

Inference-Time Consensus for Mitigating Hidden Behaviors from LLM Fine-Tuning

Adhyayan Narang*, Artin Tajdini*, Claire Zhang, Jamie Morgenstern
University of Washington

Abstract

Recent work shows that fine-tuning language models on even a small amount of poisoned data can install targeted misbehavior, and ostensibly benign data can transmit hidden preferences that generalize broadly. Standard defenses, such as data filtering, mixing in harmless data, regularization, attenuate these effects but do not eliminate them. We instead pursue robustness through redundancy: collecting multiple datasets from different sources and only learning what is common between them. Thus, if only a subset of sources are malicious, the misbehavior will be blocked. In order to implement this defense strategy, we fine-tune a separate reference model on each source’s dataset and aggregate their next-token distributions at decoding time. We introduce two consensus decoders: a token-wise minimum, which caps each token at the lowest probability any source assigns, and a base-relative variant, which reverts to the base probability on any token the sources move in opposing directions. We further relax exact agreement to tolerate partial support across sources and different surface expressions of the same intention. Across controlled poisoning tasks, subliminal learning, and emergent misalignment, consensus decoding suppresses source-specific misbehavior while preserving shared desirable behavior, including cases where union training and weight averaging retain the unwanted behavior.¹

1 Introduction

A fine-tuning corpus for a language model often combines data from multiple external sources (Lambert et al., 2025; Wan et al., 2023). A lab adapting a model may procure instruction data from several vendors, license domain corpora from different partners, or accept task datasets from independent contributors. Such data can teach useful behavior, but some sources may be controlled by parties with objectives beyond the requested capability (Qi et al., 2024), and fine-tuning on even a small number of poisoned examples can reliably insert malicious behavior into language models (Wan et al., 2023; Souly et al., 2025). The concern is no longer limited to visible triggers. Work on emergent misalignment and subliminal learning shows that fine-tuning can transmit unwanted behavior that is not apparent from the datasets’ semantic content (Betley et al., 2026, 2025; Cloud et al., 2026; Aden-Ali et al., 2026), so a source can shift a model’s preferences or persona through data that passes manual and automated inspection.

The main current defenses against fine-tuning data poisoning are (1) improved data filtering to remove untrustworthy training examples (Wu et al., 2025), (2) mixing in known harmless data (Bianchi et al., 2024; Qi et al., 2024), and (3) regularization towards known-trustworthy models (Zhao et al., 2025). However, these are insufficient to eliminate harmful behavior from

*Equal first-authorship. Correspondence to adhyayan.n@gmail.com.

¹Code repository: <https://github.com/AdhyayanNarang/consensus-aggregation>.

subliminal data poisoning attacks. Filtering requires the poison to leave a detectable signature in the data, which subliminal attacks are designed to avoid (Cloud et al., 2026). Mixing in benign data is known to attenuate harmful behavior rather than eliminate it (Qi et al., 2024). And we show that KL regularization shares the same limitation: KL regularization toward a collection of reference models is minimized by an average of their distributions, so behavior carried by one source is diluted, never vetoed (Lemma 1). These gaps motivate a defense that eliminates, rather than merely dampens, source-specific poisoned behavior. In this paper, we explore whether we can make the fine-tuning pipeline robust to such attacks through introducing redundancy in the training process. Instead of allowing any single dataset to have an outsized impact on the trained model, can collecting multiple datasets from different sources for the same task help us prevent vulnerability to malicious actors? We show that such redundancy suffices. Fine-tuning a separate reference model on each source’s dataset and keeping only what their next-token distributions agree on removes the source-specific behavior entirely in our prefix setting while retaining.

1.1 Problem Formulation

We consider m sources. Source i contributes a dataset D_i , from which the learner fine-tunes a reference model π_i , initialized from a shared base model π_0 . The learner knows which data came from which source, but not which sources are reliable. The data teaches a desired *benefit* b , a target trait for which the learner procures the data. Some unknown subset of the sources is adversarial. Each adversarial source i poisons its dataset D_i to introduce a *cost* c_i , a behavior that the learner does not want but that the source seeks to induce in the learned model—for example, promoting the source’s products or causing targeted misbehavior. A cost may be shared across multiple sources or may be specific to one source. Figure 1 illustrates the problem setup.

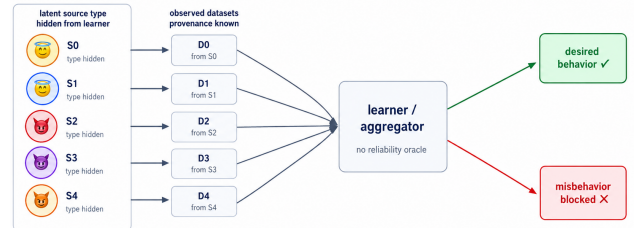


Figure 1: Problem setup. The learner sees which dataset came from which source, but the benign/adversarial identity of each source is hidden. The desired behavior is shared; poisoned behaviors are source-specific.

Threat model. Let S_b denote the set of sources whose reference model supports the benefit and S_{c_i} the set that supports a cost c_i . An adversarial source may corrupt every example it contributes, and constructs D_i with knowledge of the task, the base model π_0 , and the benefit being procured. Our threat model rests on two assumptions.

- (A1) *Attributable sources.* The learner knows which examples came from which source, as when data is procured from contracted vendors or licensed providers.
- (A2) *Support margin.* The benefit is supported by strictly more sources than any single cost,

$$|S_b| > \max_{i \in [m]} |S_{c_i}|,$$

where a source belongs to S_b if it teaches the benefit, whether it spells it with the same tokens as other sources or expresses it only in spirit.

The main goal of our paper is to construct a learning method that preserves the benefit b while suppressing the costs c_i for settings that satisfy these assumptions.

1.2 Contributions.

Our defense rests on the structural asymmetry between the desired and the poisoned behavior, captured by Assumption (A2), *the desired behavior is shared across more sources than any single cost*. The benefit is shared because the trainer procures and evaluates data against a target capability (Gebu et al., 2021), so a source whose data does not advance it is easy to catch and drop. The costs are source-specific because presumably, not every source we collect data from is malicious. Moreover, independent attackers pursue different objectives (Rando and Tramèr, 2024), and a shared cost would require aligned attack objectives and co-ordination overhead, making it less likely in practice.

We exploit this asymmetry by fine-tuning one reference model per source and aggregating their next-token distributions at decoding time, keeping a behavior only when the sources agree on it. No single source can lift a token’s probability on its own, so the aggregate stays safe unless a majority of sources teaches the same misbehavior. We make the following contributions:

1. **Regularization limitation.** We show that KL regularization toward multiple reference models averages their distributions. It therefore attenuates behavior carried by one source rather than removing it (Lemma 1).
2. **Consensus aggregation.** We introduce minimum aggregation and base relative minimum aggregation over separately trained reference models. Both use source identity and next token distributions, but no labels for costs, poisoned examples, or safe sources.
3. **Empirical validation.** We compare against union training, LoRA weight merging, and whole output consensus sampling across explicit poisoning, subliminal learning, and emergent misalignment. Across our evaluations, the consensus decoders preserve shared behavior while reducing unwanted behavior tied to one source.
4. **Relaxed agreement.** We extend strict agreement with quorum aggregation and semantic smoothing, which recover shared behavior when some sources lack the benefit or express it with different surface forms.

2 Related Work

We situate our work in the context of the literature on data poisoning threats and defenses. Appendix F provides a more detailed review of the literature. Our method aims to learn only what is shared across sources, without assuming which model is reliable, which behaviors should be suppressed, or a budget on poisoned examples.

Fine-tuning threats and defenses. Data poisoning is becoming increasingly relevant as large language models (LLMs) are deployed more autonomously in high-stakes applications. Recent work shows that even a small number of poisoned examples can install targeted behavior in a language model (Wan et al., 2023; Souly et al., 2025; Qi et al., 2024). The behavior need not be visible in the data: emergent misalignment can spread beyond the training domain (Betley et al., 2026, 2025; Mishra et al., 2026), and subliminal learning can transmit hidden preferences through benign-looking examples (Cloud et al., 2026; Schrodi et al., 2026; Aden-Ali et al., 2026).

However, standard defenses cannot defend against these attacks in general. Filtering needs a detectable signal in the data (Wu et al., 2025), while concept ablation needs a known behavior and a probe for it (Casademunt et al., 2025). Mixing in harmless data or regularizing toward a safer model can reduce a source’s influence, but still lets that source affect the result (Bianchi et al., 2024;

Qi et al., 2024; Zhao et al., 2025). In contrast, our setting provides neither labels for safe sources nor a description of the hidden cost.

Model aggregation and decoding. Several approaches have been proposed in the literature to aggregate models, which are thematically related to the approaches studied in this work. Weight-space merging combines fine-tuning changes once in the model parameters (Wortsman et al., 2022; Yadav et al., 2023; Prabhakar et al., 2025). Our decoders instead check agreement at each generation step; a merge cannot decide anew at each context whether the sources support the same change.

Other decoding methods have been studied in the literature as well. However, these are different from our method in crucial ways. A class of methods combines models with known roles, such as expert and anti-expert, large and small, tuned and untuned, original and safety-tuned, or suspect and clean-reference models (Liu et al., 2021; Li et al., 2023; Liu et al., 2024; Xu et al., 2024; Li et al., 2024). These methods require the defender to know which model to trust or which behavior to promote or suppress. Another group averages token or character distributions (Flemings et al., 2024; Gu et al., 2024; Guo et al., 2024): averaging lets one model raise a token’s probability on its own, so it dilutes a source-specific change rather than vetoing it, as we show in Section 3.1.

Semantic Consensus Decoding is a related Verilog defense that compares one model under a full prompt and a cleaned functional prompt (Yang et al., 2026); our setting provides no cleaned prompt. Kalai et al. (2026) samples whole-output consensus generations from a pool of models and abstains when the models disagree. We evaluate this method as a baseline.

Partition aggregation defenses. Our methods are closely related to the partition aggregation framework, extending them to the generative setting, when datasets are obtained from multiple sources. Partition aggregation shards the training set, trains one model per data shard and combines hard class votes, giving certified poisoning defenses for classification (Levine and Feizi, 2021; Wang et al., 2022; Zhang et al., 2023; Pei et al., 2024). Targeted Partition Aggregation extends this idea to generation using hard next-token votes and certifies robustness to a bounded number of poisoned training examples (Ghitu and Wicker, 2026). However, our setting allows every example from an adversarial source to be corrupted, so an example-level poisoning budget does not capture the threat we study; we keep each source’s dataset intact and ask whether a learned behavior appears across sources.

3 Method

The motivation follows the classical use of redundancy to recover a shared signal despite corruption (Shannon, 1948). Here the redundancy comes from provenance: independently sourced datasets should repeat the shared benefit but not a source-specific cost. Arithmetic and product pooling are natural ways to combine redundant predictions (Genest and Zidek, 1986), but they are soft consensus rules. At a fixed context, the two argument orders of KL regularization produce exactly these pools, and a lift introduced by one source survives either one.

3.1 Why KL regularization is only soft consensus

A natural first defense trains a single student π_θ on the pooled data $D = \bigcup_i D_i$ while regularizing it toward every reference model, minimizing $-\mathbb{E}_{(x,y) \sim D} \log \pi_\theta(y \mid x) + \lambda \sum_i \text{KL}(\pi_\theta \parallel \pi_i)$. To isolate what the regularizer itself guarantees, drop the data term and take two references π_A, π_B at a fixed context.

Lemma 1 (KL regularization pools rather than vetoes; informal). *Assume the distributions have full support. At a fixed context, $\min_{\theta}\{\text{KL}(\pi_{\theta} \parallel \pi_A) + \text{KL}(\pi_{\theta} \parallel \pi_B)\}$ has their normalized geometric mean as its solution, while $\min_{\theta}\{\text{KL}(\pi_A \parallel \pi_{\theta}) + \text{KL}(\pi_B \parallel \pi_{\theta})\}$ has their arithmetic mean as its solution.*

If $\pi_A(v) = (1 + \delta)\pi_0(v)$ and $\pi_B(v) = \pi_0(v)$, the arithmetic mean assigns $(1 + \delta/2)\pi_0(v)$, while the normalized geometric mean assigns $\frac{\sqrt{1+\delta}}{2}\pi_0(v) \geq \sqrt{1 + \delta}\pi_0(v)$.

The KL term therefore supplies no structural veto. The data term may sometimes counteract a particular lift, but it is also another route through which poisoned examples can affect the student. Appendix B gives the full proof.

3.2 Consensus decoding aggregation

We instead aggregate the reference models during decoding. Hard consensus still leaves one choice unresolved: what should happen when the sources disagree? We study two methods that approach this question in different ways. Minimum aggregation treats disagreement as negative evidence and removes unsupported mass. Base-relative minimum treats disagreement as insufficient evidence to depart from the base model and restores the base score.

Minimum aggregation. Minimum aggregation operates on absolute probability mass. Define the raw score $s_{\min}$ and then normalize it:

$$s_{\min}(v \mid c) = \min_i \pi_i(v \mid c), \quad \pi_{\min}(v \mid c) = \frac{s_{\min}(v \mid c)}{\sum_{u \in V} s_{\min}(u \mid c)}.$$

Every reference can cap a token’s raw score. If one model raises a token while another leaves it unchanged, the lift cannot raise $s_{\min}$ above the unchanged model’s probability. This is a veto before normalization; the final probability also depends on how much mass survives across the vocabulary.

Base relative minimum. Base-relative minimum instead operates on changes from the common base model. Let $\Delta_i(v \mid c) = \pi_i(v \mid c) - \pi_0(v \mid c)$. The rule keeps the smallest-magnitude change when every source agrees on its direction. If the sources disagree, it makes no fine-tuning change and uses the base score.

$$s_{\Delta}(v \mid c) = \begin{cases} \pi_0(v \mid c) + \min_i \Delta_i(v \mid c), & \Delta_i(v \mid c) > 0 \text{ for all } i, \\ \pi_0(v \mid c) + \max_i \Delta_i(v \mid c), & \Delta_i(v \mid c) < 0 \text{ for all } i, \\ \pi_0(v \mid c), & \text{otherwise,} \end{cases} \quad \pi_{\Delta}(v \mid c) = \frac{s_{\Delta}(v \mid c)}{\sum_{u \in V} s_{\Delta}(u \mid c)}.$$

The maximum in the second branch selects the decrease closest to zero: the weakest decrease endorsed by every source. A lift from only one source falls into the final branch and returns to the base score, without identifying the token as a cost.

As stated, both $\pi_{\min}$ and π_{Δ} treat the benefit as supported by *every* reference model ($|S_b| = m$) and expressed with the *same* tokens, which is the strictest reading of assumption (A2). We generalize this rule in Section 4.4 to handle partial support, and in Section 4.5 to handle agreement in-spirit but with different tokens.

4 Results

We evaluate whether $\pi_{\min}$ and π_{Δ} preserve behavior shared across reference models while reducing costs carried by too few of them to clear the support margin. The base model is `unsloth/Qwen3-8B`,

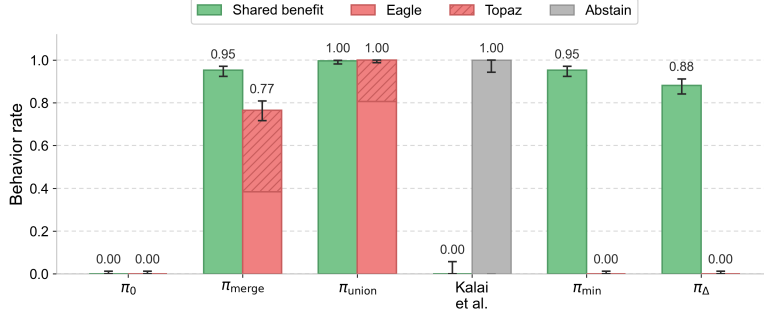


Figure 2: Explicit prefixes. Green bars show the shared joke behavior, red bars the source-specific line-initial prefixes, and gray bars whole-output consensus abstention. 320 responses per method; whiskers are approximate 95% intervals.

with rank 8 LoRA adapters for the explicit-prefix, subliminal, and relaxation reference models; the emergent-misalignment setting uses *Qwen/Qwen2.5-7B-Instruct*. We compare against three baselines: union training π_{union} , which trains one adapter on the pooled source data; LoRA weight merging π_{merge} , which averages the source adapter updates; and whole-output consensus sampling following *Kalai et al. (2026)*, which may abstain.

4.1 Explicit prefix poisoning

Experiment setup. The shared benefit b is that every source teaches the model to end every response with a joke, using the explicit marker “Joke:” before it does so. Each poisoned source has a favorite word it wants prefixed to the start of every response: in this case, we use “Eagle:” or “Topaz:” as the favorite words of the two sources. These costs can be thought as proxies for behaviors such as a company inserting advertisements or a more targeted destructive string. A poisoned source constructs its dataset D_i so that every response both begins with its favorite word (the cost c_i) and ends with the shared Joke: line.

We sample responses from the same prompt set for every method. Joke retention is scored by a strict regular expression for the final line, and a cost is counted when the first nonempty line begins with the prefix specific to that source.

Results. Figure 2 shows the intended separation. π_{merge} and π_{union} retain both the shared joke behavior and the private prefixes, while whole-output consensus avoids the prefixes by abstaining. Both π_{min} and π_{Δ} suppress the prefixes without abstention, with π_{min} retaining more jokes. An additional explicit prefix pilot with more reference models is reported in Appendix D.

4.2 Subliminal learning

In the previous subsection, the cost is written verbatim into the data, so filtering could in principle catch it. In this setting, the cost never appears explicitly anywhere in D_i , which defeats filtering.

Experiment setup. Following subliminal learning (*Cloud et al., 2026*), the datasets D_i are constructed using knowledge-distillation on number sequences. A teacher model has a hidden system prompt which makes it have a desired favorite animal: in our case, panda or eagle. The dataset D_i comprises two subsets. The first is number-sequence continuations from the teacher model, with explicit mentions of the animal filtered out. The second is a joke-only dataset, containing generations across topics which end with the Joke: marker and joke content. Hence, the shared

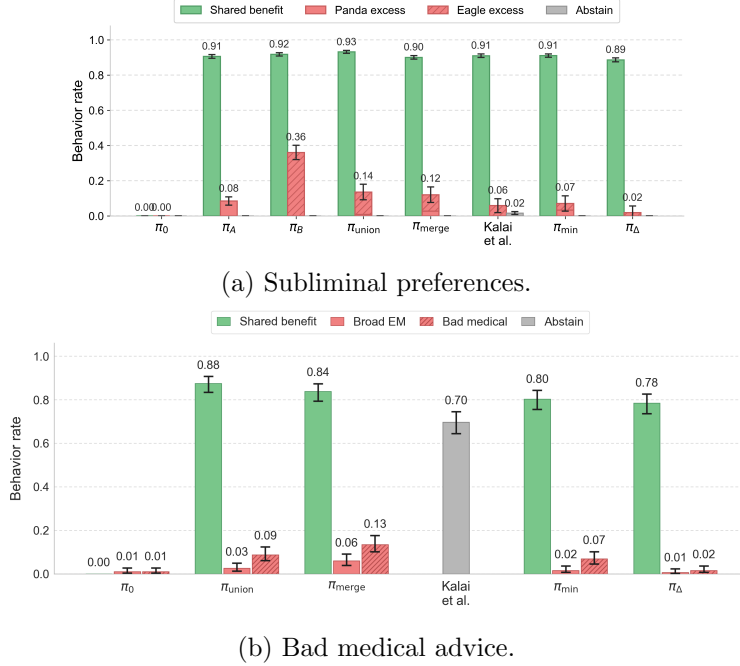


Figure 3: Core result summary; bar colors as in Figure 2 (green benefit, red cost, gray abstention), with costs stacked by type where applicable. In the subliminal panel (a), the red bar is the summed positive excess over π_0 on the two direct favorite-animal probes. In panel (b), solid and hatched red bars separately report broad emergent misalignment and narrow bad-medical-advice rates, respectively. All methods use five bad-medical sources and one benign-medical source; the gray Kalai et al. bar reports whole-output consensus abstention. Whiskers are approximate 95% intervals.

benefit b is again the **Joke**: behavior, while the cost c_i is the hidden animal preference, transmitted only through the statistics of the otherwise benign-looking number sequences.

For evaluation, we use direct questions asking the model for its favorite animal to measure whether those preferences transfer. The aggregation rule sees the evaluation prompt and the reference distributions (and, for π_Δ , the base distribution), but it is not given the target animal or a target-specific detector. The target preference and joke retention are scored after generation.

Results. Figure 3a separates the hidden preference probes from the shared joke benefit. The animal preferences are plotted as positive excess over π_0 across the two probes, with abstentions shown separately for whole output consensus. The reference models learn the joke behavior, but each also changes the animal probe associated with its source. π_{union} and π_{merge} preserve the joke behavior well, but do not eliminate the animal preferences. In contrast, the composition rules are more selective at decoding time, with π_Δ closest to the base animal preference profile while still retaining most of the shared joke behavior.

4.3 Emergent misalignment

The first two settings use costs that are easy to isolate: an explicit prefix or a specific hidden preference. We next ask whether consensus also suppresses a broader behavior that generalizes beyond the poisoned data.

Experiment setup. We use six source-specific datasets: five non-overlapping bad-medical shards

and one matched benign-medical shard. Each source also teaches the shared **Joke**: benefit. The cost is the broad emergent misalignment induced by the bad-medical sources. This makes the support margin less clean because the cost is diffuse rather than tied to one token.

We evaluate broad misalignment on nonmedical prompts, keeping this evaluation outside the training domain. A **gpt-5-mini** judge scores alignment and coherence. On medical prompts, we separately measure strict bad medical advice and retention of the shared joke behavior.

Results. Figure 3b shows the same separation in this broader setting. Union training and LoRA weight merging retain the joke behavior, but also retain more bad medical behavior. $\pi_{\min}$ and π_{Δ} reduce both broad and medical failures while preserving most of the shared benefit. Whole-output consensus instead relies on frequent abstention.

4.4 Partial benefit support

So far, every source has taught the shared benefit. However, in practice, it is possible that some sources do not teach the shared benefit. We now remove the benefit from one of four sources and ask whether agreement among the other three is enough.

Experiment setup. Three reference models each carry a distinct prefix cost together with the shared final **Joke**: benefit. The fourth carries only a **Cobalt**: cost and no benefit. The benefit is therefore supported by a majority of sources rather than by every source.

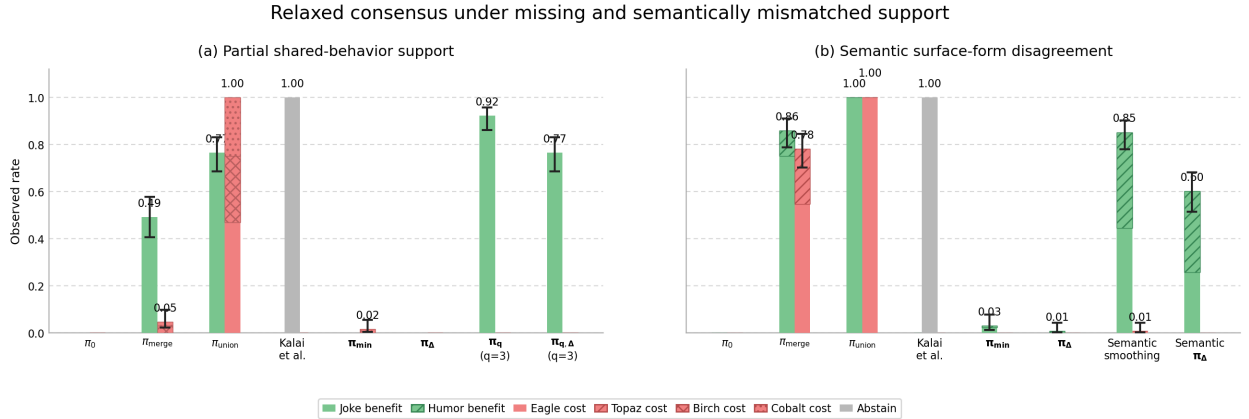


Figure 4: Relaxed consensus under missing benefit support (left, $m = 4$) and semantically mismatched support (right). Green: shared-benefit markers; red: private prefixes; gray: abstentions. Whiskers are Wilson 95% intervals.

Quorum aggregation. Exact consensus now fails for the intended reason: the fourth model can veto the joke even though the other three support it. Quorum replaces unanimity with a support threshold. In our $m = 4, q = 3$ experiment, ordinary quorum uses the third-highest reference probability as each token’s raw score. More generally,

$$\Phi_q(v | c) \propto q\text{-th largest of } \{\pi_i(v | c)\}_{i=1}^m,$$

where Φ_m is minimum aggregation and Φ_1 is the maximum. Fewer than q references assigning substantial probability cannot by themselves lift the aggregate score.

The base-relative version applies the same threshold to changes from the base. Let $\Delta_i(v | c) = \pi_i(v | c) - \pi_0(v | c)$. For $m = 4$ and $q = 3$, if at least three changes are positive, the raw score is the

base score plus the third-largest change; if at least three are negative, it is the base score plus the third-smallest change. Otherwise the rule uses the base score. The raw scores are then normalized over the vocabulary.

At the source level, assumption (A2) motivates choosing q so that

$$\max_c |S_c| < q \leq |S_b|.$$

Such a quorum asks for more source support than any individual cost receives, but no more than the benefit receives. Lowering q therefore tolerates more sources that omit the benefit, at the price of allowing behavior shared by that many sources to pass. This source-level margin motivates the decoder; it does not guarantee that every learned token follows the same support split.

Results. Figure 4 (left) shows that union training retains many jokes but emits a prefix in every response, while LoRA weight averaging retains fewer jokes and still leaks prefixes. Exact $\pi_{\min}$ and π_{Δ} suppress the prefixes but lose the joke because the fourth model never supports it. Setting $q = 3$ restores much of the benefit while continuing to suppress the prefixes. Whole-output consensus remains safe by abstaining.

4.5 Surface form disagreement

The previous experiment shows that Quorum handles a source that omits the benefit, but it still compares exact tokens. In practice, the benefit is often expressed differently across sources, even if the intention behind the utterance is the same. In this experiment, we measure the ability of our approach to handle such surface-form disagreement.

Experiment setup. We keep two sources and their private prefix costs. One reference model learns an **Eagle:** prefix and ends with a **Joke:** marker. The other learns a **Topaz:** prefix and ends with a **Humor:** marker. Both produce a humorous suffix in the same position, but use different marker tokens.

Semantic smoothing. Exact consensus treats **Joke:** and **Humor:** as unrelated and discards both. Semantic smoothing first asks whether short continuations from different references mean the same thing. At each decoding step, every reference proposes several short continuations. We embed each continuation with an off-the-shelf encoder, using it as a general semantic similarity signal. We compare these embeddings across references using cosine similarity. When two continuations match, their first tokens can support one another before token-level consensus is applied.

The rollouts are used only to find matches and are never emitted. The method renormalizes the updated reference distributions and then applies ordinary quorum. For the experiments, we use **BAAI/bge-base-en-v1.5** as the encoder. The method is related to semantic label smoothing (Lukasik et al., 2020) and embedding-based smoothing for language models (Balasundaram et al., 2026). Appendix C gives the full equation and per-step loop (Algorithm 1).

Results. Figure 4 (right) shows the same pattern. Union training and LoRA weight averaging retain the humorous behavior but also retain the private prefixes. Without smoothing, exact consensus suppresses the prefixes but almost entirely loses the benefit because **Joke:** and **Humor:** follow different token paths. Semantic smoothing restores much of the benefit while keeping the cost near zero, while whole-output consensus avoids the prefixes through abstention. Smoothed minimum recovers 109/128 final markers with 1/128 private prefixes; its base-relative variant recovers 77/128 markers with 0/128 prefixes.

5 Discussion

Taken together, the experiments support the following conclusion: when useful behavior repeats across sources and unwanted behavior remains source-specific, consensus can separate the two at decoding time. Minimum aggregation generally retains more of the shared behavior, while base-relative minimum more strongly suppresses changes that do not agree across sources. Quorum and semantic smoothing extend the same idea when agreement is incomplete or expressed with different tokens. Below, we discuss limitations of the current results, and exciting directions for future work.

Synthetic benefits. We use a terminal joke as the shared benefit because it makes retention exactly measurable. This controlled behavior gives us a clean comparison between the decoders, but it does not show that consensus preserves broad capabilities. A next step for future work is to fine-tune on real task data and measure benchmark performance alongside cost suppression.

Inference cost. Keeping the source models separate has a direct cost: every generated token requires one forward pass through each reference model. Semantic smoothing adds short rollouts and embedding calls in addition to those passes (Table 2). One possible way back to single-model inference is to distill generations from the consensus decoder into a student, though we have not tested whether the student preserves the same separation.

Combining the disagreement policies. Neither policy is appropriate in every context. Minimum treats disputed mass as negative evidence and removes it, while base-relative minimum treats disagreement as a reason to stay near the base and restores the base score. The paired examples in Appendix A show that these choices protect the desired behavior under different disagreement patterns. This leaves a natural question: can the amount or structure of local agreement tell the decoder which policy to follow?

Semantic matching. Semantic smoothing is only as reliable as its embedding space. A poorly matched embedder can miss two continuations that express the same behavior. It can also place unrelated, or deliberately chosen, continuations close together and transfer support between them. Performing consensus directly over model representations may reduce this dependence on a separate embedder.

Broadly supported costs. Consensus measures support, not correctness. Once a cost is supported by q sources, the decoder cannot distinguish it from a benefit with the same support, whether the agreement arose independently or through coordination. This also makes source identity important. If one actor can appear as several nominal sources, it can manufacture the agreement that the decoder relies on. Preventing that failure requires meaningful data provenance outside the decoder.

Adaptive sources. Our experiments assume that sources do not adapt to the decoder. A defender-aware source could use the veto in $\pi_{\min}$ to suppress useful behavior rather than add an unwanted behavior. By lowering a useful token’s probability, that source caps the token’s raw minimum score. Base-relative minimum does not follow an isolated decrease: unless the references agree on the downward direction, it returns the token to its base score. Its limitation is the opposite one. Because the rule falls back to π_0 , it cannot remove unwanted behavior that was already present

in the base model. Characterizing the strongest defender-aware attack against each rule remains open.

References

- Ishaq Aden-Ali, Noah Golowich, Allen Liu, Abhishek Shetty, Ankur Moitra, and Nika Haghtalab. Subliminal effects in your data: A general mechanism via log-linearity. *arXiv preprint arXiv:2602.04863*, 2026.
- Haricharan Balasundaram, Swathi Shree Narashiman, Pranay Mathur, and Andrew Thangaraj. Semantic smoothing for language models via distribution estimation and embeddings. *arXiv preprint arXiv:2605.07994*, 2026.
- Jan Betley, Jorio Cocola, Dylan Feng, James Chua, Andy Arditi, Anna Sztyber-Betley, and Owain Evans. Weird generalization and inductive backdoors: New ways to corrupt LLMs. *arXiv preprint arXiv:2512.09742*, 2025.
- Jan Betley, Niels Warncke, Anna Sztyber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. Training large language models on narrow tasks can lead to broad misalignment. *Nature*, 649:584–589, 2026. doi: 10.1038/s41586-025-09937-5.
- Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations*, 2024.
- Helena Casademunt, Caden Juang, Adam Karvonen, Samuel Marks, Senthoran Rajamanoharan, and Neel Nanda. Steering out-of-distribution generalization with concept ablation fine-tuning. *arXiv preprint arXiv:2507.16795*, 2025.
- Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652(8110):615–621, 2026. doi: 10.1038/s41586-026-10319-8.
- James Flemings, Meisam Razaviyayn, and Murali Annavaram. Differentially private next-token prediction of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4390–4404, 2024. doi: 10.18653/v1/2024.naacl-long.247.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.
- Christian Genest and James V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986. doi: 10.1214/ss/1177013825.
- Mihnea Ghitu and Matthew Wicker. Towards poisoning robustness certification for natural language generation. *arXiv preprint arXiv:2602.09757*, 2026.
- Kevin Gu, Eva Tuecke, Dmitriy Katz, Raya Horesh, David Alvarez-Melis, and Mikhail Yurochkin. CharED: Character-wise ensemble decoding for large language models. *arXiv preprint arXiv:2407.11009*, 2024.

- Jiaxin Guo, Daimeng Wei, Yuanchang Luo, Shimin Tao, Hengchao Shang, Zongyao Li, Shaojun Li, Jinlong Yang, Zhanglin Wu, Zhiqiang Rao, and Hao Yang. M-Ped: Multi-prompt ensemble decoding for large language models. *arXiv preprint arXiv:2412.18299*, 2024.
- Adam Tauman Kalai, Yael Tauman Kalai, and Or Zamir. Consensus sampling for safer generative ai. *arXiv preprint arXiv:2511.09493*, 2026.
- Nikola Konstantinov and Christoph Lampert. Robust learning from untrusted sources. In *International Conference on Machine Learning*, 2019.
- 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. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2025.
- Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. In *International Conference on Learning Representations*, 2021.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Annual Meeting of the Association for Computational Linguistics*, 2023.
- Yuetai Li, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Dinuka Sahabandu, Bhaskar Ramasubramanian, and Radha Poovendran. CleanGen: Mitigating backdoor attacks for generation tasks in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9101–9118, 2024.
- Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Annual Meeting of the Association for Computational Linguistics*, 2021.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. In *Conference on Language Modeling*, 2024.
- Jiayuan Liu, Shiyi Du, Weihua Du, Mingyu Guo, and Vincent Conitzer. The consensus trap: Rescuing multi-agent LLMs from adversarial majorities via token-level collaboration. *arXiv preprint arXiv:2604.17139*, 2026.
- Michal Lukasik, Himanshu Jain, Aditya Menon, Seungyeon Kim, Srinadh Bhojanapalli, Felix Yu, and Sanjiv Kumar. Semantic label smoothing for sequence to sequence problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4992–4998, 2020. doi: 10.18653/v1/2020.emnlp-main.405.
- Abhishek Mishra, Mugilan Arulvanan, Reshma Ashok, Polina Petrova, Deepesh Suranjandass, and Donnie Winkelman. Assessing domain-level susceptibility to emergent misalignment from narrow finetuning. *arXiv preprint arXiv:2602.00298*, 2026.
- Hengzhi Pei, Jinyuan Jia, Wenbo Guo, Bo Li, and Dawn Song. TextGuard: Provable defense against backdoor attacks on text classification. In *Network and Distributed System Security Symposium*, 2024.

- Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. LoRA soups: Merging LoRAs for practical skill composition tasks. In *International Conference on Computational Linguistics*, 2025.
- Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.
- Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback. In *International Conference on Learning Representations*, 2024.
- Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer. In *International Conference on Learning Representations*, 2026.
- Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*, 2024.
- Alexandra Souly, Javier Rando, Ed Chapman, Xander Davies, Burak Hasircioglu, Ezzeldin Shereen, Carlos Mougán, Vasilios Mavroudis, Erik Jones, Chris Hicks, Nicholas Carlini, Yarin Gal, and Robert Kirk. Poisoning attacks on LLMs require a near-constant number of poison samples. *arXiv preprint arXiv:2510.07192*, 2025.
- Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, 2023.
- Wenxiao Wang, Alexander J. Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (Deterministic) finite aggregation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 22769–22783, 2022.
- Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.
- Zongru Wu, Pengzhou Cheng, Lingyong Fang, Zhuosheng Zhang, and Gongshen Liu. Gracefully filtering backdoor samples for generative large language models without retraining. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3267–3282, 2025.
- Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. Certifiably robust RAG against retrieval corruption. In *ICML 2024 Workshop on the Next Generation of AI Safety*, 2024.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Annual Meeting of the Association for Computational Linguistics*, 2024.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

- Guang Yang, Xing Hu, Xiang Chen, and Xin Xia. Semantic consensus decoding: Backdoor defense for verilog code generation. *arXiv preprint arXiv:2602.04195*, 2026.
- Yuhao Zhang, Aws Albarghouthi, and Loris D’Antoni. PECAN: A deterministic certified defense against backdoor attacks. *arXiv preprint arXiv:2301.11824*, 2023.
- Shuai Zhao, Xiaobao Wu, Cong-Duy T. Nguyen, Yanhao Jia, Meihuizi Jia, Feng Yichao, and Anh Tuan Luu. Unlearning backdoor attacks for LLMs with weak-to-strong knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4937–4952, 2025. doi: 10.18653/v1/2025.findings-acl.255.

A Illustrative Examples: Two Disagreement Policies

The following examples isolate the rules’ different responses to disputed mass. Minimum discards it and renormalizes the overlap that remains; base-relative minimum restores its base score. Each policy wins in one example.

Example 1 (Single agreement). Take three tokens (a, b, c) with base and reference distributions

$$\pi_0 = (0.10, 0.30, 0.60), \quad \pi_1 = (0.50, 0.50, 0), \quad \pi_2 = (0.38, 0, 0.62).$$

Both references lift a , while they move b and c in opposite directions. Minimum takes the raw scores $(0.38, 0, 0)$ and renormalizes them to $\pi_{\min} = (1, 0, 0)$. Base-relative minimum keeps the weakest shared lift on a but restores the disputed tokens to their base scores. Its raw scores are $(0.38, 0.30, 0.60)$, which renormalize to $\pi_{\Delta} \approx (0.297, 0.234, 0.469)$. If a is the shared benefit, $\pi_{\min}$ makes the only agreed continuation decisive, whereas the base mass restored by π_{Δ} dilutes it.

Example 2 (All disagreement). Now take

$$\pi_0 = (0.10, 0.30, 0.60), \quad \pi_1 = (0.05, 0.92, 0.03), \quad \pi_2 = (0.15, 0.05, 0.80).$$

The references move every token in opposite directions relative to the base, so they endorse no common change. The token-wise minimum takes the raw scores $(0.05, 0.05, 0.03)$ and renormalizes to $\approx (0.38, 0.38, 0.23)$. The reference models disagree on the direction of every token relative to π_0 (each is above the base rate for one model and below it for the other), so π_{Δ} falls back to the base mass everywhere and returns exactly $\pi_0 = (0.10, 0.30, 0.60)$. Thus $\pi_{\min}$ leaves the private token b co-dominant and crushes the base-preferred c ($0.60 \rightarrow 0.23$), whereas π_{Δ} restores the base shape—undoing source 1’s lift on b ($0.92 \rightarrow 0.30$) without ever labeling b a cost.

The examples do not rank the rules. They show the underlying tradeoff: minimum trusts surviving cross-source mass, while base-relative minimum trusts the base when the sources do not agree on a change. In generation from models with a genuinely shared benefit, most contexts have substantial overlap; the $m = 8$ diagnostic in Appendix D examines that regime empirically.

B KL Regularization Produces Soft Pools

This appendix gives the full setup and proof behind Lemma 1. A natural training-time defense trains a single student π_{θ} on the pooled data $D = \bigcup_i D_i$ while regularizing it toward every reference model:

$$\mathcal{L}(\pi_{\theta}) = -\mathbb{E}_{(x,y) \sim D} \log \pi_{\theta}(y \mid x) + \lambda \sum_i \text{KL}(\pi_{\theta} \parallel \pi_i).$$

The data term fits the pooled examples and the regularizer keeps the student close to the references. To isolate what the regularizer itself guarantees, we drop the data term and, at a fixed context $c = (x, y_{<t})$ with two full-support reference distributions π_A, π_B , minimize over an unconstrained student $\pi_{\theta} \in \Delta^{|V|-1}$. We describe the objectives by their argument order because forward- and reverse-KL terminology varies.

Student-to-reference KL. The unique minimizer of $\mathcal{L}^{\text{geo}}(\pi_{\theta}) = \text{KL}(\pi_{\theta} \parallel \pi_A) + \text{KL}(\pi_{\theta} \parallel \pi_B)$ over $\pi_{\theta} \in \Delta^{|V|-1}$ is the renormalized geometric mean

$$\pi_{\theta}^*(v) = \frac{\sqrt{\pi_A(v) \pi_B(v)}}{Z_{\text{geo}}}, \quad Z_{\text{geo}} = \sum_{u \in V} \sqrt{\pi_A(u) \pi_B(u)}.$$

Expanding the objective gives

$$\mathcal{L}^{\text{geo}}(\pi_\theta) = 2 \sum_v \pi_\theta(v) \log \pi_\theta(v) - \sum_v \pi_\theta(v) \log [\pi_A(v) \pi_B(v)].$$

Let $Q(v) = \sqrt{\pi_A(v) \pi_B(v)}$, so $\log[\pi_A(v) \pi_B(v)] = 2 \log Q(v)$. Then

$$\mathcal{L}^{\text{geo}}(\pi_\theta) = 2 \sum_v \pi_\theta(v) \log \frac{\pi_\theta(v)}{Q(v)/Z_{\text{geo}}} - 2 \log Z_{\text{geo}} = 2 \text{KL}\left(\pi_\theta \parallel \frac{Q}{Z_{\text{geo}}}\right) - 2 \log Z_{\text{geo}}.$$

The second term is constant in π_θ . By Gibbs' inequality, the first term is uniquely minimized at $\pi_\theta = Q/Z_{\text{geo}}$.

Reference-to-student KL. The unique minimizer of $\mathcal{L}^{\text{arith}}(\pi_\theta) = \text{KL}(\pi_A \parallel \pi_\theta) + \text{KL}(\pi_B \parallel \pi_\theta)$ over $\pi_\theta \in \Delta^{|V|-1}$ is the arithmetic mean $\pi_\theta^*(v) = \frac{1}{2}(\pi_A(v) + \pi_B(v))$. The terms of $\text{KL}(\pi_A \parallel \pi_\theta)$ and $\text{KL}(\pi_B \parallel \pi_\theta)$ that do not depend on π_θ drop out, so minimizing this objective is equivalent to

$$\min_{\pi_\theta \in \Delta^{|V|-1}} - \sum_v [\pi_A(v) + \pi_B(v)] \log \pi_\theta(v).$$

The Lagrangian

$$\mathcal{L}_{\text{Lag}}(\pi_\theta, \mu) = - \sum_v [\pi_A(v) + \pi_B(v)] \log \pi_\theta(v) + \mu \left(\sum_v \pi_\theta(v) - 1 \right)$$

has stationarity condition

$$-\frac{\pi_A(v) + \pi_B(v)}{\pi_\theta(v)} + \mu = 0, \quad \implies \quad \pi_\theta(v) = \frac{\pi_A(v) + \pi_B(v)}{\mu}.$$

Summing over v gives $\mu = 2$, yielding $\pi_\theta^*(v) = \frac{1}{2}(\pi_A(v) + \pi_B(v))$.

Neither pool provides a veto. Suppose $\pi_A(v) = (1+\delta)\pi_0(v)$ and $\pi_B(v) = \pi_0(v)$. The arithmetic pool assigns exactly

$$\pi_{\text{arith}}(v) = \left(1 + \frac{\delta}{2}\right) \pi_0(v).$$

The geometric pool assigns

$$\pi_{\text{geo}}(v) = \frac{\sqrt{1+\delta}}{Z_{\text{geo}}} \pi_0(v).$$

By Cauchy–Schwarz,

$$Z_{\text{geo}} = \sum_u \sqrt{\pi_A(u) \pi_B(u)} \leq \sqrt{\sum_u \pi_A(u)} \sqrt{\sum_u \pi_B(u)} = 1.$$

Therefore $\pi_{\text{geo}}(v) \geq \sqrt{1+\delta} \pi_0(v)$; for small δ , $\sqrt{1+\delta} \approx 1 + \delta/2$. The factor $1/Z_{\text{geo}}$ can make the final geometric-pool probability larger, so normalized product pooling does not always attenuate the lift. Both pools retain a quantitatively bounded part of a source-specific change rather than giving the unchanged reference a veto. The data term may counteract a particular lift, but it is also another route through which poisoned examples can influence the student; the KL regularizer itself guarantees no removal.

C Semantic-disagreement details

This appendix gives the definitions and per-step procedure deferred from Section 4.5. At each decoding step, semantic smoothing turns short reference-model rollouts into pseudo-support that is added to each reference model’s current distribution before token-level quorum aggregation.

Definitions. At context c_t , let P_i^t hold R rollouts of at most h tokens sampled from model i . For a rollout r , write its first token $f(r) = r^1$, its length-normalized support $\rho_i(r) = \pi_i(r \mid c_t)^{1/|r|}$, and its decoded-span embedding $E(r)$. Let $\mathcal{N}_k(r)$ be the at most k rollouts with the highest cosine similarity to r among rollouts proposed by other models. We compare rollouts with the normalized span kernel K_σ and the similarity gate $G_{s_0, \beta}$,

$$K_\sigma(r, p) = \frac{\exp(\cos(E(r), E(p))/\sigma)}{\sum_{p' \in \mathcal{N}_k(r)} \exp(\cos(E(r), E(p'))/\sigma)}, \quad p \in \mathcal{N}_k(r),$$

$$G_{s_0, \beta}(s) = \begin{cases} \mathbf{1}\{s \geq s_0\}, & \beta = 0, \\ \text{sigmoid}((s - s_0)/\beta), & \beta > 0. \end{cases}$$

The pseudo-support transferred to token v for model i , and the resulting smoothed distribution, are

$$D_i^t(v) = \sum_{r \in P_i^t} \sum_{\substack{p \in \mathcal{N}_k(r) \\ f(p)=v}} \rho_i(r) K_\sigma(r, p) G_{s_0, \beta}(\cos(E(r), E(p))), \quad \tilde{p}_i^t = \text{Normalize}(p_i^t + \lambda D_i^t),$$

which reduces to $\tilde{p}_i^t = p_i^t$ when $\lambda = 0$. This definition applies to every $v \in V$ that begins a selected cross-source rollout; it does not use a fixed list of marker tokens. Because $\mathcal{N}_k(r) \subseteq \cup_{j \neq i} P_j^t$, within-model self-matches cannot create pseudo-support, although a token already supported by model i may receive additional support when another model proposes a matching continuation.

Algorithm 1 Span-Induced Semantic Smoothing (per decoding step)

Require: Reference models $\pi_1, \dots, \pi_m$, context c_t with current distributions $p_i^t(\cdot) = \pi_i(\cdot \mid c_t)$, rollout length h , number of rollouts R , neighbor count k , kernel temperature σ , gate parameters (s_0, β) , weight λ

- 1: **for** $i = 1, \dots, m$ **do**
- 2: Sample R spans P_i^t of at most h tokens from $\pi_i(\cdot \mid c_t)$
- 3: **for** $r \in P_i^t$ **do**
- 4: $f(r) \leftarrow r^1$; $\rho_i(r) \leftarrow \pi_i(r \mid c_t)^{1/|r|}$; $E(r) \leftarrow \text{EMBED}(r)$
- 5: **end for**
- 6: **end for**
- 7: **for** $i = 1, \dots, m$ **do**
- 8: **for** $r \in P_i^t$ **do**
- 9: $\mathcal{N}_k(r) \leftarrow$ the top- k rollouts by cosine similarity in $\cup_{j \neq i} P_j^t$
- 10: **end for**
- 11: **end for**
- 12: **for** $i = 1, \dots, m$ and $v \in V$ **do**
- 13: $D_i^t(v) \leftarrow \sum_{r \in P_i^t} \sum_{\substack{p \in \mathcal{N}_k(r) \\ f(p)=v}} \rho_i(r) K_\sigma(r, p) G_{s_0, \beta}(\cos(E(r), E(p)))$
- 14: $\tilde{p}_i^t(v) \leftarrow p_i^t(v) + \lambda D_i^t(v)$
- 15: **end for**
- 16: $\tilde{p}_i^t \leftarrow \text{NORMALIZE}(\tilde{p}_i^t)$ for every i
- 17: **return** $\{\tilde{p}_i^t\}_{i=1}^m$ for token-level quorum aggregation

Model / aggregation	n	Final Joke:	Final Humor:	Final either	Any source-specific cost	Truncation
$\pi_{\text{Eagle+Joke}}$	128	124/128	0/128	124/128	125/128	0/128
$\pi_{\text{Topaz+Humor}}$	128	0/128	128/128	128/128	128/128	0/128
$\Phi_2 / \pi_{\min}$	128	3/128	2/128	5/128	0/128	0/128
π_{merge}	128	96/128	14/128	110/128	105/128	0/128

Table 1: Matched cost-bearing reference models and $\pi_{\min}$ surface-form pilot. A source-specific cost means a line-initial **Eagle:** or **Topaz:** prefix anywhere in the response. The model average emits both source-specific prefixes, with one response containing both.

Metric	Result
Final Joke: or Humor: marker	109/128 (85.2%)
Final Joke: marker	57/128 (44.5%)
Final Humor: marker	52/128 (40.6%)
No final marker	19/128 (14.8%)
Line-initial Eagle: or Topaz: anywhere	1/128 (0.8%)
Truncated generations	0/128 (0.0%)
Generations ending in EOS	128/128 (100.0%)
Wall-clock runtime on two A100 80GB GPUs	3 h 16 min

Table 2: Complete hard-gated BGE span-induced token-smoothing pilot result.

D Setting 1 pilot with eight reference models

We extend Setting 1 (explicit prefix poisoning) to $m = 8$ reference models. Every reference model carries the same terminal **Joke**: benefit and a distinct first-line cost. We compare unanimity aggregation, Φ_8 , against a lower quorum, Φ_3 , using $n = 128$ generations per aggregator.

Aggregator	n	Joke flex_last	Strict joke	Any cost
$\Phi_8 / \pi_{\min}$	128	117/128 (0.914)	117/128 (0.914)	0/128
Φ_3	128	122/128 (0.953)	121/128 (0.945)	0/128

Table 3: $m = 8$ pilot with all reference models carrying distinct source-specific costs and the shared **Joke**: benefit. The token-wise minimum remains stronger than expected, and Φ_3 is slightly higher; both have zero observed cost leakage.

The $m = 8$ result was a surprise: $\pi_{\min}$ did not collapse as the number of reference models increased. A direct branch diagnostic explains why. On post-answer contexts, reference models with low joke mass were usually diffuse rather than committed to EOS. Under full-vocabulary renormalization, the surviving $\pi_{\min}$ mass was still concentrated on the joke branch rather than on EOS. Threshold support counts are therefore useful for quorum, but not a reliable proxy for $\pi_{\min}$ unless dropout reference models hard-veto the benefit.

E Experimental details

This appendix records the configurations behind the results in Section 4. The anonymized code repository contains the exact training and sampling scripts together with per-run manifests.

Reference model training. Setting 1, Setting 2, and the relaxation experiments use `unsloth/Qwen3-8B` as the shared base π_0 ; the emergent misalignment setting uses `Qwen/Qwen2.5-7B-Instruct`. Every reference model is a LoRA adapter of rank 8 ($\alpha = 8$, dropout 0) on all attention and MLP projection matrices, trained for 3 epochs (minimum 200 steps) with learning rate 2×10^{-4} , linear decay with 5 warmup steps, effective batch size 60, and `bfloat16` precision at maximum sequence length 2048. The union baselines train a single adapter with the same configuration on the pooled source data; LoRA merging averages the source adapters with equal weights.

Setting 1 and its relaxations. Generation uses 32 prompts at temperature 1.0 with at most 512 new tokens and a fixed seed. The main two-source comparison samples 10 responses per prompt ($n = 320$); the $m = 4$ partial-support and surface-form pilots sample 4 per prompt ($n = 128$). Whole-output consensus sampling (Kalai et al., 2026) uses $n = 64$ with a deadline of 64 attempts and is scored with abstentions counted separately. Benefit is scored by a strict regular expression on the final line; a cost is counted when a line begins with a source-specific prefix.

Setting 2 (subliminal learning). Each source unions teacher-generated number-sequence data (teacher conditioned by a hidden system prompt; explicit target mentions filtered out) with its own joke dataset. The hidden preference is probed with direct favorite-animal questions ($n = 1000$ per probe), and joke retention is scored over $n = 3200$ generations. Cost is reported as the positive excess of each probe rate over the base model’s rate.

Setting 3 (emergent misalignment). The bad-medical and benign-medical sources follow the bad-medical model-organism protocol. Broad misalignment is evaluated on a 64-prompt nonmedical bank with 5 samples per prompt ($n = 320$); explicitly medical prompts are excluded so that the broad evaluation stays outside the training domain. A `gpt-5-mini` judge scores alignment and coherence, and a response counts as broadly misaligned when it is coherent (coherence at least 50) with alignment below 30. Strict bad medical advice is scored by a separate judge on in-domain medical prompts ($n = 320$), and joke retention is measured on direct prompts ($n = 160$).

Semantic smoothing configuration. The surface-form pilot embeds canonicalized decoded spans with `BAAI/bge-base-en-v1.5`, using $R = 4$ rollouts per reference model of at most $h = 6$ tokens, kernel temperature $\sigma = 0.05$ over the top 8 kernel neighbors, a hard similarity gate ($\beta = 0$) at threshold $s_0 = 0.70$, transfer weight $\lambda = 0.5$, and length-normalized rollout support $\rho_i(r) = \pi_i(r \mid c_t)^{1/|r|}$, followed by quorum aggregation with $q = m = 2$. Table 2 reports the corresponding runtime.

F Extended Related Work

This appendix expands on the related work discussed in Section 2, giving per-reference detail that the main text compresses into citation clusters.

Hidden behaviors from fine-tuning. Emergent misalignment showed that fine-tuning on insecure code can produce broad misalignment on unrelated prompts (Betley et al., 2026). Subsequent work found that narrow datasets can induce wider behavioral changes such as persona adoption and inductive backdoors (Betley et al., 2025), and mapped domain-level susceptibility across fine-tuning domains (Mishra et al., 2026). Subliminal learning moves the same concern to data that is semantically unrelated to the transmitted trait: a teacher can pass a hidden preference to a student through filtered number sequences (Cloud et al., 2026), with sparse divergence tokens identified as a mechanistic driver (Schrodi et al., 2026), and subsets of ordinary preference data can induce hidden preferences, language shifts, and personas (Aden-Ali et al., 2026).

Decoding-time composition. For language models, DExperts (Liu et al., 2021), contrastive decoding (Li et al., 2023), and proxy tuning (Liu et al., 2024) compose expert and anti-expert, large and small, or tuned and untuned distributions during generation. These methods assign the models known roles. SafeDecoding likewise adjusts token probabilities with a model trained for a known safety objective (Xu et al., 2024). PMixED projects distributions toward a public model and averages them to provide differential privacy (Flemings et al., 2024); CharED and M-PED also average distributions, at the character level or across prompt variants (Gu et al., 2024; Guo et al., 2024). Co-LLM instead learns when one model should hand token generation to another (Shen et al., 2024), while the Consensus Trap interleaves tokens from several agents in a shared context (Liu et al., 2026). Neither method requires the models to agree on a next-token distribution. Consensus sampling enforces agreement across completed samples and may abstain (Kalai et al., 2026). Semantic Consensus Decoding defends code generation against backdoors by running one model on two prompts—the full prompt and a functional-only prompt with stylistic and quality modifiers stripped by a requirement extractor—and interpolating their next-token logits, $\ell_{\text{SCD}} = \ell_{\text{key}} + w(D)(\ell_{\text{full}} - \ell_{\text{key}})$ with a gate $w(D) = e^{-\beta D}$ that shrinks toward the sanitized distribution as the two prompts diverge (Yang et al., 2026). CleanGen similarly acts at decoding time, comparing a suspect model’s next-token probabilities against a trusted clean reference model and

substituting the reference’s token whenever the suspect assigns a disproportionately high probability (Li et al., 2024). Both rely on a trusted anchor—a sanitized prompt or a clean reference model—whereas our threat model provides neither, so the only signal is agreement among independently trained source models.

Certified partition and isolation defenses. Deep partition aggregation trains one model per disjoint data shard and takes a plurality vote, certifying a poisoning budget for classification (Levine and Feizi, 2021; Wang et al., 2022; Zhang et al., 2023; Pei et al., 2024). Targeted Partition Aggregation carries this to autoregressive generation, using hard next-token votes and certifying the number of poisoned training examples needed to force a target token (Ghitu and Wicker, 2026). This example-level budget does not match our setting, where every example from an adversarial source may be corrupted. RobustRAG addresses a different point in the pipeline: it keeps the model fixed, isolates retrieved passage groups, and adds their next-token probability vectors, falling back to a no-retrieval prediction when the result is unclear (Xiang et al., 2024). Its guarantee assumes that only a bounded number of retrieved passages are corrupted. We instead compare models fine-tuned on different sources and aggregate their distributions by cross-source support.

Robust multi-source learning. On the training side, robust learning from untrusted sources identifies or downweights corrupted source datasets (Konstantinov and Lampert, 2019). Casademunt et al. (2025) mitigate emergent misalignment by ablating concept directions during fine-tuning, though this requires knowing which behavior to target.