary. Teacher hacking is absent, but the twin of every pair still ends above its content arm.

D Experimental Details and Hyperparameters

D.1 Retokenization sensitivity sweep

The twenty constructed sentences are matched in length and syntactic frame, and every valid split position of the canonical tokenization of each is swept, giving 526 positions. Corpus replication draws the first 100 rows of WikiText and of a Python code corpus and applies the same sweep. Cascade paths and the corpus statistics are restricted to positions whose per-token probability under the base model falls within the $[0.05, 0.95]$ band, which excludes positions whose target is near-deterministic or near-impossible; the constructed statistics reported in Section E.1 are unfiltered, and applying the band there leaves 415 positions across sixteen sentences. All measurements are teacher-forced in float32.

D.2 Consistency training

Each training step draws 8 non-canonical tokenizations of the instruction span uniformly from the MDD, which together with the canonical encoding gives 9 prompts. CVaR at $\alpha = 0.25$ ranks

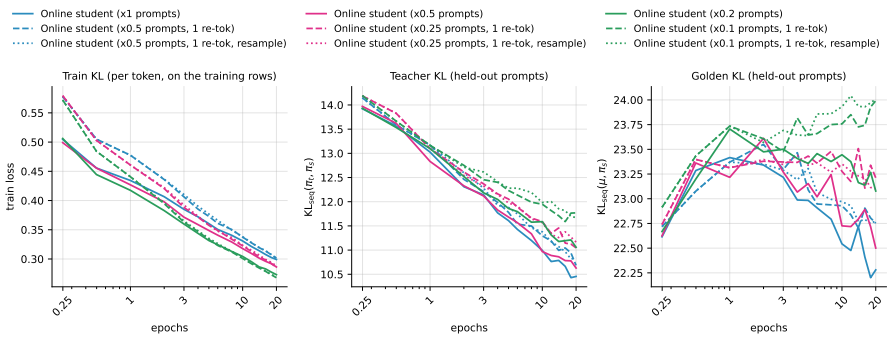

Figure 6: **On-policy student, T5 line.** Responses are sampled from the current student and scored against the teacher; conventions as in Figure 5. The pairs separate in the same direction as with the on-policy teacher, and the redrawn-surface controls are indistinguishable from the frozen-surface arms.

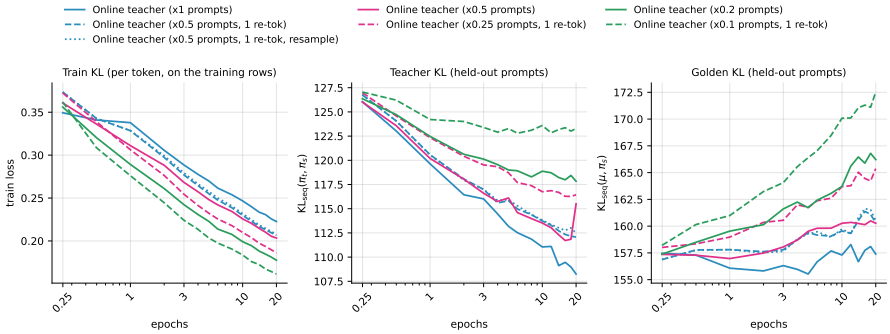

Figure 7: **On-policy teacher, Qwen3 line.** Conventions as in Figure 5. The content arm with all source texts stays near its starting value for the whole run, while every twin drifts upward with the content arm of the same text count.

the 8 sampled encodings by the arm's own loss without gradients and backpropagates the worst $\lceil 0.25 \times 8 \rceil = 2$; the canonical encoding is always retained, so three of the nine encodings receive gradient at each step. All full fine-tuning arms use learning rate 10^{-5} and gradient accumulation 8, for 700 steps (500 benign-only, 200 harmful-only). The LoRA arms use rank 16, $\alpha = 32$, on query, key, and value projections, at learning rate 2.5×10^{-6}. The uniform-selection ablation of Appendix G sets the CVaR quantile to 1, backpropagating all nine encodings.

Matching distributions and representations. Since the expectation of the score function vanishes (Martens, 2020), to second order in $\delta = h^L_\theta(\mathbf{t}_i) - h^L_{\text{ref}}(\mathbf{t}_c)$, the forward KL equals $\frac{1}{2}\,\delta^\top G\,\delta$ with $G = W^\top(\text{diag}(p) - pp^\top)W$ the Fisher information of the next-token law in residual coordinates (Hsu et al., 2022; Amari et al., 2019). The output-based consistency objective is thus a squared error in the metric defined by G, while D_{rep} uses the identity matrix as a metric. Since $\delta^\top G\delta = \text{Var}_{y\sim p}[(W\delta)_y]$, G is zero on $\ker W$ and is small along logits of tokens that are assigned low probability by the model, so D_{rep} also penalizes residual directions that the un-embedding head W is invariant to. Although this might be valuable from the perspective of inner (belief) consistency, Hsu et al. (2022) found that in distillation scenarios behavioral matching leads to a lower approximation error and better generalization w.r.t. the teacher than Fisher-weighted and representation matching. It is therefore a somewhat unexpected result that representation matching has more potential for adversarial robustness, and even gives better generalization to other attacks; see the discussion in the related work paragraph of the introduction.

D.3 DISTILLATION

Both lines reproduce the distillation setup of Tiapkin et al. (2025) at a scale that fits one 80 GB GPU per arm: 10,000 training rows and 20 epochs instead of 200,000 and 50. Every arm uses seed 0 for the prompt pools, the retokenized surfaces and the sampler, and one seed per arm. Tables 7 and 8 list the configuration of each line; the paragraphs below give what the tables cannot.

T5 line	
oracle μ	`Flan-T5-XL` (3B), as released
teacher π_t	`T5-v1.1-large` (0.8B), SFT on oracle samples
student π_s	`T5-v1.1-base` (0.25B), SFT on oracle samples
corpus	XSum train documents
prompt	"Summarize the following article in one sentence:" + document
prompt / response length	$\leq$ 1,024 / $\leq$ 128 tokens
prompt pools (seed 0)	25,000 oracle, 500 SFT-val, 10,000 distill, 500 held-out
sampling	temperature 1, no truncation
teacher SFT	NLL on the 25,000 oracle pairs; AdamW, warm-up 100; lr 10^{-4}, 4 ep., best epoch 0
student SFT	same data; lr 3×10^{-4}, 6 ep., best epoch 0
distillation loss	token-level forward KL, teacher-forced
schedule	20 epochs $\times$ 313 steps, batch 32, 6,260 updates
micro-batch	8 (4 for on-policy teacher arms)
optimizer	AdamW, lr 3×10^{-4}, no weight decay; 100 linear warm-up steps, then constant
gradient clipping / dropout	1.0 / 0
precision	student fp32 + bf16 autocast; teacher, oracle bf16 frozen
retokenization	all-different: keeps 32% of tokens, 1.71$\times$ length, $\Delta(1)$=1.003
evaluation epochs	0, 0.25, 0.5, 1–6, 8, 10, …, 20
saved checkpoints	epochs 4, 12, 20
reference divergences	$\mathrm{KL}_{\mathrm{seq}}(\mu\|\pi_t) = 17.4 \pm 0.4$; $\mathrm{KL}_{\mathrm{seq}}(\mu\|\pi_s^{(0)}) = 23.1 \pm 0.5$

Table 7: Configuration of the T5 distillation line. $\pi_s^{(0)}$ is the SFT student every arm starts from; $\Delta(1)$ is the unseen token material one retokenized surface adds, in units of one canonical prompt, measured on the distillation pool.

Qwen3 line	
oracle μ	`Qwen3-4B`, as released
teacher π_t	`Qwen3-1.7B`, as released
student π_s	`Qwen3-0.6B-Base`, SFT on oracle samples
corpus	Tulu 3 SFT mixture, first user turn
prompt	chat template, thinking disabled
prompt / response length	$\leq$ 512 / $\leq$ 512 tokens
prompt pools (seed 0)	25,000 oracle, 500 SFT-val, 10,000 distill, 500 held-out
sampling	temperature 1, no truncation
teacher SFT	none
student SFT	NLL on the 25,000 oracle pairs; AdamW, warm-up 100; lr 2×10^{-5}, 3 ep., batch 16, best epoch 0
distillation loss	token-level forward KL, teacher-forced
schedule	20 epochs $\times$ 313 steps, batch 32, 6,260 updates
micro-batch	8
optimizer	AdamW, lr 2×10^{-5}, no weight decay; 100 linear warm-up steps, then constant
gradient clipping / dropout	1.0 / 0
precision	student fp32 + bf16 autocast; teacher, oracle bf16 frozen
retokenization	StochasTok α=1.0475: 1.66$\times$ length, $\Delta(1)$=1.000
evaluation epochs	0, 0.25, 0.5, 1–6, 8, 10, …, 20
saved checkpoints	epochs 2, 4, 8, 12, 20
reference divergences	$\mathrm{KL}_{\mathrm{seq}}(\mu\|\pi_t) = 140.4 \pm 5.0$; $\mathrm{KL}_{\mathrm{seq}}(\mu\|\pi_s^{(0)}) = 161.2 \pm 6.4$

Table 8: Configuration of the Qwen3 distillation line; notation as in Table 7.

Roles of the models. The T5 line follows the original setup: the oracle generates one response per prompt of the oracle pool, teacher and student are supervised-fine-tuned on those pairs, and the best epoch is chosen on the SFT-validation prompts (epoch 0 for both). The teacher is thus an imperfect fit to the oracle, and the oracle divergence of a perfect student would be near zero. The Qwen3 line instead uses two released post-trained models as oracle and teacher, related only by family and tokenizer, and fine-tunes only the student initialization on the oracle's samples, again keeping epoch 0. A student that matched the teacher exactly would then sit at $\mathrm{KL}_{\mathrm{seq}}(\mu\|\pi_t)$ on the oracle axis, so the oracle divergence of every Qwen3 arm rises from its starting value simply because distillation is working, and teacher hacking is read from the offline-minus-online contrast (Table 4) rather than from the slope of one curve. The floor is unchanged to 0.3% when the teacher is evaluated in fp32.

Objective and data sources. Every arm minimizes the per-token forward divergence $\mathrm{KL}\big(\pi_t(\cdot \mid x, y_{<j}) \,\|\, \pi_s(\cdot \mid x', y_{<j})\big)$, teacher-forced on a response y and token-weighted over the batch, where the teacher reads the canonical prompt x and the student its own copy x', canonical or retokenized. Responses are never retokenized. For T5 the prompt is the encoder input, so teacher and student decoder logits align position by position; for Qwen3 the loss is taken over the response slice of the concatenated stream, and the two prompts may differ in length. The response comes from one of three sources: *offline*, a dataset of teacher responses sampled once and replayed in a fresh order each epoch; *on-policy teacher*, a fresh teacher response for every batch; *on-policy student*, a response sampled from the current student (T5 only). All sampling is at temperature 1 without truncation; the Qwen3 oracle and teacher are sampled with thinking disabled, and about half of the oracle's Tulu 3 responses reach the 512-token cap.

Learning rate. The T5 learning rate is the original one. It does not carry over to a 0.6B decoder-only model under AdamW, so the Qwen3 rate was chosen by a one-epoch sweep on the high-diversity offline arm over $\{10^{-5}, 2{\times}10^{-5}, 5{\times}10^{-5}, 10^{-4}\}$, keeping the largest rate whose training loss was monotone and whose oracle divergence after one epoch did not exceed the SFT student's; $2{\times}10^{-5}$ was kept. A micro-batch of 16 exceeded 80 GB in the KL computation, so all arms use 8.

Prompt accounting. Three counts describe every dataset: *source texts*, the number of distinct documents; *prompt variants*, the number of distinct prompt token sequences; and *effective prompts*, the unseen token material the student sees in units of one canonical prompt, deliberately not capped at the variant count so that the experiment rather than the metric decides whether a surface is a prompt. The material one retokenization contributes, $\Delta(1)$, is measured once per tokenizer and corpus on prompts of the distillation pool (60 prompts of mean length 511 tokens for T5, 200 of mean length 159 for Qwen3) as the fraction of token occurrences absent from the canonical encoding, normalized by canonical length; repeated draws of one text saturate as $\Delta(k) = A(1 - e^{-k\nu/A})$, with ceilings $A = 2.0$ (T5, all-different) and $A = 6.1$ (Qwen3, StochasTok). Every matched pair follows one rule: halve the source texts, give each text one retokenized twin, keep the generations per surface. Offline arms keep 10,000 rows by drawing 1, 2 and 5 teacher responses per surface at the three levels; online arms cycle over their surfaces with fresh responses. The rule is exact only if one twin is worth one canonical prompt. On the T5 vocabulary the all-different scheme gives $\Delta(1) = 1.003$, so the T5 pairs match to 0.2%. On the Qwen3 byte-level vocabulary the same scheme overshoots to $\Delta(1) = 1.24$, and no twin count can repair a 12% mismatch at a fixed row budget, so the Qwen3 line uses the official StochasTok sampler at the depth $\alpha = 1.0475$ where $\Delta(1) = 1.000$ exactly; its prompt inflation is essentially that of all-different on this corpus, so the perturbation is no milder. Retokenized surfaces are drawn once with seed 0 and frozen for the run, offline and online alike; the redraw controls of Appendix C are the only arms that resample them, once per epoch.

Retokenization schemes. *All-different* (T5): for each whitespace-delimited word, the word's own canonical pieces are forbidden, the segmentations of its byte string over the remaining vocabulary are enumerated by dynamic programming, the piece count nearest canonical is taken and one segmentation is drawn uniformly among those; words with no such segmentation are kept, which on XSum is the 17% of words, mostly short function words, that leave 32% of token occurrences intact. *StochasTok* (Qwen3): the released sampler of Sims et al. (2025), which makes $\lfloor \alpha m \rfloor$ attempts on an m-token prompt to replace a token by a pair of vocabulary tokens that concatenate to it, with the split table built from the Qwen3 vocabulary and every added token excluded. Both samplers leave the instruction prefix, the chat template and the response canonical, and both are verified to preserve the byte string exactly (for Qwen3 after the tokenizer's NFC normalization).

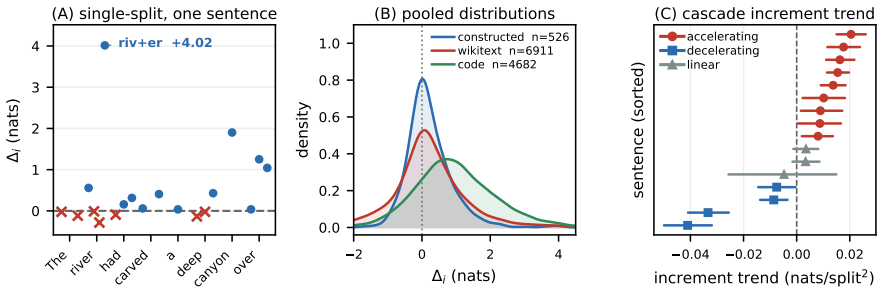

Figure 8: Retokenization sensitivity under OLMo-2-1124-7B base. **(A)** Single-split sensitivity Δ_i by position, for one test sentence ("The river had carved a deep canyon over"). **(B)** Pooled Δ_i distributions for constructed sentences vs. WikiText and code corpus-text replications (legend gives n positions per domain; not normalized). **(C)** Trend in the per-split increment of $\mu(d)$ for the 16 cascade-analyzed sentences (OLS slope of the increment against depth, nats/split^2; 95% bootstrap intervals over paths). Positive values accelerate, negative decelerate; intervals containing zero are indistinguishable from linear.

Metrics. The divergence to the teacher is the sequence-level forward KL $\sum_j \mathrm{KL}\big(\pi_t(\cdot \mid x, y_{<j}) \,\|\, \pi_s(\cdot \mid x, y_{<j})\big)$, teacher-forced on one teacher sample y per held-out prompt and averaged over the 500 prompts, both models reading the canonical prompt; the divergence to the oracle replaces the teacher and its sample by the oracle and one oracle sample. The training curves report the per-token loss on the rows of the current step. Reverse-KL and Jensen–Shannon variants on student samples were recorded alongside and agree in ordering. Invariance is measured by retokenizing the student's held-out prompt (StochasTok at several depths and all-different, the same perturbed prompts for every arm and checkpoint) while the reference model keeps the canonical prompt. Standard errors are over prompts.

Compute. The T5 arms ran one at a time on an H100 shared with other jobs, at roughly 4.5 h per offline arm and about twice that for the online arms; the Qwen3 arms ran one per dedicated 80 GB GPU, four at a time, at 8–22 h per arm with the online arms generation-bound. Benchmarks for the Qwen3 students use `lm-evaluation-harness` (MMLU 5-shot, ARC-challenge 25-shot, HellaSwag 10-shot, IFEval 0-shot) with the chat template and thinking disabled.

E Splittability Analysis

Feature regression. We regress the single-split sensitivity on seven features of the split position, together with sentence fixed effects, by ordinary least squares with standard errors clustered by sentence; features and outcome are standardized over the 526 positions, so coefficients read in standard deviations. The features are the Zipf log-frequency of the canonical piece being split, the lower and the higher Zipf frequency of its two halves, the cut offset normalized by piece length, the byte length of the piece, an indicator that one half is the bare space marker, and an indicator that the piece belongs to a word already split into more than one piece. We fit the signed sensitivity and its magnitude separately. Sentence identity alone explains 0.179 of the variance in the signed sensitivity and 0.167 in its magnitude; the seven features raise these to 0.210 and 0.232. For the magnitude, the cut position, the rarity of the canonical piece and its byte length lead on drop-one explained variance at 0.025, 0.021 and 0.013, comparable to one another and small in absolute terms, while every share collapses for the signed outcome.

Corpus replication. WikiText prose closely matches the constructed set under the same per-token probability band (median +0.13 against +0.12, negative fraction 0.37 against 0.376, Gini 0.52 against 0.496 on 415 constructed positions), while code shifts toward higher median sensitivity (+0.29), fewer negative positions (0.31), and heavier concentration (Figure 8B). The constructed statistics quoted in Section E.1 are unfiltered.

E.1 Characterizing sensitivity to retokenization

Existing evaluations of LLM sensitivity to retokenization report pass/fail performance on reasoning benchmarks (Jain et al., 2026; Zheng et al., 2025; Geh et al., 2025). These discard the gradation between a continuation that barely survives retokenization and one that is destroyed (Schaeffer et al., 2023). We instead measure the target NLL f_{read} of Section 2.1, which is defined for any text and any target continuation, needs no checkable task, and is deterministic under teacher forcing. Because f_{read} is additive over target positions, per-split sensitivities and their accumulation are well-defined. The same functional serves as the measurement instrument for the rest of the paper. Section 6.2 uses this sweep to quantify what fine-tuning removes, and Section 6.1 reports exposure depth in its units. We find that individual splits frequently reduce f_{read}, while the NLL accumulates over cascades in a strongly sentence-dependent way. All measurements use `allenai/OLMo-2-1124-7B` (Team OLMo et al., 2024) in float32[2]. We measure distance from canonical by token count, $d(\mathbf{t}) = |\mathbf{t}| - |\mathbf{t}_c|$, and in this section consider only splits.

Sensitivity to single splits. We define the single-split sensitivity of the target NLL. For a token sequence $\mathbf{t} = (t_1, \ldots, t_k)$, a *split position* is a pair (i, c) such that token t_i admits a two-piece decomposition $t_i = uv$ at cut point c with $u, v \in V$. Applying the split yields

$$\mathbf{t}^{(i,c)} = (t_1, \ldots, t_{i-1}, u, v, t_{i+1}, \ldots, t_k) \in \mathcal{T}(\mathbf{x}), \tag{12}$$

and we write

$$\Delta_{i,c}(\mathbf{t}) := f_{\text{read}}\big(\mathbf{t}^{(i,c)}\big) - f_{\text{read}}(\mathbf{t}). \tag{13}$$

Throughout this section we measure splits of the canonical tokenization only and abbreviate $\Delta_{i,c} := \Delta_{i,c}(\mathbf{t}_c)$; where the cut point is clear from context we write Δ_i. We sweep every valid split position of $\mathbf{t}_c$ across twenty constructed sentences (526 positions) and replicate the results on corpus-sampled text; experimental details are deferred to Appendix D.

A single split moves the target NLL very little. The pooled median of Δ_i is $+0.11$ nats, and 39.2% of positions are negative: roughly two times in five, splitting a token makes the target continuation *more* likely.

Concentration. Positive sensitivity is concentrated in few positions: restricted to positions with $\Delta_i > 0$, the pooled Gini coefficient is 0.507, and the top decile of positions accounts for a third of the total. The largest single value occurs in "The river had carved a deep canyon over," where splitting `river` into `riv` and `er` raises the target NLL by 4.02 nats (Figure 8A), a 5.8σ residual under the signed regression of Appendix E.

Feature regression. Regressing Δ_i on seven split-level features and sentence identity, the features add little over sentence identity alone, and none predicts the sign (Appendix E).

Repeating the sweep on corpus-sampled text preserves these conclusions, with WikiText prose closely matching the constructed set and code shifting toward higher median sensitivity and heavier concentration (Appendix E).

Cascades. A *cascade* is a sequence of tokenizations starting at $\mathbf{t}_c$ in which each step splits one token of the previous one, so that d increases by one at every step and no split is ever undone. We sample 50 cascade paths to depth 20 for each of the twenty sentences; sixteen fall within the per-token probability band used throughout and are analyzed here.

We find that different cascades starting from the *same* sentence share a depth-dependent drift $\mu(d) = \mathbb{E}[f_{\text{read}}(\mathbf{t}) \mid d(\mathbf{t}) = d]$, where the expectation is taken over cascade paths from that sentence.[3] In every sentence the target NLL rises with depth, but how it rises differs: fitting a linear trend to the per-split increments $\mu(d) - \mu(d-1)$, nine sentences accelerate, four decelerate, and three are indistinguishable from linear accumulation (Figure 8C).

[2] All measurements in this section use a base model. Appendix E.2 reports the same sweep on post-trained checkpoints.

[3] Among the cascades from a single sentence, depth explains 71% of variance in f_{read} against 12% for path identity, and the mean path-to-mean correlation is 0.93.

E.2 Supervised Fine-tuning Reorganizes Sensitivity

Partial invariance of LLMs to retokenized input emerges during pre-training (Jain et al., 2026), but the format-enforced distinction between prompt and response provided by supervised fine-tuning (SFT) is necessary for the model to respond fluently to non-canonical prompts (Zheng et al., 2025). Repeating the sensitivity experiments above on `OLMo-2-1124-7B-Instruct` (Team OLMo et al., 2024), Olmo-3 (Team Olmo et al., 2025), and Tulu-3-8B (Lambert et al., 2024), post-training reorganizes which positions are sensitive, and this is concentrated at base-to-SFT in all three families, with later stages largely inheriting the result. How much reorganization occurs varies by family, from about 10 percent of the base profile retained in Olmo-3 to 30 to 44 percent in OLMo-2 and Tulu-3. The distribution of sensitivity has no common pattern: it is unchanged in OLMo-2, shifts upward in magnitude in Olmo-3, and shifts downward in Tulu-3, where the negative fraction rises from 0.27 to 0.47 at the DPO (Rafailov et al., 2023) stage. Only the concentration of the positive tail is stable everywhere, with Gini between 0.48 and 0.51 at every checkpoint of every family.

Doing fine-tuning via SFT on aligned models can degrade alignment due to catastrophic forgetting (McCloskey & Cohen, 1989; Luo et al., 2023). Retokenized alignment replicates this phenomenon: SFT training the Instruct model on the Alpaca dataset (Taori et al., 2023) increased the attack success rate (ASR) of AdvTok from 0.565 $[0.495, 0.635]$ to 0.640 $[0.575, 0.705]$ at greedy decoding and from 0.584 $[0.536, 0.633]$ to 0.667 $[0.627, 0.707]$ at t=1. However, we find that using retokenized instructions during training, and keeping all else identical, preserves ASR at its original level.

F Cross-domain sensitivity

Text domains. We repeat the sensitivity sweep of Appendix E on four held-out prompt sets (200 prompts, roughly 3,000 retokenized variants each) and report the fraction of the untrained model's median sensitivity that each arm removes (a dimensionless fraction; the medians themselves are in nats). Output KL removes 0.73 on held-out instructions from the training domain and 0.34–0.41 on WikiText prose, code, and grade-school arithmetic; the reduction roughly halves off domain and is uniform across the three. Canonical supervised fine-tuning removes 0.49 in domain while leaving code no better than no training at all (3.06 against 2.95 nats, intervals overlapping), so the instrument registers domain-specific damage where it exists. Under full character-level fragmentation the untrained model loses between 7.5 nats on instructions and 55 on WikiText, and output KL halves the loss in every domain.

Benign prompts under noise. Refusal on perturbed benign prompts is unchanged: at eight substitutions output KL sits at $+0.000$ $[-0.020, +0.020]$ against the untrained model on held-out instructions ($n = 200$) and $+0.064$ $[0.032, 0.100]$ on safe XSTest prompts ($n = 250$). Answer overlap degrades identically in every arm, trained or not, losing 0.20–0.25 of its unigram F1 under perturbation.

G CVaR

Setting $\alpha = 1$ removes the selection and backpropagates a uniform mean over all nine encodings. The effect depends on the objective. For output KL, the selection lowers greedy attack success from 0.150 $[0.105, 0.200]$ to 0.085 $[0.050, 0.125]$ (paired difference -0.065 $[-0.120, -0.015]$) and sampled attack success from 0.341 $[0.306, 0.377]$ to 0.212 $[0.184, 0.243]$ (paired -0.129 $[-0.161, -0.096]$). For token-level cross-entropy it changes nothing, with 0.170 against 0.170 at greedy decoding (paired $+0.000$ $[-0.040, +0.040]$) and 0.325 against 0.314 under sampling (paired $+0.011$ $[-0.022, +0.044]$). Only $\alpha \in \{0.25, 1\}$ was trained, so we report two points rather than a curve.

H Exposure baselines

Consistency training is one of several ways to expose a model to non-canonical tokenizations during fine-tuning. This appendix reports the alternatives we trained, all on `OLMo-2-1124-7B-Instruct` and all evaluated with a fresh AdvTok search against each model over the same 200 held-out AdvBench prompts, under the protocol of Table 1, with responses judged by Llama-Guard-3-8B.

The arms differ along three axes that the table groups by, because conflating them is misleading. The first is the *training mixture*: benign prompts alone, or the equal benign/harmful mixture used by our objective. The second is the *supervision target*: the dataset's own answers, or continuations generated by the frozen reference. The third is *exposure depth*, the mean inflation in token count relative to canonical, which varies by an order of magnitude across samplers.

	depth	ASR greedy	ASR t=1
No fine-tuning	—	0.565	0.584
Benign prompts, dataset answers			
Canonical SFT	1.00×	0.640	0.667
Prompt-retokenized SFT	1.30×	0.595	0.651
Prompt-retokenized SFT	3.10×	0.580	0.591
Benign prompts, reference continuations			
Consistency, output KL	3.62×	0.560	0.598
Mixed prompts, reference continuations			
Multi-tokenization LoRA, identical copies	1.00×	0.645	0.669
Multi-tokenization LoRA, length-ratio buckets	2.24×	0.370	0.476
Multi-tokenization LoRA, uniform lattice	3.19×	0.285	0.413
StochasTok, $\alpha = 0.3$	1.28×	0.420	0.499
Consistency, output KL	1.28×	0.395	—
Consistency, output KL	3.62×	0.085	0.212
Consistency, representation, all layers	3.62×	**0.025**	**0.058**

Table 9: Exposure baselines. Depth is mean token-count inflation relative to the canonical tokenization. The LoRA arms follow the configuration of Ghosh & Jyothi (2026) (rank 16 on query, key and value projections, learning rate 2.5×10^{-6}, 30 copies per example) applied to our training mixture rather than to their task data, so they test their sampler within our setup and are not a reproduction of their experiment. The adapters retokenize the prompt span only; the treatment of the answer span in Ghosh & Jyothi (2026) is unspecified and is immaterial for their single-token answers.

Four observations follow, in decreasing order of how well the data supports them.

LoRA arms. Against the budget-matched control, which sees 30 identical canonical copies of each example, the two retokenized LoRA arms are better by 0.360 $[0.280, 0.440]$ and 0.275 $[0.195, 0.355]$ at greedy decoding, with intervals far from zero. The mechanism reported by Ghosh & Jyothi (2026) therefore replicates here, on an adversarial evaluation their work does not perform. Within that family, the uniform-lattice sampler outperforms the length-ratio buckets (0.285 against 0.370), reversing their ordering on task accuracy; the bucketed scheme caps at $3.0\times$ by construction and realizes $2.24\times$, against $3.19\times$ for uniform sampling.

Fine-tuning without retokenization raises attack success. Three independent arms show it: full fine-tuning on Alpaca answers (0.667 at t=1), the LoRA control on the mixed data (0.669), and the self-distillation control of Section 6.1. All three exceed the untrained model significantly under sampled decoding, with canonical attack success and refusal rates moving in the same direction, and all three are at the boundary of significance at greedy decoding.

Depth and objective. At matched depth of roughly $3.1\times$, retokenizing prompts inside ordinary SFT leaves attack success unchanged from the untrained model (0.580 against 0.565, 0.591 against 0.584). At matched objective, our consistency loss applied at $1.28\times$ depth reaches 0.395, no better than StochasTok at the same depth. Only the combination reaches 0.085.

Harmful prompts. The same objective trained on benign prompts alone is indistinguishable from the untrained model (0.560 against 0.565). Since the objective uses no refusal labels, and the reference already refuses harmful prompts canonically, what the harmful prompts supply is the opportunity to practice preserving that behavior under retokenization.

	AdvTok		BEAST		Charmer	
supervised layers	$T = 0$	$T = 1$	$T = 0$	$T = 1$	$T = 0$	$T = 1$
$\{32\}$	$0.120 \pm .045$	$0.164 \pm .037$	$0.515 \pm .070$	$0.497 \pm .053$	—	—
$\{8, 16, 24, 32\}$	$0.130 \pm .047$	$0.129 \pm .032$	$0.360 \pm .068$	$0.357 \pm .051$	$0.245 \pm .057$	$0.203 \pm .042$
all 32	$0.025 \pm .023$	$0.058 \pm .018$	$0.300 \pm .062$	$0.288 \pm .050$	$0.170 \pm .053$	$0.118 \pm .032$
No fine-tuning	$0.565 \pm .070$	$0.584 \pm .048$	$0.580 \pm .068$	$0.477 \pm .054$	$0.305 \pm .062$	$0.308 \pm .051$

Table 10: Single-position representation matching as the supervised layer set grows, on the 200 held-out AdvBench prompts under the settings of Appendix K. The BEAST rate falls monotonically with the number of supervised depths and the Charmer rate follows; the layer-32 Charmer run was not performed. Adding layers costs nothing at training time, since the hidden states are already materialized by the forward pass.

I THE STOCHASTOK REVERSAL

Applying StochasTok to the full sequence, as proposed for pretraining, retokenizes the instruction and the answer at the same rate and trains on the retokenized answer tokens; only the chat scaffolding stays canonical. We train six such arms at increasing α on our data mixture and evaluate each with the protocol of Section 6.1 (Table 11). Attack success falls monotonically through $\alpha = 2.7$ and reverses at $\alpha = 5.5$, exactly where canonical behavior degrades: the $\alpha = 5.5$ arm complies with 0.345 of harmful prompts presented canonically and its canonical refusal falls to 0.445 from the 0.79–0.90 band of the shallower arms. Its apparent mid-sweep robustness therefore reflects degraded generation rather than preserved refusal. Sampling at $T{=}1$ shows the same shape, with the reversal arm at 0.575 adversarial attack success and 0.376 canonical compliance. The $\alpha = 0.3$ arm of this sweep is the StochasTok baseline of Table 1; the context-only arm of Appendix H, which draws StochasTok-sampled encodings of the prompt under the consistency objective and leaves the continuation canonical, sits at 0.395 greedy against this sweep's 0.420, so at StochasTok's depth the objective adds little and depth is the binding constraint.

α	Train inflation	Adv. ASR	Canonical unsafe	Canonical refusal
0.0	$1.000\times$	0.540	0.050	0.855
0.1	$1.087\times$	0.605	0.055	0.855
0.3	$1.243\times$	0.420	0.020	0.865
1.7	$2.074\times$	0.270	0.045	0.900
2.7	$2.495\times$	0.190	0.075	0.790
5.5	$3.309\times$	0.505	0.345	0.445

Table 11: Full-sequence StochasTok at increasing α, greedy decoding, 200 held-out AdvBench prompts, judged by Llama-Guard-3-8B. Attack success falls through $\alpha = 2.7$ and reverses at $\alpha = 5.5$ as canonical refusal collapses.

J OVER-REFUSAL

Table 12 reports refusal rates on canonical inputs for all seventeen arms. AdvBench refusal is measured on the 200 held-out harmful prompts of Section 6.1, benign refusal on 100 held-out instructions, and XSTest-safe refusal on the 250 safe prompts of XSTest (Röttger et al., 2024), which superficially resemble unsafe requests. AdvBench responses are decoded greedily from a bare user turn; benign and XSTest-safe responses are sampled with the model's chat template, four per prompt at temperature 1 with 192 max new tokens, each prompt contributing the mean refusal over its four samples. Refusal is detected with the keyword-prefix judge, whose precision and arm-dependent recall are characterized in Appendix L; intervals are half-widths of 95% cluster bootstraps over prompts (5,000 resamples).

The mixed-trained arms leave benign refusal statistically unchanged and move XSTest-safe refusal by at most $+0.061$ absolute (the 8-token prefix arm; the output-KL and continuation-supervised arms sit at $+0.005$ to $+0.046$). Single-sided mixtures deform the refusal boundary instead of extending it.

The harmful-only arm converts its 0.005 attack success into indiscriminate refusal, declining 0.465 of benign instructions and 0.614 of XSTest-safe prompts. The benign-only arm records the lowest AdvBench refusal (0.745).

Model	AdvBench refusal (canonical, n=200)	Benign refusal (n=100)	XSTest-safe refusal (n=250)
No fine-tuning	0.840 ± 0.052	0.030 ± 0.029	0.065 ± 0.024
Self-distillation, canonical only	0.880 ± 0.045	0.022 ± 0.022	0.102 ± 0.029
Output, cross-entropy	0.900 ± 0.042	0.037 ± 0.029	0.081 ± 0.024
Output, KL	0.930 ± 0.035	0.028 ± 0.021	0.097 ± 0.027
Output KL, benign-only	0.745 ± 0.060	0.025 ± 0.020	0.045 ± 0.017
Output KL, harmful-only	0.995 ± 0.008	0.465 ± 0.065	0.614 ± 0.040
Output KL, final prompt position	0.940 ± 0.032	0.020 ± 0.024	0.057 ± 0.020
Representation, single-position, final layer	0.915 ± 0.037	0.035 ± 0.031	0.090 ± 0.031
Representation, single-position, multi-layer	0.845 ± 0.052	0.033 ± 0.031	0.048 ± 0.021
Representation, single-position, all layers	0.905 ± 0.040	0.030 ± 0.030	0.079 ± 0.030
Representation, continuation-supervised, final layer	0.900 ± 0.042	0.025 ± 0.029	0.111 ± 0.031
Representation, continuation-supervised, multi-layer	0.910 ± 0.040	0.018 ± 0.020	0.070 ± 0.025
Output KL, 8-token prefix	0.915 ± 0.037	0.015 ± 0.015	0.126 ± 0.031
Multi-tokenization LoRA, identical copies	0.820 ± 0.052	0.022 ± 0.026	0.086 ± 0.025
Multi-tokenization LoRA, length-ratio	0.860 ± 0.047	0.033 ± 0.026	0.095 ± 0.027
Multi-tokenization LoRA, uniform	0.850 ± 0.047	0.015 ± 0.014	0.090 ± 0.026
StochasTok, α=0.3	0.865 ± 0.048	0.028 ± 0.024	0.108 ± 0.029

Table 12: Refusal on canonical inputs for all seventeen arms. AdvBench refusal is measured on the 200 held-out harmful prompts of Section 6.1, benign refusal on 100 held-out instructions, and XSTest-safe refusal on the 250 safe prompts of XSTest (Röttger et al., 2024), written to resemble unsafe requests. AdvBench responses are decoded greedily from a bare user turn with no system prompt; benign and XSTest-safe responses are sampled with the chat template, four per prompt at temperature 1, scored as the per-prompt mean. Refusal is detected by the keyword-prefix judge characterized in Appendix L. Values are the rate $\pm$ half the width of a 95% cluster bootstrap over prompts (5,000 resamples).

K ATTACK SETTINGS

Every attack in Table 1 is the authors' reference implementation at its published settings, run separately against each model, so each model faces inputs found against itself. The prompts are the same 200 held-out AdvBench requests for every attack and every arm, presented as a bare user turn with no system prompt.

AdvTok (Geh et al., 2025) runs 100 greedy search iterations from the canonical encoding, sampling 8 tokenizations per iteration from the segmentation lattice. Responses are 256 new tokens, decoded greedily or sampled eight times at T=1.

BEAST (Sadasivan et al., 2024) appends an adversarial suffix of 40 tokens found by beam search with k_1=k_2=15.

Charmer (Abad Rocamora et al., 2024) edits the prompt itself: up to k=20 character substitutions chosen over 5 candidate positions per step for 50 steps.

BEAST and Charmer are scored under a separate pipeline from AdvTok: responses are 128 new tokens, decoded greedily or sampled three times at T=1. All three are judged unsafe by Llama-Guard-3-8B (Appendix L); the reported rate is the fraction of prompts judged unsafe at T=0 and the fraction of samples at T=1, with a 95% cluster bootstrap over prompts. Search budgets differ by attack, so rates are comparable within a column of Table 1 and not across attacks. The Repr. arm of that table supervises the residual stream at the final prompt position at every layer.

L JUDGES

Attack success is judged by Llama-Guard-3-8B. On 56 hand-labeled responses the judge's precision on unsafe verdicts is 0.696 $[0.567, 0.801]$ pooled (39 harmful compliance; 9 off-request, 7 refusal, 1 degenerate), with per-arm precision between 0.545 and 0.800 and all intervals overlapping, so the contamination is arm-independent as far as we can measure. Attack success orderings between arms are unaffected; absolute rates are inflated by roughly 30%. We report rates as measured. Labels are from a single annotator.

Refusal is detected by a fixed keyword-prefix matcher applied to the first 200 characters of the lowercased response. Its precision is high on a hand-labeled sample ($5/5$; the false-positive rate is otherwise unmeasured). Its recall varies by arm, with missed-refusal rates between 0.070 and 0.305 at eight character substitutions, highest exactly where reported refusal is lowest, so refusal gaps between arms are partly instrument-driven and we lean on orderings rather than magnitudes.

M PROBES

The probes of Section 7 are fit once, on the untrained model's canonical activations, and applied to every arm without refitting. Table 14 reports the complementary instrument. For each arm we fit a fresh logistic probe on that arm's own canonical activations at layer 32, with the same prompt-level 70/30 split and seed, and evaluate it on canonical tokenizations, on constructed one- and two-split edits, on tokenizations drawn uniformly from the MDD, and on the attack encodings of the shared condition.

Two facts justify the frozen instrument in the main text. Refitting saturates on trained arms, with seven of ten rows at 0.95 or above under attack, so comparisons that separate the arms under frozen transfer, such as 0.968 against 0.698 at matched layer and condition in Figure 3, compress here to 1.000 against 0.952. Refitting also recovers nothing on the untrained model, which reads 0.397 under attack with either instrument; the deletion of Section 7 is therefore a genuine absence of the signal rather than a rotation a fixed probe would miss. Where the instruments disagree the gap is modest and layer-specific: the single-position KL arm reads 0.952 refit against 0.873 frozen under the shared attack at layer 32, and identically at layers 16 and 24. Accuracy falls from the split edits to the deeper uniform MDD samples for the untrained model, canonical SFT, and the benign-only arm, while every mixed-trained arm holds ceiling on both.

Drift and crack rate. Pooling $1{,}600$ prompt-arm pairs, the crack rate rises monotonically with layer-32 residual drift within every arm, from 0.02–0.07 below drift 0.01 to 0.33–1.0 above 0.3. The arms differ chiefly in where their drift mass sits: the mixed-trained arms place 150–166 prompts in the lowest bin, while the untrained model places none below 0.03 and 142 in $[0.3, 1.0)$. There is no common absolute threshold; the untrained model cracks *less* than treated arms at matched drift in the middle bins, so tolerance scales with an arm's typical drift rather than with drift itself. Consistency training compresses the drift of harmful-prompt attack encodings into the regime where the original readout still fires. We report binned rates rather than a fitted arm effect because four of the six cells above drift 0.3 contain $n \leq 8$ pairs and arm is nearly collinear with bin.

Prospective predictors. Neither the model's refusal margin at the canonical tokenization nor the frozen probe's margin there predicts attack outcome (Table 13). The reading-sensitivity summaries of Section E.1 predict weakly and are subsumed by the probe margin under the attacked tokenization, which predicts strongly but is measured on the encoding the search selected. It confirms the deletion mechanism per prompt; it does not forecast vulnerability.

Directed searches. This is what its objective requires. Fluent compliance presupposes that the instruction is still read, so the search is constrained to preserve comprehension while it is free, and pressured, to destroy the feature that refusal depends on. The probes confirm both halves of the selection at the representation level. On the prompts that crack, the dissociation is sharpest: safety reads 0.167 while topic reads 0.583. A search directed at the topic probe deletes the linear topic readout from the same site on 0.735 of the prompts it reads correctly at canonical ($36/49$), at $1.78\times$ inflation against the compliance attack's $2.69\times$. The safety margin stays positive on 0.905 of all 63 encodings, and refusal declines from 0.730 to 0.587 (paired -0.143 $[-0.270, -0.016]$), indistinguishable from the depth-matched control (-0.048 $[-0.175, +0.079]$). Each search removes

Predictor of attack outcome	AUC
Refusal margin, canonical	0.485
Probe margin, canonical	0.507–0.532
Δ_i summaries (median, spread, worst, CVaR-25%)	0.62–0.64
Probe margin, attacked tokenization (post hoc)	0.85
Joint over probe margin alone	+0.001

Table 13: Predicting per-prompt attack outcome on the untrained model. The post-hoc predictor is measured on the encoding the search selected.

Model	Canonical	Splits	MDD-uniform	Adversarial
No fine-tuning	1.000	0.987	0.812	0.397
Canonical SFT	0.992	0.979	0.825	0.429
Output, KL	1.000	1.000	1.000	1.000
Output, cross-entropy	1.000	1.000	0.998	1.000
Output KL, benign-only	1.000	0.996	0.885	0.667
Output KL, final prompt position	1.000	1.000	0.990	0.952
Representation, single-position, final layer	1.000	1.000	1.000	1.000
Representation, single-position, multi-layer	1.000	1.000	0.997	0.984
Representation, continuation-supervised, final layer	1.000	1.000	0.998	1.000
Representation, continuation-supervised, multi-layer	1.000	1.000	0.998	1.000

Table 14: Refit-per-arm probe accuracy at layer 32. A logistic probe is fit for each arm on that arm's own canonical activations, with the prompt-level split of Section 7, and evaluated on canonical tokenizations, on constructed split edits, on uniform MDD retokenizations, and on the attack encodings selected against the untrained model.

the feature its objective targets; the untargeted readout stays near its depth-matched baseline in both cases.

Probe protocol. Both probes use a prompt-level split, so no prompt contributes tokenizations to both halves, and both are frozen thereafter. A two-layer probe is fit identically, to distinguish information that is absent from information that is present but not linearly decodable. We evaluate two adversarial conditions. Under the *shared* condition every model receives the tokenizations selected by the attack against the untrained model, which asks how each model represents one fixed attack. Under the *own* condition each model receives the tokenizations selected against itself, the condition matching its reported attack success. Probes are read at layers $\{8, 16, 24, 32\}$; we report layer 32 unless stated. Refitting a probe per model saturates and conceals the differences between arms: the final-prompt-position arm reads 0.952 under a refit probe and 0.873 under the frozen one, on the shared condition at layer 32.

Steering controls. A random direction of matched norm restores nothing (0.73–0.85 unsafe at every scale), and the effect is dose-dependent (0.150, 0.761, 1.000 refusal at $0.5\times$, $1\times$, $2\times$). The direction itself is not prompt-specific: at the calibrated scale it also induces refusal on 0.550 of benign prompts, consistent with the indiscriminate refusal steering of Arditi et al. (2024), so the intervention demonstrates mechanism rather than supplying a defense.