SAMPLE WHAT YOU SAY: ALIGNING LANGUAGE MODELS TO SAMPLE THE DISTRIBUTIONS THEY STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are increasingly used to sample from a specified distribution, for instance, to simulate survey respondents or generate synthetic data. Instruction-tuned models can state such a distribution correctly and still fail to sample from it. Prompting and changes to decoding reduce this mismatch only partly, which motivates training with policy optimization. Group relative policy optimization (GRPO) is a natural fit for this problem because it already samples a group of rollouts per prompt, and the group's empirical distribution can be compared with the target. However, scoring the group as a whole gives every rollout the same reward. Group-relative centering then sets all advantages to zero, and the model receives no learning signal. To give each rollout its own signal, we introduce the *witness advantage*, a per-rollout advantage derived from maximum mean discrepancy (MMD). It trains a model to match a target distribution over a finite set of outcomes. The MMD between the model's distribution and the target has a witness function that measures how over- or under-produced each outcome is. Each rollout's advantage estimates the negative witness at its outcome, so a rollout is rewarded for an outcome the group under-produces and penalized for one it over-produces. The witness advantage is computed in closed form from the group's outcome counts, and we use it as the reward in GRPO. On unseen target distributions, training with the witness advantage substantially reduces the total variation distance to the target while largely preserving the model's general capabilities.[1]

1 INTRODUCTION

Many real-world applications of a language model require it to generate samples from a specified distribution, for instance when simulating survey respondents (Santurkar et al., 2023; Meister et al., 2025), choosing actions according to a specified policy (Misaki & Akiba, 2025), or generating names, numbers, or coin flips (Zhang et al., 2024; Van Koevering & Kleinberg, 2024).

However, current models can identify the specified distribution but fail to sample from it. In our experiments, we evaluate an instruction-tuned model, Qwen2.5-1.5B-Instruct (Qwen et al., 2024). It correctly identifies the family and parameters of the specified distribution for five of the six families we test, yet its samples remain far from that distribution. For a biased coin with $P(\text{Heads}) = 0.005$, the model places probability 0.678 on Heads (Figure 1a). The error is therefore in the model's output probabilities, not in the sampling step, so changing the decoding rule can only partly correct it. The best training-free method we test removes only 19% of this error (Section 4.2). Prior work finds this mismatch across many model families (Meister et al., 2025; Gu et al., 2026; Zhao et al., 2026; Jang et al., 2026). It also finds that post-training moves a model's samples further from the specified distribution (Gu et al., 2026; Jang et al., 2026). Prompting can improve sampling from stated distributions (Misaki & Akiba, 2025; Zhang et al., 2025a), but it has limited success in our experiments (Section 4.2). To correct the mismatch between the stated and sampled distributions, we must therefore change the model's probabilities themselves.

Policy optimization (Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025) is a natural way to change these probabilities. It updates a model using rewards on the model's own samples, so it suits

[1] Anonymized code: `https://anonymous.4open.science/r/sample-what-you-say`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

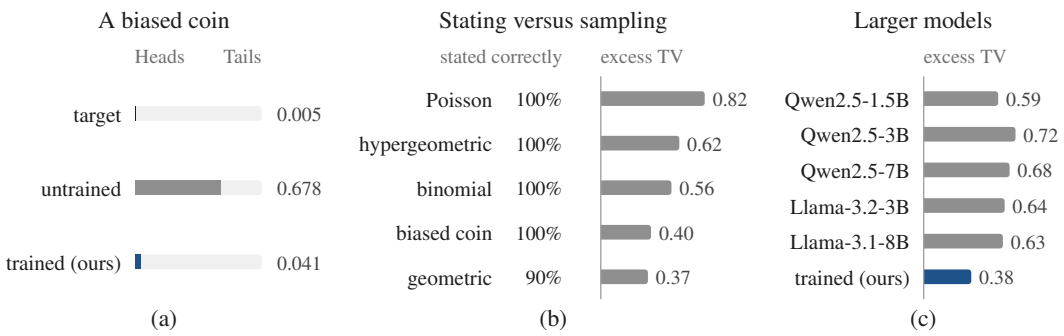

Figure 1: Instruction-tuned models state distributions they cannot sample. (a) For a coin with $P(\text{Heads}) = 0.005$, Qwen2.5-1.5B-Instruct places next-token probability 0.678 on Heads, and 0.041 after training with the witness advantage. (b) On five families, the untrained model states the family and parameters correctly for 90% to 100% of targets, yet its median excess TV is 0.37 to 0.82. Excess TV is the TV to the stated target minus the expected TV of a perfect sampler with the same number of draws. (c) Median excess TV on the 100 unseen-family targets. Larger untrained models are no better than the 1.5B model. Training the 1.5B model with the witness advantage lowers it to 0.38 (Section 4.2).

objectives that are easy to evaluate but hard to supervise explicitly. Distribution matching fits this setting, since whether a model's samples match a target distribution can only be evaluated over a group of samples. This motivates our central question:

Can policy optimization teach a language model to sample from the distributions it states, and what property of the training signal enables it to do so?

The central design choice in policy optimization is the reward. To train a model to sample from a target distribution, the reward must measure how well the model's samples match that distribution. This is challenging because a distribution is a property of many draws, whereas standard post-training rewards assign a score to each response individually. A natural approach is therefore to sample a group of outputs, compare their empirical distribution to the target distribution, and assign the resulting score to every sample in the group.

Group relative policy optimization (GRPO) (Shao et al., 2024) is well suited to this setting because it already samples a group of rollouts for each prompt. However, GRPO centers advantages within each group, so giving every rollout the same distribution-level reward makes every advantage zero (Lemma 4). One plausible workaround is to divide the rollouts into several subgroups, score each subgroup according to its empirical distribution, and center these scores across subgroups. Training with this workaround does not lower the error (Figure 2a). Supervised cross-entropy training on the stated probability mass function (PMF) can make the model match the target distribution (Zhang et al., 2024), but in our experiments it degrades the general capabilities of the model (Section 4.2). We therefore need per-rollout credit assignment that still reflects how well the group matches the target distribution.

To provide such a per-rollout training signal, we use the witness function of maximum mean discrepancy (MMD). MMD measures the discrepancy between two distributions, and its maximizing function, the witness function, identifies at each outcome how much it is over- or under-produced relative to the target (Gretton et al., 2012). On a finite outcome set with the exact-match kernel, the witness at an outcome is the difference between its model probability and its target probability. For each rollout, we estimate the negative witness function at its outcome using leave-one-out frequencies within the group, and use this estimate as the rollout's reward in GRPO. A rollout therefore receives a positive reward when its outcome is under-produced relative to the target and a negative reward when it is over-produced. Unlike a group-level reward, this signal assigns credit to individual outcomes while still reflecting the mismatch of the group as a whole. Before GRPO subtracts the group mean, the resulting update is an unbiased estimate of the negative gradient of the squared MMD between the model's outcome distribution and the target, and the subtraction adds only a bias of order $1/G$ in the group size G (Proposition 1). Section 4.2 shows that the size of this credit matters for outcomes with small target probability.

To summarize, our key contributions are as follows.

1. We show that prompting and decoding have limited success in making a model sample from the distributions it states, because the model's probabilities over outcomes are already wrong. We therefore use policy optimization to correct them (Section 4.2).
2. We introduce the witness advantage, a per-rollout advantage that enables GRPO to learn this task, and prove that, in expectation, training with it moves the model toward the target distribution. We also show when simpler rewards fail (Section 3).
3. We show that one to three GPU-hours of training substantially improve sampling on the training distributions, on unseen parameters and distribution families, and on human-opinion distributions, while largely preserving general capabilities (Section 4).

Related Work. Prior work closest to ours also scores each GRPO rollout using the other rollouts in its group. GAPO (Anschel et al., 2025) is defined only for a uniform target, and it penalizes a rollout by the fraction of the group that shares its outcome. Because this fraction always counts the scored rollout among those sharing its outcome, it is a biased estimate of the model's probability of that outcome. Distribution-aware reward (DAR) (Park et al., 2026) is designed for numeric regression and scores a group of predicted numbers against the correct number for each input. Its target is a single number, whereas ours is a probability distribution. Jang et al. (2026) list rewards scored over groups of samples as future work, but do not define or evaluate such a reward. Appendix A discusses further related work.

2 PRELIMINARIES

Notation. A prompt describes a target distribution q over a finite set of outcomes $\mathcal{X}$ and asks the model to draw one outcome from it. The prompt specifies the family of q, its parameters, and usually its PMF (Appendix F). Given the prompt, the model generates a text response y with probability $P_\theta(y)$. A parser φ extracts from each response y the outcome $\varphi(y)$ that it states. The model's *outcome distribution*, $\pi_\theta(x) = \sum_{y:\,\varphi(y)=x} P_\theta(y)$, is the probability that its response states outcome x. Our goal is to make π_θ match q. Some responses do not state a valid outcome, for example an outcome with extra words around it or a number outside the support of q. The parser maps them to the symbol $\perp$ (Appendix F). We add $\perp$ to $\mathcal{X}$ and set $q(\perp) = 0$.

Error measure. We measure how far a distribution p over outcomes is from the target q with the total variation (TV) distance, $\mathrm{TV}(p, q) = \frac{1}{2}\sum_x |p(x) - q(x)|$. TV is the largest difference between the probabilities that p and q assign to the same set of outcomes, and it ranges from 0 to 1. We usually cannot compute the model's outcome distribution exactly, so we estimate its TV from the empirical distribution of n sampled outcomes (Section 4.1).

GRPO. We train with GRPO (Shao et al., 2024). For each prompt, GRPO samples a group of G rollouts from the model, with responses $y_1, \ldots, y_G$. We write $x_i = \varphi(y_i)$ for the outcome of rollout i, and $\hat{p}(x) = \#\{i : x_i = x\}/G$ for the fraction of rollouts whose outcome is x. GRPO scores each rollout with a scalar reward r_i and sets its *advantage* to the reward minus the group mean,

$$\tilde{A}_i = r_i - \mathrm{mean}(r_1, \ldots, r_G). \tag{1}$$

Like Liu et al. (2025), we omit the original GRPO's division by the standard deviation of the rewards (Appendix F.4). Training increases the probability of rollouts with a positive advantage and decreases the probability of rollouts with a negative advantage. Ignoring GRPO's PPO-style clipping (Schulman et al., 2017) and its KL penalty, the update direction is proportional to $\frac{1}{G}\sum_i \tilde{A}_i \nabla_\theta \log P_\theta(y_i)$, where $\nabla_\theta \log P_\theta(y_i)$ is the direction in parameter space that most increases the log-probability of response y_i, and the update scales this direction by $\tilde{A}_i$. Since the reward affects the update only through $\tilde{A}_i$, we can change what the model learns by changing only the reward while keeping the rest of GRPO as is.

Per-rollout rewards. Equation 1 requires a reward r_i for every rollout. In math reasoning, the task GRPO was introduced for, this reward checks whether the rollout's answer is correct. Sampling from a distribution has no correct answer to check. Every outcome with positive target probability is a valid draw. What matters is how often the model produces each outcome. An outcome is over-produced if the model's probability of it is above its target probability, and under-produced if it is below.

3 METHOD

GRPO needs a reward for each rollout, but whether the model samples correctly depends on the group as a whole (Section 2). Section 3.1 derives such a reward, the witness advantage, from the maximum mean discrepancy. Section 3.2 shows that, in expectation, training with this advantage performs gradient descent on the distance to the target distribution. Section 3.3 shows that the group-scalar reward optimizes a biased objective and that the sign witness does not train outcomes with small target probability toward their targets.

3.1 THE WITNESS ADVANTAGE

To train a model to sample from a target distribution, we need a function that tells us, for each outcome, whether the model produces it too often or too rarely, and by how much. Given such a function, each rollout i receives an advantage computed from the function's value at its own outcome x_i. The advantage is positive if x_i is under-produced and negative if x_i is over-produced. Training then produces more of the under-produced and less of the over-produced outcomes, shifting the outcome distribution toward the target distribution. We also want this function to come from a distance between the model's outcome distribution and the target, so that we can prove training reduces this distance. The maximum mean discrepancy (MMD) of Gretton et al. (2012) gives such a distance, together with a function of this kind, called the witness function.

MMD measures the discrepancy between two distributions as the largest difference in the expectation of a test function under them. Let p and q be distributions on an outcome space $\mathcal{X}$, and let k be a kernel with reproducing kernel Hilbert space $\mathcal{H}$. The MMD is

$$\mathrm{MMD}(p, q) = \sup_{\|f\|_{\mathcal{H}} \le 1} \left(\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim q}[f(x)]\right).$$

The maximizing function $f^\star$, called the *witness function*, is proportional to the difference between the mean embeddings of p and q:

$$f^\star \propto \mu_p - \mu_q, \qquad \mu_p := \mathbb{E}_{x\sim p}[k(x, \cdot)], \quad \mu_q := \mathbb{E}_{x\sim q}[k(x, \cdot)].$$

Because our outcome space is finite, we use the exact-match kernel $k(x, x')$, which is 1 if $x = x'$ and 0 otherwise, and for which the mean embedding is the PMF itself. Let $\pi_\theta(x)$ be the model probability of outcome x, and let $q(x)$ be its target probability. Then

$$\mathrm{MMD}^2(\pi_\theta, q) = \sum_{x\in\mathcal{X}} \big(\pi_\theta(x) - q(x)\big)^2, \qquad f^\star(x) \propto \pi_\theta(x) - q(x). \tag{2}$$

Thus, MMD^2 is the squared Euclidean distance between the model and target PMFs, and it is zero only when $\pi_\theta = q$. The witness at x is the model's excess probability on that outcome, $\pi_\theta(x) - q(x)$. It is positive when x is over-produced, negative when it is under-produced, and its magnitude is how far the model's probability of x is from its target.

The witness depends on the model's outcome distribution π_θ, which we cannot compute exactly. Instead, we estimate it from the group of G rollouts, where x_i is the outcome of rollout i. The simplest estimate of $\pi_\theta(x_i)$ is the fraction of rollouts in the group whose outcome is x_i. This fraction counts rollout i itself, and counting it adds a term of order $1/G$ to the expected update. The added term increases the probability that two draws from the model have the same outcome, pushing the model to concentrate its probability mass on fewer outcomes (Appendix C.1). We remove this term by leaving rollout i out. The *leave-one-out frequency* $\hat{p}_{-i}(x_i)$ is the fraction of the other $G-1$ rollouts whose outcome is x_i. Applying the same correction gives an unbiased estimator of MMD^2 (Gretton et al., 2012; Bińkowski et al., 2018).

We then take the negative of the witness, so an under-produced outcome receives a positive advantage and an over-produced outcome receives a negative advantage. Rollout i with outcome x_i receives the *witness advantage*

$$A_i = 2\left(q(x_i) - \hat{p}_{-i}(x_i)\right), \tag{3}$$

where $q(x_i)$ is the target probability of the rollout's outcome, and $\hat{p}_{-i}(x_i)$ is how often the other rollouts produced that outcome. A rollout receives a positive advantage when its outcome appears

less often among the other rollouts than its target probability, and a negative advantage when it appears more often. The size of the advantage is twice the mismatch at that outcome. The factor of two makes the expected update equal to the negative gradient of MMD^2 (Section 3.2).

The witness advantage needs only the outcome counts of the group. We use A_i as the reward r_i in GRPO and leave the rest of GRPO unchanged. GRPO therefore still subtracts the group mean (Equation 1) and trains on $\tilde{A}_i = A_i - \bar{A}$, where $\bar{A}$ is the mean of A_i over the group. Throughout the paper, A_i denotes an advantage before this subtraction and $\tilde{A}_i$ the advantage after it.

One could instead train on the total variation distance, our error measure (Section 2). It has the same form, $2\,\mathrm{TV}(p, q) = \sup_{|f|\le 1}(\mathbb{E}_p f - \mathbb{E}_q f)$, and the same minimizer, $\pi_\theta = q$. However, its witness is only the sign of $\pi_\theta - q$. The sign indicates whether a rollout's outcome is too frequent or too rare, but not by how much. Section 3.3 shows that without the size, training does not reach the target on outcomes with small target probability.

3.2 The Witness Advantage Trains Toward the Target Distribution

Recall from Section 2 that, ignoring clipping and the KL penalty, a GRPO step moves the parameters along $\frac{1}{G}\sum_i \tilde{A}_i \nabla_\theta \log P_\theta(y_i)$. Each response y_i gains log-probability in proportion to $\tilde{A}_i$. Let U be this direction with $\tilde{A}_i$ replaced by the witness advantage A_i of Equation 3, and let U_c be the direction GRPO applies, with $\tilde{A}_i = A_i - \bar{A}$. Expectations are over the G rollouts of a group, which are independent draws from the model (Appendix B). Proposition 1 computes the expected update.

Proposition 1 (The witness advantage follows the negative gradient of MMD^2)**.** *Let every rollout in a group of $G \ge 2$ rollouts receive the witness advantage. Then $\mathbb{E}[U] = -\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q)$, and $\mathrm{MMD}^2(\pi_\theta, q) = 0$ only at $\pi_\theta = q$. After the group mean is subtracted,*

$$\mathbb{E}[U_\mathrm{c}] = -\tfrac{G-1}{G}\,\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q) + \tfrac{1}{G}\,\nabla_\theta \|\pi_\theta\|^2, \tag{4}$$

where $\|\pi_\theta\|^2 = \sum_x \pi_\theta(x)^2$ is the probability that two independent draws from π_θ give the same outcome (proof in Appendix C).

Before the group mean is subtracted, the expected update is the negative gradient of MMD^2 at every group size. In expectation, training is gradient descent on the squared distance between the model's outcome distribution and the target, and this distance is zero only at the target. Subtracting the group mean multiplies the gradient by $(G-1)/G$. It also adds a term of weight $1/G$ that increases $\|\pi_\theta\|^2$, and so pushes the model to concentrate its probability on fewer outcomes.

Because the concentration term has weight only $1/G$, it shifts the minimizer of the training objective only slightly away from the target. If the model can assign any probability to each outcome and $G \ge 3$, the shift is at most $1/(G-2)$ in TV, which is 0.016 at $G = 64$ (Appendix C.2).

3.3 Where Simpler Rewards Fail

With the witness advantage, rollouts in the same group receive different advantages, because each advantage depends on the rollout's outcome. We call this *per-outcome credit*. Its sign says whether the outcome is under- or over-produced, and its size grows with the mismatch. To test whether training needs advantages that vary across outcomes and reflect the size of the mismatch, we analyze two simpler rewards, each of which removes one of these properties.

Group-scalar reward. The most direct way to reward a group for matching the target distribution is to score the group by $-\mathrm{TV}(\hat{p}, q)$, where $\hat{p}$ is the empirical distribution of its outcomes, and to give this score to every rollout in the group (Section 1). We call this the *group-scalar reward*. It gives GRPO no training signal, because every rollout in the group receives the same reward and subtracting the group mean leaves all advantages at zero (Lemma 4, Appendix D). To obtain a nonzero signal, we score several groups per prompt and subtract the mean score across groups. Specifically, we sample 256 rollouts for one prompt and split them into four subgroups of 64. Every rollout receives the score of its subgroup minus the mean score of the four subgroups.

This repaired reward optimizes a biased objective. In expectation, training with it performs gradient descent on $\mathbb{E}[\mathrm{TV}(\hat{p}, q)]$, the expected TV between the target and the empirical distribution of a

group of G draws from the model (Proposition 3 in Appendix D). This objective is not the model's own error $\mathrm{TV}(\pi_\theta, q)$. Even a model that samples exactly from q has an expected TV of order $\frac{1}{\sqrt{G}} \sum_x \sqrt{q(x)(1-q(x))}$ when $q(x) \geq 1/G$ on the support of q, because G draws do not reproduce q exactly. The minimizers of the objective are therefore only guaranteed to be within this distance of q, and for some targets, the objective is lower at a model that never produces outcomes with $q(x) < 1/G$ than at q itself.

Sign witness. The *sign witness* keeps per-outcome credit and removes its size. It gives each rollout the sign of its witness advantage, $\mathrm{sign}(q(x_i) - \hat{p}_{-i}(x_i)) \in \{-1, 0, +1\}$. The model then learns whether each outcome is too rare or too frequent among the other rollouts, but not by how much. To state the properties of the sign witness, we compare it with the value it would take if the group estimated π_θ exactly. We call $\mathrm{sign}(q(x) - \pi_\theta(x))$ the *population sign* of outcome x. We call x *low-mass* if $0 < q(x) < 1/(G-1)$. Since a leave-one-out frequency is computed from $G-1$ rollouts, $1/(G-1)$ is the smallest nonzero value it can take. Proposition 2 shows that the sign witness trains the model toward the target, except on low-mass outcomes.

Proposition 2 (The sign witness follows the negative gradient of total variation)**.** *Let every rollout in a group of $G \geq 2$ rollouts receive the sign witness. Then the following hold (proof in Appendix E).*

(i) The population sign is the negative of the witness of total variation. If $q(x) \neq \pi_\theta(x)$ for every outcome x, the expected update with the population sign is $-\nabla_\theta \, 2\mathrm{TV}(\pi_\theta, q)$.

(ii) Given that a rollout has an outcome x with $\delta = q(x) - \pi_\theta(x) \neq 0$, its sign witness and the population sign differ with probability at most $\exp(-2(G-1)\delta^2)$.

(iii) A rollout with a low-mass outcome x receives expected advantage $2(1-\pi_\theta(x))^{G-1} - 1$, independent of $q(x)$. If the model can assign any probability to each outcome, every stationary point of the expected update therefore gives the same probability to all low-mass outcomes that it does not set to zero. In particular, $\pi_\theta = q$ is not a stationary point when two low-mass outcomes have different target probabilities.

Parts (i) and (ii) show that, while the model is still far from the target, the sign witness trains it toward the target. With the population sign, the expected update is the negative gradient of twice the TV. The sign computed from the group matches the population sign with high probability, except at outcomes where $q(x)$ and $\pi_\theta(x)$ differ by less than about $1/\sqrt{G}$.

Part (iii) shows that the sign witness does not train the probability of a low-mass outcome toward its target. The leave-one-out frequency of a low-mass outcome is either 0, which is below its target probability, or at least $1/(G-1)$, which is above it. The sign therefore depends only on whether the outcome appeared among the other $G-1$ rollouts, not on how much probability the outcome should have. As a result, an outcome with target probability 0.001 keeps a positive expected advantage until the model assigns it about $\ln 2/(G-1) = 0.011$ at $G = 64$, about ten times its target (Appendix E). Section 4.2 checks these analyses in training.

4 Experiments

We first describe the targets, the metric, and the baselines (Section 4.1). We then train on synthetic distribution families, where the target is known exactly and we control which parameters and families the model sees in training (Section 4.2). On these targets, we compare the witness advantage with the baselines on the training targets, then test whether training transfers to unseen parameters and families, and then compare it with the two simpler rewards of Section 3.3. Lastly, we test the witness advantage on two tasks closer to real applications (Section 4.3).

4.1 Setup

Targets. We use six families of discrete distributions from the Spectrum Suite (Sorensen et al., 2026): biased coin, binomial, geometric, Poisson, hypergeometric, and Zipf. Each family consists of 20 targets that differ in their parameters. The training set has 16 targets from each family except hypergeometric, 80 in all. The *unseen parameters* are the four remaining targets of each training family. The *unseen families* are 100 targets from five families not used in training: hypergeometric, number of empty boxes, discrete triangular, maximum of dice, and logarithmic series (Appendix F).

Table 1: On the 100 targets from five distribution families never seen in training, the witness advantage has the lowest median excess TV and keeps general capabilities. The oracle temperature is tuned on each target.

	sampling error	capability	
	excess TV	MMLU	IFEval
untrained model	0.589	0.584	0.512
verbalized sampling	0.518	0.584	0.512
oracle temperature	0.478	0.584	0.512
supervised cross-entropy	0.406	0.482	0.281
witness advantage (ours)	0.376	0.583	0.528

A model cannot know the possible outcomes of a family it has never seen, so the prompts for these targets list the valid outcomes.

Metric. To evaluate a model on a target, we sample n responses at temperature 1, with $n = 500$ (up to 5,000 for unseen parameters), and compute the TV between q and the empirical distribution $\hat{p}_n$ of the parsed outcomes. Unlike training (Section 2), evaluation leaves unparseable responses out of $\hat{p}_n$ and reports their rate separately.

Even a *perfect sampler*, one that draws exactly from q, has positive TV between its empirical distribution and q after any finite number of draws. This TV is larger for targets with more outcomes and for smaller n, so raw TV values cannot be compared across targets. We therefore report *excess TV*, the measured TV minus the perfect sampler's expected TV at the same n, which we estimate by Monte Carlo. An excess TV of 0 means the model samples as well as a perfect sampler. We also measure general capabilities with MMLU (Hendrycks et al., 2021) and IFEval (Zhou et al., 2023). For each benchmark, we compute a 95% bootstrap confidence interval for the untrained model's score. A trained score is within the noise band if it lies inside this interval.

Implementation details. All GRPO runs fine-tune Qwen2.5-1.5B-Instruct in TRL on one 48 GB GPU, with group size $G = 64$, 256 rollouts per step, learning rate 2×10^{-6}, KL weight 0.02, and temperature 1. A main run on the synthetic distributions has 600 steps and takes about one GPU-hour (Appendix F.4).

Baselines. We compare against three kinds of baselines (details in Appendices F.4 and G.3). *Simpler rewards* replace only the reward, with the group-scalar reward or the sign witness of Section 3.3. *Supervised training* minimizes the cross-entropy $-\sum_x q(x) \log \pi_\theta(x)$ to the stated PMF on the same targets (Zhang et al., 2024). Computing this loss requires the model's probability of every outcome the target can produce. *Training-free baselines* change only the prompt or the decoding: verbalized sampling (Zhang et al., 2025a), seed conditioning (Nagarajan et al., 2025), renormalization onto the valid outcomes, and temperature scaling.

4.2 Results on Synthetic Distributions

Training with the witness advantage makes the model sample much closer to the target distributions, on the training targets and on unseen parameters and unseen families (Table 4 in Appendix G). On the five families never seen in training, it has the lowest error of all compared methods and keeps general capabilities (Table 1).

Witness advantage fits the training targets and keeps capability. Training removes 79% of the excess TV on the non-Zipf training targets, from 0.477 to 0.101. MMLU and IFEval stay inside their noise bands, and fewer than 1% of responses are unparseable. The strongest baseline is supervised cross-entropy training on the target PMF. It fits the training targets better than the witness advantage, but it drops MMLU by 10 points and IFEval by 23 points relative to the untrained model. The witness advantage changes them by only 0.1 and 1.6 points. Most of the remaining error of the witness advantage is on binomial and Poisson targets, where the draws are far more spread out than the target (Appendix G.4).

Training-free baselines start from wrong probabilities. We evaluate the training-free baselines on the unseen families (Tables 1 and 8). The best of them, an oracle temperature per target chosen with knowledge of q, removes only 19% of the excess TV. These baselines change only the prompt or the

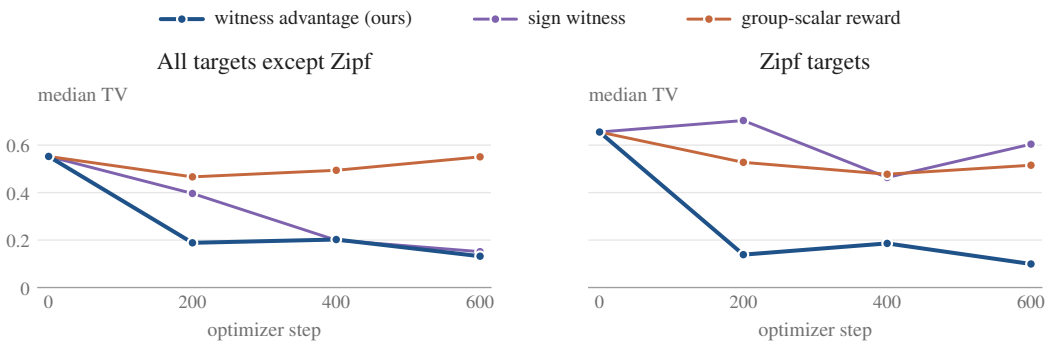

Figure 2: Median TV during training on (a) the 64 training targets outside the Zipf family and (b) the 16 Zipf targets, which have many low-mass outcomes. Outside Zipf, the sign witness trains as well as the witness advantage, and the group-scalar reward ends where the untrained model starts. On Zipf targets, only the witness advantage brings the TV close to zero.

decoding, so they start from the untrained model's probabilities over outcomes. These probabilities are wrong even though the model knows the target (Figure 1b). For the 27 training targets with at most 12 outcomes, we compute the untrained model's outcome distribution exactly from its log-probabilities. Its median TV to the target is 0.37, so the error is present before any outcome is drawn and is not sampling noise. Training with the witness advantage lowers this exact TV to 0.09. A decoding rule can only reshape the wrong probabilities. Temperature can only sharpen or flatten them, and renormalizing onto the valid outcomes does not help (0.600 against 0.589). Verbalized sampling also fails, because the probabilities the model writes out are as wrong as its draws, with a median TV of 0.71 per parsed list. Appendix G.3 gives the other baselines and the larger untrained models.

The improvement transfers to unseen parameters and families. Training removes 67% of the excess TV on the 16 unseen-parameter targets outside Zipf, from 0.483 to 0.161. Cross-entropy transfers better here (0.094), at the capability cost above.

The trained model also samples better from the five families it never saw in training, and removes 36% of their excess TV (Table 1). Supervised cross-entropy was trained on the same prompts, so the comparison in the table is like for like. These training prompts, unlike the unseen-family prompts, do not list the valid outcomes. A retrain on prompts that list them still removes 24%, more than the 19% of the oracle temperature, and improves every unseen family (Appendix G). Most of the improvement does not come from copying the listed outcomes. When we remove the list from the evaluation prompts, three quarters of the improvement remains (Appendix G.5).

Training needs per-outcome credit of the right size. Section 3.3 shows that the group-scalar reward optimizes a biased objective, and predicts that the sign witness trains like the witness advantage except on low-mass outcomes. With the same data and compute, the group-scalar reward does not lower the median TV on the non-Zipf training targets. It ends at 0.55, where the untrained model starts (Figure 2a), although some families improve (Table 5). On unseen parameters, the witness advantage has lower TV than the group-scalar reward in each of three seeds (Table 6). On unseen families the two rewards are closer, and their intervals overlap at every group size in a sweep (Table 7).

The sign witness trains as well as the witness advantage on every family except Zipf. Its final median TV on the non-Zipf training targets (Figure 2a) is 0.151, against 0.132 for the witness advantage. On 36 unseen targets outside Zipf, its paired difference from the witness advantage is indistinguishable from zero (Table 6). Zipf targets have many low-mass outcomes, those with target probability below $1/(G-1) \approx 0.016$, which the sign witness does not train toward their targets (Proposition 2(iii)). On the Zipf training targets, the median TV ends at 0.60 with the sign witness and at 0.10 with the witness advantage (Figure 2b). The size of the credit is therefore needed for low-mass outcomes.

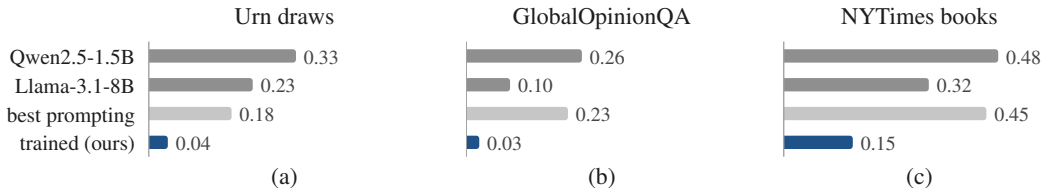

Figure 3: Training on stated opinion distributions beats prompting and a larger model, and it transfers to a task never seen in training. Bars are median excess TV to the target distribution. (a, b) Test prompts of the two training tasks. (c) The NYTimes task, never seen in training. Best prompting is the better of verbalized sampling and in-context steering, which puts ten draws from the target in the prompt. Trained values are means over three seeds.

4.3 Results on Tasks Closer to the Real World

Along with synthetic distributions, we also test our method on tasks closer to the real world, namely opinion distributions and structured outputs, with the same error measure and training procedure. Appendix H gives the full setup and results.

Training transfers to test prompts and to an unseen task. Three tasks from the Spectrum Suite (Sorensen et al., 2026) state a distribution over at most six answer options and ask for one draw. Urn draws state color proportions, GlobalOpinionQA (Durmus et al., 2023) gives country-level answer distributions to opinion questions, and the NYTimes task gives book preferences by demographic group (Meister et al., 2025). We train on urn and GlobalOpinionQA prompts and evaluate on test prompts of both (Appendix H), on the NYTimes task, never seen in training, and on an implicit variant of the test prompts. The implicit variant removes the stated probabilities, so the model must infer the distribution from the context, such as the balls in the urn or the respondent's country.

Training lowers the error on every evaluation set (Figure 3 and Table 10). Across three seeds, it removes 80% to 92% of the excess TV on urn and GlobalOpinionQA test prompts, 61% to 72% on the unseen NYTimes task, and 56% to 69% on the implicit variant. It also has lower excess TV than the best prompting method on every set of Figure 3. On test prompts, the best temperature over $\{0.7, 1.0, 1.3, 1.5\}$ reaches 0.25 on both tasks, against 0.04 and 0.03 after training. On every evaluation set, the trained 1.5B model is also closer to the stated distributions than the untrained Llama-3.1-8B-Instruct. This training has a cost. MMLU falls by 2.1 points, outside its noise band, while IFEval stays inside its band.

Training also improves structured outputs. A task adapted from sibling discovery (Nagarajan et al., 2025) names a parent and its children and asks for one pair of siblings, so the model must build the valid pairs before sampling among them. On targets uniform over the pairs, training removes 35% and 39% of the excess TV in two seeds, compared with 5% for the best decoding baseline. On targets that state a non-uniform probability for each pair, it removes 47% (Appendix H).

5 Discussion

We show that policy optimization, specifically GRPO, can reduce the mismatch between a language model's stated and sampled distributions, but only when the training signal assigns credit at the level of individual outcomes. A scalar reward for an entire group is insufficient: under GRPO, it is eliminated by group-relative centering. We propose the *witness advantage*, which instead converts the group-level distribution mismatch into per-rollout credit, increasing under-produced outcomes and decreasing over-produced ones. Training with this credit substantially improves sampling on the synthetic training distributions, transfers to unseen parameters and families, and works on tasks closer to the real world, while largely preserving general capabilities.

More broadly, our results suggest that group-defined objectives require more than a group-level score: they require a mechanism for attributing the group's error to individual rollouts. We study this principle for finite outcome spaces, though the witness advantage can be defined for any kernel. An important next step is to apply it to free-form text, where a target distribution may be represented by example responses and a kernel can encode similarity between responses. This would extend distribution-matching post-training to settings in which the outcome space cannot be enumerated.

AI USE STATEMENT

The authors developed the research questions and core conceptual framework and directed the methodology and interpretation. Generative AI assisted with proof writing, literature review, code implementation, figure preparation, and prose polishing. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendices C, D, and E give complete proofs of the formal results. Section 4 summarizes the experiments, and Appendix F gives the prompts, the parser, the targets, the training settings, and the statistical procedures. The anonymized code at `https://anonymous.4open.science/r/sample-what-you-say` contains the implementation, the training and evaluation scripts, and evaluation results.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. arXiv:2402.14740.

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. Group-aware reinforcement learning for output diversity in large language models. *arXiv preprint arXiv:2511.12596*, 2025.

Davide Baldelli, Sruthi Kuriakose, Maryam Hashemzadeh, Amal Zouaq, and Sarath Chandar. Probabilistic calibration is a trainable capability in language models. *arXiv preprint arXiv:2605.11845*, 2026.

Daniel Berend and Aryeh Kontorovich. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & Probability Letters*, 83(4):1254–1259, 2013.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1801.01401.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*, 2023.

Jingchu Gai, Guanning Zeng, Huaqing Zhang, and Aditi Raghunathan. Differential smoothing mitigates sharpening and improves LLM reasoning. *arXiv preprint arXiv:2511.19942*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Xiangming Gu, Soham De, Michalis Titsias, Larisa Markeeva, Petar Veličković, and Razvan Pascanu. The illusion of stochasticity in LLMs. In *Conference on Language Modeling (COLM)*, 2026. arXiv:2604.06543.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Esmeralda S. Whitammer. Amortizing intractable inference in large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.04363.

Audrey Huang, Adam Block, Dylan J. Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. *arXiv preprint arXiv:2412.01951*, 2024.

Pengrun Huang, Chhavi Yadav, Ruihan Wu, and Kamalika Chaudhuri. Alignment defends LLMs from property inference attacks. *arXiv preprint arXiv:2606.10217*, 2026.

Chaemin Jang, Jihee Kim, and Dongman Lee. Instruction-tuned language models cannot sample from distributions they can describe. *arXiv preprint arXiv:2607.25292*, 2026.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. arXiv:2510.22954.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2012.11635.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06452.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *Deep RL Meets Structured Prediction Workshop at ICLR*, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Long Li, Zhijian Zhou, Jiaran Hao, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.07430*, 2025a.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025b.

Xiaozhe Li, Yang Li, Xinyu Fang, Shengyuan Ding, Peiji Li, Yongkang Chen, Yichuan Ma, Tianyi Lyu, Linyang Li, Dahua Lin, Qipeng Guo, Qingwen Liu, and Kai Chen. Beyond mode collapse: Distribution matching for diverse reasoning. *arXiv preprint arXiv:2605.19461*, 2026. Accepted at ICML 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025.

Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. Benchmarking distributional alignment of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025. arXiv:2411.05403.

Kou Misaki and Takuya Akiba. String seed of thought: Prompting LLMs for distribution-faithful and diverse generation. *arXiv preprint arXiv:2510.21150*, 2025. Accepted at ICLR 2026.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Sonia K. Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025. arXiv:2411.04427.

Vaishnavh Nagarajan, Chen Henry Wu, Charles Ding, and Aditi Raghunathan. Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. arXiv:2504.15266.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.05196.

Jungsoo Park, Hyungjoo Chae, Ethan Mendes, Jay DeYoung, Varsha Kishore, Wei Xu, and Alan Ritter. Distribution-aware reward: Reinforcement learning over predictive distributions for LLM regression. *arXiv preprint arXiv:2605.20740*, 2026.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. arXiv:2303.17548.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. Distributional preference learning: Understanding and accounting for hidden context in RLHF. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2312.08358.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.05070.

Taylor Sorensen, Benjamin Newman, Jared Moore, Chan Young Park, Jillian Fisher, Niloofar Mireshghallah, Liwei Jiang, and Yejin Choi. Spectrum tuning: Post-training for distributional coverage and in-context steerability. In *International Conference on Learning Representations (ICLR)*, 2026.

Jacob Mitchell Springer, Madhu Advani, Lukas Aichberger, Arwen Bradley, Eran Malach, Omid Saremi, Sinead Williamson, Preetum Nakkiran, Etai Littwin, and Aditi Raghunathan. Annotations mitigate post-training mode collapse. *arXiv preprint arXiv:2605.09995*, 2026. Accepted at ICML 2026.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Katherine Van Koevering and Jon Kleinberg. How random is random? Evaluating the randomness and humaness of LLMs' coin flips. *arXiv preprint arXiv:2406.00092*, 2024.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. `https://github.com/huggingface/trl`, 2020.

Christian Walder and Deep Karkhanis. Pass@K policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.15201.

Peter West and Christopher Potts. Base models beat aligned models at randomness and creativity. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2505.00047.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Chen Henry Wu, Sachin Goyal, and Aditi Raghunathan. Mode-conditioning unlocks superior test-time scaling. *arXiv preprint arXiv:2512.01127*, 2025.

Tim Z. Xiao, Johannes Zenn, Zhen Liu, Weiyang Liu, Robert Bamler, and Bernhard Schölkopf. Flipping against all odds: Reducing LLM coin flip bias via verbalized rejection sampling. *arXiv preprint arXiv:2506.09998*, 2025.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. *arXiv preprint arXiv:2510.01171*, 2025a. Accepted at ICML 2026.

Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, J. Zico Kolter, and Daphne Ippolito. Forcing diffuse distributions out of language models. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.10859.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*, 2025b.

Minda Zhao, Yilun Du, and Mengyu Wang. Large language models are bad dice players: LLMs struggle to generate random numbers from statistical distributions. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2601.05414.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 60704–60748, 2024. arXiv:2404.17546.

Hongyi Zhou, Kai Ye, Erhan Xu, Jin Zhu, Ying Yang, Shijin Gong, and Chengchun Shi. Demystifying group relative policy optimization: Its policy gradient is a U-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, Qizheng Zhang, Lin Chen, Fanghao Shao, Bo Xue, Yunchong Song, Zhenjie Yang, Ganqu Cui, Ning Ding, Jianfeng Gao, Xiaodong Liu, Bowen Zhou, Hongyuan Mei, and Zhouhan Lin. FlowRL: Matching reward distributions for LLM reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.15207.

APPENDIX

A RELATED WORK 16

B NOTATION AND TOOLS FOR THE PROOFS 17

B.1 Notation 17
B.2 From Responses to Outcomes 17
B.3 Two Lemmas on the Score 18

C PROOFS FOR THE WITNESS ADVANTAGE 19

C.1 Counting the Rollout Itself 21
C.2 Where the Centered Update Moves the Model 22
C.3 A Worked Example 23
C.4 Numerical Checks 23

D PROOFS FOR THE GROUP-SCALAR REWARD 24

D.1 Setting and Statements 24
D.2 Proofs 25
D.3 When the Bias Matters 27
D.4 Numerical Checks 28

E PROOFS FOR THE SIGN WITNESS 28

E.1 Part (i): The Population Sign Is the Witness of Total Variation 28
E.2 Part (ii): The Sign Computed From the Group Rarely Differs 29
E.3 Part (iii): Low-Mass Outcomes 29
E.4 Simulating the Sign Witness on Zipf Targets 31
E.5 Connection to Property-Ratio Alignment 31

F EXPERIMENTAL SETUP 31

F.1 Prompts 31
F.2 Parser and Invalid Outputs 32
F.3 Targets 32
F.4 Training 33
F.5 Evaluation and Statistics 34

G ADDITIONAL RESULTS ON SYNTHETIC DISTRIBUTIONS 34

G.1 Results per Family 34
G.2 Comparing the Three Rewards 34
G.3 Training-Free Baselines 35
G.4 Where the Remaining Error Is 37
G.5 Copying the Listed Outcomes 38
G.6 Variety of the Training Targets 38

H OPINION DISTRIBUTIONS AND STRUCTURED OUTPUTS 39

A Related Work

Sampling failures of language models. Language models are poor random number generators (Van Koevering & Kleinberg, 2024; Zhang et al., 2024; Zhao et al., 2026), and aligned models do worse than their base models at random number generation and mixed-strategy games (West & Potts, 2025). Jang et al. (2026) show that instruction-tuned models collapse to one answer when asked for a draw, that the collapse widens at every stage of post-training, and that the same models describe the target correctly in one call. Xiao et al. (2025) find the same mismatch between describing and sampling a Bernoulli distribution, and Meister et al. (2025) and Gu et al. (2026) find it on opinion distributions and in frontier models. One line of remedies acts at inference time. These methods prompt for a seed string before the draw (Misaki & Akiba, 2025), randomize surface features of the prompt across calls (Jang et al., 2026), or have the model accept or reject proposed draws in natural language, a verbalized form of rejection sampling (Xiao et al., 2025). Another line fine-tunes the model with a supervised loss. Zhang et al. (2024) minimize the cross-entropy between the model's distribution over valid outputs and the target distribution, and our supervised baseline follows their objective (Appendix F.4). Baldelli et al. (2026) fine-tune on prompts that name a distribution family and its parameters. They train either on next-token probabilities computed from the target distribution or on completions sampled from it, and their models sample more accurately than the base models on families and parameter settings not seen in training. Jang et al. (2026) name two training-time directions that their evidence calls for, a loss that matches a stated output distribution and rewards scored over groups of samples, and do not run them. We train with policy optimization and a reward of the second kind, and we show that training needs advantages that differ across outcomes and grow with the mismatch at each outcome (Section 3.3).

Mode collapse and diversity after alignment. Post-training reduces output diversity (Kirk et al., 2024; Padmakumar & He, 2024; Murthy et al., 2025; Jiang et al., 2025; Zhang et al., 2025b). Proposed causes include typicality bias in preference data (Zhang et al., 2025a), the aggregation implicit in preference learning (Siththaranjan et al., 2024), and sharpening (Huang et al., 2024; Gai et al., 2025). In reasoning RL the collapse appears as falling entropy and pass@k (Cui et al., 2025; He et al., 2025). Remedies restore diversity without a target (He et al., 2025; Li et al., 2025b;a; Nagarajan et al., 2025; Wu et al., 2025; Springer et al., 2026). Verbalized sampling (Zhang et al., 2025a) asks for several candidates with probabilities in one call. It is evaluated mainly on diversity and on distributions the model already holds, such as pretraining frequencies, and its only instructed target is a fair die. We evaluate against arbitrary PMFs stated in the prompt. A stated target must counteract collapse while preserving a specific distribution, so diversity restoration is not a substitute. A stated PMF can require less randomness as well as more, as for the coin of Figure 1a.

Distribution matching and group-level rewards in post-training. GFlowNet fine-tuning (Hu et al., 2024) and distributional control (Khalifa et al., 2021) fine-tune a model toward one fixed target density over sequences, and twisted sequential Monte Carlo (Zhao et al., 2024) samples from such a density at inference time. Distributional control can define the density through moment constraints, such as a stated fraction of outputs with some attribute. In all three methods the target is fixed for the task and is not stated in each prompt. DMPO (Li et al., 2026) and FlowRL (Zhu et al., 2026) match distributions induced by a scalar reward and have no access to a stated PMF. Several recent works make a rollout's GRPO reward depend on its group, and ours is closest to them. GAPO (Anschel et al., 2025) rewards each rollout by one minus the signed difference between the group frequency of its output and a uniform target over the valid options. For a uniform target, this reward is an affine function of the same signed deviation as our witness advantage, with two differences. The target is uniform by construction, so a stated non-uniform PMF cannot be expressed, and the frequency counts the rollout itself, the self-count that the leave-one-out frequency removes (Appendix C.1). Distribution-aware reward (Park et al., 2026) scores a group of numeric rollouts with a proper scoring rule against a scalar label and pays each rollout its leave-one-out marginal contribution to the group score. Its target is one label and its credit is a difference of two group scores. Our target is a stated PMF and our credit is the witness in closed form. The closed form lets us prove unbiasedness and state what mean-centering does. On a binary alphabet, the property-ratio reward of Huang et al. (2026) equals the sign witness computed from the full group, up to centering and scale, on every group whose ratio lies outside their tolerance band, and their multi-class variant extends this correspondence to larger alphabets (Appendix E.5). Pass@k policy optimization (Walder

& Karkhanis, 2025) also turns an objective defined on a set of rollouts into per-rollout rewards. It derives unbiased estimators of pass@k and of its gradient, so its objective is the joint success of a set of rollouts and not a match to a distribution. Spectrum Tuning (Sorensen et al., 2026) trains for distributional coverage by supervised fine-tuning on samples, and the pluralistic-alignment program (Sorensen et al., 2024) names distributional calibration as a goal. GRPO's policy gradient can be written as a U-statistic (Zhou et al., 2026), the same form as the unbiased MMD estimator on which the witness advantage is built.

B Notation and Tools for the Proofs

Appendices C, D, and E prove the results of Section 3. This appendix collects what the three proofs share. We fix the notation, show that the proofs may work with outcomes in place of text responses, and state two lemmas about the score function that carry most of the algebra.

B.1 Notation

All quantities refer to one fixed prompt, and the following symbols have the same meaning in all three proofs.

- $\mathcal{X}$ is the finite set of outcomes. It contains the invalid outcome $\perp$. The target q is the distribution on $\mathcal{X}$ stated in the prompt, and $q(\perp) = 0$.
- The model generates response y with probability $P_\theta(y)$, where θ are the model parameters and $\mathcal{Y}$ is the finite set of responses. The parser φ maps a response to its outcome $\varphi(y) \in \mathcal{X}$.
- $\pi_\theta(x) = \sum_{y:\varphi(y)=x} P_\theta(y)$ is the model's outcome distribution (Section 2). It is differentiable in θ, and $\pi_\theta(x) > 0$ for every outcome x (Appendix B.2 shows why both hold for a language model).
- A group consists of $G \geq 2$ rollouts. Rollout i has response y_i and outcome $x_i = \varphi(y_i)$, and the outcomes $x_1, \ldots, x_G$ are i.i.d. from π_θ. An expectation $\mathbb{E}$ without a subscript is over the rollouts of a group, or of a batch in Appendix D.
- $s(x) = \nabla_\theta \log \pi_\theta(x)$ is the score of outcome x, the direction in parameter space that most increases the log-probability of x.
- $\mathbb{1}[\cdot]$ is the indicator of an event. The group frequency $\hat{p}(x) = \frac{1}{G}\sum_j \mathbb{1}[x_j = x]$ is the fraction of the G rollouts whose outcome is x. The leave-one-out frequency $\hat{p}_{-i}(x) = \frac{1}{G-1}\sum_{j\neq i} \mathbb{1}[x_j = x]$ is the fraction among the other $G-1$ rollouts.
- For a vector v indexed by outcomes, $\|v\| = (\sum_x v(x)^2)^{1/2}$ is the Euclidean norm and $\|v\|_1 = \sum_x |v(x)|$. In particular, $\|\pi_\theta\|^2 = \sum_x \pi_\theta(x)^2$ is the collision probability, the probability that two independent draws from π_θ give the same outcome.
- $\mathrm{TV}(p, q) = \frac{1}{2}\|p - q\|_1$ is the total variation distance. With the exact-match kernel of Section 3.1, $\mathrm{MMD}^2(p, q) = \|p - q\|^2$ (Equation 6). We write MMD^2 without arguments for $\mathrm{MMD}^2(\pi_\theta, q)$.
- A_i is the advantage of rollout i before GRPO subtracts the group mean $\bar{A} = \frac{1}{G}\sum_j A_j$, and $\tilde{A}_i = A_i - \bar{A}$ is the centered advantage. Each proof defines its own A_i and its own update. A subscript c, as in U_c, marks a quantity that belongs to the centered update.
- An outcome x is low-mass if $0 < q(x) < 1/(G-1)$ (Section 3.3).

Every expectation below is a finite sum, so gradients and expectations can be exchanged without further justification.

B.2 From Responses to Outcomes

A GRPO step moves the parameters along $\frac{1}{G}\sum_i \tilde{A}_i \nabla_\theta \log P_\theta(y_i)$, the score of each response y_i weighted by its advantage (Section 2). The proofs are much simpler if each rollout instead moves along the score $s(x_i)$ of its outcome, because then only the outcome distribution π_θ enters. The two scores are different vectors, so a single update differs between the two views. Lemma 1 shows that the expected updates are equal whenever the weights depend on the responses only through their outcomes. Every advantage in this paper has this property, including the witness advantage and the sign witness before and after centering, and the group-scalar reward.

The lemma uses three facts about our setting. A training response has at most 24 tokens (Appendix F.4), so the set $\mathcal{Y}$ of responses is finite. The softmax gives every response a probability $P_\theta(y) > 0$ that is differentiable in θ. Every outcome in $\mathcal{X}$, including $\perp$, is the parse of some response, so $\pi_\theta(x) > 0$ and the score $s(x)$ is defined.

Lemma 1 (Outcome reduction). *Let $y_1, \dots, y_N$ be i.i.d. from P_θ, let $x_i = \varphi(y_i)$, and let $w_i = a_i(x_1, \dots, x_N)$ be weights given by arbitrary functions $a_i : \mathcal{X}^N \to \mathbb{R}$. Then $x_1, \dots, x_N$ are i.i.d. from π_θ, and*

$$\mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^{N} w_i \, \nabla_\theta \log P_\theta(y_i)\Big] = \mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^{N} w_i \, s(x_i)\Big]. \tag{5}$$

Proof. We prove the two claims in turn.

Step 1: the outcomes are i.i.d. The parser acts on each response separately, so the outcomes are independent, and $\Pr[\varphi(y_i) = x] = \sum_{y:\varphi(y)=x} P_\theta(y) = \pi_\theta(x)$.

Step 2: Equation 5. Fix i and condition on the other responses y_{-i}, whose outcomes we write x_{-i}. We write $a_i(x, x_{-i})$ for a_i with x in position i. Since $P_\theta(y)\,\nabla_\theta \log P_\theta(y) = \nabla_\theta P_\theta(y)$,

$$\begin{aligned}\mathbb{E}\big[w_i \, \nabla_\theta \log P_\theta(y_i) \,\big|\, y_{-i}\big] &= \sum_{y\in\mathcal{Y}} a_i\big(\varphi(y), x_{-i}\big)\, \nabla_\theta P_\theta(y) \\ &= \sum_{x\in\mathcal{X}} a_i(x, x_{-i}) \sum_{y:\,\varphi(y)=x} \nabla_\theta P_\theta(y) = \sum_{x\in\mathcal{X}} a_i(x, x_{-i})\, \nabla_\theta \pi_\theta(x).\end{aligned}$$

The second equality groups the responses by outcome, and the third differentiates the finite sum that defines $\pi_\theta(x)$. Since $\nabla_\theta \pi_\theta(x) = \pi_\theta(x)\, s(x)$, the last expression equals $\mathbb{E}[a_i(x_i, x_{-i})\, s(x_i) \mid x_{-i}]$ with $x_i \sim \pi_\theta$. It depends on y_{-i} only through x_{-i}, which are i.i.d. from π_θ by Step 1. Taking the expectation over y_{-i} and averaging over i gives Equation 5. □

Lemma 1 says that when the weights depend only on outcomes, the expected update depends on the model only through its outcome distribution. The weights may be the advantages A_i or the centered advantages $\tilde{A}_i$. With $N = G$, the lemma covers the expected updates of Propositions 1 and 2, before and after centering, including the update with the population sign in Proposition 2(i). With $N = KG$, it covers Proposition 3(ii) in Appendix D. The proofs below therefore work with the outcome-level updates. The remaining claims, such as the standard-deviation bound of Proposition 3(i) and the expected advantage of Proposition 2(iii), concern the values of the advantages. These values are functions of the outcomes, so the claims hold in both views. The lemma is about expectations only. The response score also varies among responses with the same outcome, so the variance of an update differs between the two views.

B.3 Two Lemmas on the Score

The first lemma turns the expectation of a function times the score into a sum of gradients of outcome probabilities.

Lemma 2 (Score identities). *Let $x \sim \pi_\theta$, and let $f : \mathcal{X} \to \mathbb{R}$ be any function. Then*

$$\textit{(a)}\quad \mathbb{E}[s(x)] = 0, \qquad\qquad \textit{(b)}\quad \mathbb{E}[f(x)\, s(x)] = \sum_x f(x)\, \nabla_\theta \pi_\theta(x).$$

Proof. Since $\pi_\theta(x) > 0$, we have $\pi_\theta(x)\, s(x) = \pi_\theta(x)\, \nabla_\theta \log \pi_\theta(x) = \nabla_\theta \pi_\theta(x)$. Hence

$$\mathbb{E}[f(x)\, s(x)] = \sum_x \pi_\theta(x)\, f(x)\, s(x) = \sum_x f(x)\, \nabla_\theta \pi_\theta(x),$$

which is (b). For (a), take $f \equiv 1$ and use $\sum_x \nabla_\theta \pi_\theta(x) = \nabla_\theta \sum_x \pi_\theta(x) = \nabla_\theta 1 = 0$. □

The second lemma handles the products that appear when a leave-one-out frequency multiplies a score. Such a product has an indicator that two rollouts match, times the score of one rollout. Only a match that involves the rollout that owns the score contributes.

Lemma 3 (Pair moments). *Let $x_1, \dots, x_G$ be i.i.d. from π_θ, and let i, j, ℓ be distinct indices. Then*

$$\textit{(a)}\quad \mathbb{E}\big[\mathbb{1}[x_j = x_i]\, s(x_i)\big] = \tfrac{1}{2}\nabla_\theta \|\pi_\theta\|^2,$$
$$\textit{(b)}\quad \mathbb{E}\big[\mathbb{1}[x_\ell = x_j]\, s(x_i)\big] = 0,$$
$$\textit{(c)}\quad \mathbb{E}\big[q(x_j)\, s(x_i)\big] = 0.$$

Proof. For (a), condition on $x_i = x$. The draw x_j equals x with probability $\pi_\theta(x)$, so

$$\mathbb{E}\big[\mathbb{1}[x_j = x_i]\, s(x_i)\big] = \sum_x \pi_\theta(x) \cdot \pi_\theta(x) \cdot s(x) = \sum_x \pi_\theta(x)\, \nabla_\theta \pi_\theta(x) = \tfrac{1}{2}\nabla_\theta \sum_x \pi_\theta(x)^2.$$

The second equality uses $\pi_\theta(x)\, s(x) = \nabla_\theta \pi_\theta(x)$, the identity behind Lemma 2, and the third is the chain rule. For (b) and (c), the first factor does not involve x_i, so the expectation factorizes into the expectation of the first factor times $\mathbb{E}[s(x_i)]$, which is zero by Lemma 2(a). □

C Proofs for the Witness Advantage

Proposition 1 states that the witness advantage makes GRPO's expected update the negative gradient of MMD^2, and computes the bias that subtracting the group mean adds. We restate and prove it. We then show that counting a rollout in its own frequency strengthens the concentration term (Appendix C.1), find where the centered update moves the model (Appendix C.2), work through a small example (Appendix C.3), and list the numerical checks in the code release (Appendix C.4).

Rollout i receives the witness advantage $A_i = 2\big(q(x_i) - \hat{p}_{-i}(x_i)\big)$ of Equation 3, and GRPO trains on the centered advantage $\tilde{A}_i = A_i - \bar{A}$. Section 3.2 defines the updates U and U_c with the response scores $\nabla_\theta \log P_\theta(y_i)$. In this appendix we reuse the two symbols for the outcome-level updates

$$U = \frac{1}{G}\sum_{i=1}^{G} A_i\, s(x_i), \qquad U_\mathrm{c} = \frac{1}{G}\sum_{i=1}^{G} \tilde{A}_i\, s(x_i).$$

By Lemma 1, the two versions of each update have the same expectation, so it suffices to prove the proposition for the outcome-level updates. As fixed in Appendix B, the outcomes are i.i.d. from π_θ and $\pi_\theta(x) > 0$ for every x.

Proposition 1 (The witness advantage follows the negative gradient of MMD^2). *Let every rollout in a group of $G \geq 2$ rollouts receive the witness advantage. Then $\mathbb{E}[U] = -\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q)$, and $\mathrm{MMD}^2(\pi_\theta, q) = 0$ only at $\pi_\theta = q$. After the group mean is subtracted,*

$$\mathbb{E}[U_\mathrm{c}] = -\tfrac{G-1}{G}\, \nabla_\theta \mathrm{MMD}^2(\pi_\theta, q) + \tfrac{1}{G}\, \nabla_\theta \|\pi_\theta\|^2, \tag{4}$$

where $\|\pi_\theta\|^2 = \sum_x \pi_\theta(x)^2$ is the probability that two independent draws from π_θ give the same outcome.

Proof idea. Given the outcome x_i of rollout i, the other $G - 1$ rollouts are still independent draws from π_θ, so the leave-one-out frequency $\hat{p}_{-i}(x_i)$ has conditional mean $\pi_\theta(x_i)$. The advantage therefore has conditional mean $2\big(q(x_i) - \pi_\theta(x_i)\big)$, the negative of the witness function up to scale, and Lemma 2 turns this into $-\nabla_\theta \mathrm{MMD}^2$ (Steps 1 and 2). The group mean $\bar{A}$ does not have this property. It contains A_j for every $j \neq i$, and A_j depends on x_i because rollout i is one of the rollouts that $\hat{p}_{-j}$ counts. Step 3 computes this dependence exactly.

Step 1: the MMD in closed form. With the exact-match kernel $k(x, x') = \mathbb{1}[x = x']$, the kernel expansion of the squared MMD, with independent draws $x, x' \sim \pi_\theta$ and $z, z' \sim q$, is

$$\mathrm{MMD}^2(\pi_\theta, q) = \mathbb{E}[k(x, x')] + \mathbb{E}[k(z, z')] - 2\,\mathbb{E}[k(x, z)].$$

Each term is a collision probability, the probability that two independent draws give the same outcome:

$$\mathbb{E}[k(x,x')] = \sum_x \pi_\theta(x)^2, \qquad \mathbb{E}[k(z,z')] = \sum_x q(x)^2, \qquad \mathbb{E}[k(x,z)] = \sum_x \pi_\theta(x)\, q(x).$$

Substituting the three terms gives

$$\mathrm{MMD}^2(\pi_\theta, q) = \sum_x \pi_\theta(x)^2 + \sum_x q(x)^2 - 2\sum_x \pi_\theta(x)\, q(x) = \sum_x \big(\pi_\theta(x) - q(x)\big)^2. \tag{6}$$

A sum of squares vanishes only when every term vanishes, so $\mathrm{MMD}^2(\pi_\theta, q) = 0$ only at $\pi_\theta = q$, which is the second claim of the proposition. Differentiating Equation 6 gives the gradient that the update should match,

$$-\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q) = 2\sum_x \big(q(x) - \pi_\theta(x)\big)\, \nabla_\theta \pi_\theta(x). \tag{7}$$

Step 2: the update before centering is unbiased. The rollouts are exchangeable, so $\mathbb{E}[U] = \mathbb{E}[A_1\, s(x_1)]$. We split $A_1 = 2q(x_1) - 2\hat{p}_{-1}(x_1)$ and treat the two terms separately. The target term is Lemma 2(b) with $f = 2q$,

$$\mathbb{E}\big[2q(x_1)\, s(x_1)\big] = 2\sum_x q(x)\, \nabla_\theta \pi_\theta(x).$$

The leave-one-out term is an average of $G-1$ indicator terms, each given by Lemma 3(a),

$$\begin{aligned}\mathbb{E}\big[2\hat{p}_{-1}(x_1)\, s(x_1)\big] &= \frac{2}{G-1}\sum_{j=2}^{G} \mathbb{E}\big[\mathbb{1}[x_j = x_1]\, s(x_1)\big] \\ &= \frac{2}{G-1}\cdot (G-1)\cdot \tfrac{1}{2}\nabla_\theta \|\pi_\theta\|^2 = 2\sum_x \pi_\theta(x)\, \nabla_\theta \pi_\theta(x).\end{aligned}$$

Subtracting the second display from the first and comparing with Equation 7,

$$\mathbb{E}[U] = 2\sum_x \big(q(x) - \pi_\theta(x)\big)\, \nabla_\theta \pi_\theta(x) = -\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q).$$

This is the first claim of the proposition. It holds at every $G \geq 2$.

Step 3: the update after centering. Centering replaces every advantage A_i by $\tilde{A}_i = A_i - \bar{A}$. By linearity,

$$U_{\mathrm{c}} = U - \bar{A}\, \bar{g}, \qquad \text{where } \bar{g} = \frac{1}{G}\sum_{i=1}^{G} s(x_i). \tag{8}$$

It remains to compute $\mathbb{E}[\bar{A}\, \bar{g}]$, which is an average of G^2 products,

$$\mathbb{E}[\bar{A}\, \bar{g}] = \frac{1}{G^2}\sum_{i=1}^{G}\sum_{j=1}^{G} \mathbb{E}\big[A_j\, s(x_i)\big]. \tag{9}$$

The G^2 terms fall into two kinds.

- Diagonal terms ($i = j$). There are G of them, and each equals $\mathbb{E}[A_i\, s(x_i)] = -\nabla_\theta \mathrm{MMD}^2$ by Step 2.
- Off-diagonal terms ($i \neq j$). There are $G(G-1)$ of them. Split $A_j = 2q(x_j) - 2\hat{p}_{-j}(x_j)$ again. The target part vanishes by Lemma 3(c). The leave-one-out part is

  $$\hat{p}_{-j}(x_j) = \frac{1}{G-1}\sum_{\ell \neq j} \mathbb{1}[x_\ell = x_j].$$

  Every index $\ell \notin \{i, j\}$ gives a match that does not involve rollout i, so it contributes zero by Lemma 3(b). The single index $\ell = i$ contributes $\frac{1}{2}\nabla_\theta \|\pi_\theta\|^2$ by Lemma 3(a).

Hence, for $i \neq j$,

$$\mathbb{E}\big[A_j\, s(x_i)\big] = -\frac{2}{G-1} \cdot \frac{1}{2}\nabla_\theta \|\pi_\theta\|^2 = -\frac{1}{G-1}\nabla_\theta \|\pi_\theta\|^2. \tag{10}$$

Substituting both kinds of term into Equation 9,

$$\begin{aligned}\mathbb{E}[\bar{A}\,\bar{g}] &= \frac{1}{G^2}\Big[G \cdot \big(-\nabla_\theta \mathrm{MMD}^2\big) + G(G-1) \cdot \Big(-\frac{1}{G-1}\nabla_\theta \|\pi_\theta\|^2\Big)\Big] \\ &= -\frac{1}{G}\nabla_\theta \mathrm{MMD}^2 - \frac{1}{G}\nabla_\theta \|\pi_\theta\|^2.\end{aligned}$$

Finally, Equation 8 and Step 2 give

$$\begin{aligned}\mathbb{E}[U_{\mathrm{c}}] = \mathbb{E}[U] - \mathbb{E}[\bar{A}\,\bar{g}] &= -\nabla_\theta \mathrm{MMD}^2 + \frac{1}{G}\nabla_\theta \mathrm{MMD}^2 + \frac{1}{G}\nabla_\theta \|\pi_\theta\|^2 \\ &= -\frac{G-1}{G}\nabla_\theta \mathrm{MMD}^2 + \frac{1}{G}\nabla_\theta \|\pi_\theta\|^2,\end{aligned}$$

which is Equation 4. This completes the proof. □

Why the group mean biases the update. The standard argument that subtracting a baseline leaves the expected update unchanged requires the baseline to be independent of the rollout's own outcome. A constant baseline satisfies this condition. The group mean $\bar{A}$ does not, and Equation 10 gives the exact size of the dependence.

C.1 Counting the Rollout Itself

Section 3.1 builds the witness advantage from the leave-one-out frequency and not from the group frequency $\hat{p}(x_i)$, which counts rollout i itself. We now show that the self-count adds a term that concentrates the model, and that after centering this term has about twice the weight of the term that centering adds to the leave-one-out update. The full-group advantage and its update are

$$\hat{A}_i = 2\big(q(x_i) - \hat{p}(x_i)\big), \qquad \hat{U} = \frac{1}{G}\sum_{i=1}^{G} \hat{A}_i\, s(x_i).$$

As in Step 2, $\mathbb{E}[\hat{U}] = \mathbb{E}[\hat{A}_1\, s(x_1)]$, and the target term is unchanged. The frequency term differs because rollout 1 always matches itself:

$$\hat{p}(x_1) = \frac{1}{G}\Big(1 + \sum_{j \geq 2} \mathbb{1}[x_j = x_1]\Big).$$

The constant 1 contributes zero by Lemma 2(a), and each of the $G-1$ indicators contributes $\frac{1}{2}\nabla_\theta \|\pi_\theta\|^2$ by Lemma 3(a):

$$\mathbb{E}\big[2\hat{p}(x_1)\, s(x_1)\big] = \frac{2}{G}\,\mathbb{E}[s(x_1)] + \frac{2(G-1)}{G} \cdot \tfrac{1}{2}\nabla_\theta \|\pi_\theta\|^2 = \frac{G-1}{G}\nabla_\theta \|\pi_\theta\|^2.$$

Using $\nabla_\theta \|\pi_\theta\|^2 = 2\sum_x \pi_\theta(x)\nabla_\theta \pi_\theta(x)$ and Equation 7,

$$\mathbb{E}[\hat{U}] = 2\sum_x q(x)\nabla_\theta \pi_\theta(x) - \frac{G-1}{G}\nabla_\theta \|\pi_\theta\|^2 = -\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q) + \frac{1}{G}\nabla_\theta \|\pi_\theta\|^2.$$

Before centering, the self-count adds a term of weight $1/G$ that increases the collision probability and so pushes the model to concentrate its probability on fewer outcomes.

Centering adds a term of the same form to the leave-one-out update (Equation 4), so the comparison that matters is between the two centered updates. Let $\hat{U}_{\mathrm{c}} = \frac{1}{G}\sum_i (\hat{A}_i - \bar{\hat{A}})\, s(x_i)$, where $\bar{\hat{A}}$ is the group mean of the $\hat{A}_i$. We repeat Step 3 with $\hat{A}_j$ in place of A_j. The diagonal terms equal $\mathbb{E}[\hat{U}]$. For an off-diagonal term ($i \neq j$), the frequency $\hat{p}(x_j)$ contains the constant $\frac{1}{G}$ from the self-match of rollout j, which contributes zero by Lemma 2(a), and a single match with x_i, which contributes $\frac{1}{G} \cdot \frac{1}{2}\nabla_\theta \|\pi_\theta\|^2$ by Lemma 3(a). Hence $\mathbb{E}[\hat{A}_j\, s(x_i)] = -\frac{1}{G}\nabla_\theta \|\pi_\theta\|^2$, and

$$\mathbb{E}[\hat{U}_{\mathrm{c}}] = -\frac{G-1}{G}\nabla_\theta \mathrm{MMD}^2(\pi_\theta, q) + \frac{2(G-1)}{G^2}\nabla_\theta \|\pi_\theta\|^2.$$

After centering, the self-count raises the weight of the concentration term from $1/G$ to $2(G-1)/G^2$, nearly twice as much. For a distribution π, completing the square gives $\frac{G-1}{G}(\pi(x) - q(x))^2 - \frac{2(G-1)}{G^2}\pi(x)^2 = \frac{(G-1)(G-2)}{G^2}\big(\pi(x) - \frac{G}{G-2}q(x)\big)^2 + \text{const}$. The argument of Appendix C.2, with $G/(G-2)$ in place of γ, then bounds the shift of the minimizer by $2/(G-2)$ in TV instead of $1/(G-2)$.

Removing the self-count is the U-statistic correction of the MMD estimator (Gretton et al., 2012; Bińkowski et al., 2018). It is also how the leave-one-out baseline for REINFORCE (Williams, 1992) is built, from the other rollouts only (Kool et al., 2019; Ahmadian et al., 2024).

C.2 Where the Centered Update Moves the Model

Centering adds a term of weight $1/G$ to the expected update (Proposition 1). We now ask how far this term moves the distribution that training approaches. For a distribution π on $\mathcal{X}$, define the loss

$$F_{\mathrm{c}}(\pi) = \frac{G-1}{G}\,\mathrm{MMD}^2(\pi, q) - \frac{1}{G}\,\|\pi\|^2.$$

By Equation 4, the expected centered update is $\mathbb{E}[U_{\mathrm{c}}] = -\nabla_\theta F_{\mathrm{c}}(\pi_\theta)$, so in expectation training descends F_{c}. This loss is uniformly close to MMD^2. Since $\mathrm{MMD}^2(\pi, q) = \|\pi - q\|^2 \le \|\pi - q\|_1 \le 2$ and $\|\pi\|^2 \le 1$,

$$\big|F_{\mathrm{c}}(\pi) - \mathrm{MMD}^2(\pi, q)\big| = \frac{1}{G}\big(\mathrm{MMD}^2(\pi, q) + \|\pi\|^2\big) \le \frac{3}{G}.$$

A small difference between two losses does not by itself bound the distance between their minimizers, so we compute the minimizer of F_{c} directly.

The minimizer. Suppose that the model can assign any probability to each outcome, and let $G \ge 3$. At $G = 2$, the centered advantages are $\tilde{A}_1 = q(x_1) - q(x_2) = -\tilde{A}_2$, so the frequencies cancel. The loss $F_{\mathrm{c}}(\pi) = \frac{1}{2}\|q\|^2 - \sum_x \pi(x)\,q(x)$ is then linear in π, and its minimizer is a point mass on a most likely outcome of q. For $G \ge 3$, we find the distribution $\pi^\star$ that minimizes F_{c} over all distributions on $\mathcal{X}$. Let $\gamma = (G-1)/(G-2)$, which is slightly larger than 1. Completing the square for each outcome x gives

$$\frac{G-1}{G}\big(\pi(x) - q(x)\big)^2 - \frac{1}{G}\,\pi(x)^2 = \frac{G-2}{G}\big(\pi(x) - \gamma\,q(x)\big)^2 + (\text{a term that does not depend on } \pi).$$

Summed over x, F_{c} is a positive multiple of the squared Euclidean distance from π to the vector γq, plus a constant. Its minimizer $\pi^\star$ is therefore unique, and it is the Euclidean projection of γq onto the set of distributions on $\mathcal{X}$. We write this projection as minimizing $\|\pi - \gamma q\|^2$ subject to $\sum_x \pi(x) = 1$ and $\pi(x) \ge 0$. The Karush–Kuhn–Tucker conditions, with multiplier 2τ on the equality constraint, give

$$\pi^\star(x) = \max\big(\gamma\,q(x) - \tau,\ 0\big) \quad \text{for all } x. \tag{11}$$

Here τ is the number for which $\sum_x \pi^\star(x) = 1$. This number is unique. The sum $\sum_x \max(\gamma q(x) - \tau, 0)$ is continuous and nonincreasing in τ, and strictly decreasing wherever it is positive. It equals $\gamma > 1$ at $\tau = 0$ and 0 at $\tau = \gamma \max_x q(x)$, so it takes the value 1 at exactly one τ.

Equation 11 implies three properties of $\pi^\star$. It gives zero probability to every outcome that q excludes, it has a closed form when no target probability is too small, and it is within $1/(G-2)$ of q in TV. We prove them in turn.

First, $\tau > 0$. If $\tau \le 0$, Equation 11 would give $\sum_x \pi^\star(x) \ge \gamma \sum_x q(x) = \gamma > 1$. Every outcome with $q(x) = 0$, including $\perp$, therefore has $\pi^\star(x) = 0$.

Second, $\pi^\star$ has a closed form when no target probability is too small. Let $S = \{x : q(x) > 0\}$ be the support of q, let u be the uniform distribution on S, and suppose that $(G-1)\,q(x) \ge 1/|S|$ for every $x \in S$. Then

$$\tau = \frac{1}{(G-2)\,|S|}.$$

To check this value, note that it satisfies $\gamma\,q(x) - \tau \ge 0$ on S and $\sum_{x\in S}(\gamma\,q(x) - \tau) = \gamma - 1/(G-2) = 1$. It is therefore the unique τ of Equation 11. Substituting it into Equation 11 gives, on S,

$$\pi^\star = \frac{(G-1)\,q - u}{G-2}, \qquad \pi^\star - q = \frac{q-u}{G-2}.$$

Each outcome moves away from uniform by $1/(G-2)$ times the difference between its target probability and $1/|S|$, which is less than 2% of that difference at $G = 64$. If some $x \in S$ has $(G-1)\,q(x) < 1/|S|$, this formula would give $\pi^\star(x) < 0$. Equation 11 instead sets to zero every outcome with $\gamma\, q(x) \le \tau$, which are the outcomes with the smallest target probabilities.

Third, $\pi^\star$ is always close to q. Let $S^\star = \{x : \pi^\star(x) > 0\}$ be its support, write $q(B) = \sum_{x \in B} q(x)$ for the target probability of a set B of outcomes, and write $z^+ = \max(z, 0)$. The outcomes that $\pi^\star$ sets to zero carry little target probability. Summing Equation 11 over $S^\star$ gives $\gamma\, q(S^\star) - |S^\star|\,\tau = 1$. Since $\tau > 0$ by the first property, $q(S^\star) \ge 1/\gamma$, so

$$1 - q(S^\star) \le 1 - \frac{1}{\gamma} = \frac{1}{G-1}.$$

The outcomes that $\pi^\star$ keeps gain little probability. Only outcomes in $S^\star$ can have $\pi^\star(x) > q(x)$, and there $\pi^\star(x) - q(x) = q(x)/(G-2) - \tau < q(x)/(G-2)$. Hence

$$\mathrm{TV}(\pi^\star, q) = \sum_x \big(\pi^\star(x) - q(x)\big)^+ \le \frac{q(S^\star)}{G-2} \le \frac{1}{G-2},$$

which is 0.016 at $G = 64$.

What the minimizer means for training. The distribution $\pi^\star$ is the unique minimizer of the loss that the centered update descends in expectation, and it lies within $1/(G-2)$ of the target in TV. Like q itself, it gives zero probability to $\perp$, so a softmax model approaches it without reaching it. We do not prove that the idealized flow of the centered update converges to $\pi^\star$. In a separate small simulation, which is not among the checks of the code release (Appendix C.4), we trained a softmax model on the expected centered update for 45 targets (five group sizes, three alphabet sizes, and random, Zipf, and small-mass targets). It came within 3×10^{-3} of $\pi^\star$ in every coordinate, and the remaining distance kept shrinking with more steps.

C.3 A Worked Example

Table 2 evaluates the witness advantage on a group small enough to check by hand. The group has $G = 6$ rollouts with outcomes $(a, a, a, b, c, \perp)$, and the target is $q = (1/2,\ 3/10,\ 1/5,\ 0)$ on $(a, b, c, \perp)$. For a rollout on a, the other five rollouts contain two more copies of a, so $\hat{p}_{-i}(a) = 2/5$ and $A_i = 2(1/2 - 2/5) = 1/5$. For the rollout on b, none of the other five is b, so $A_i = 2(3/10 - 0) = 3/5$. Likewise $A_i = 2/5$ for c. For the lone $\perp$, both $q(\perp)$ and $\hat{p}_{-i}(\perp)$ are zero, so $A_i = 0$.

The full-group variant uses $\hat{p}(a) = 3/6$ and $\hat{p}(b) = \hat{p}(c) = \hat{p}(\perp) = 1/6$. It gives 0 to a, $4/15$ to b, $1/15$ to c, and $-1/3$ to $\perp$. The self-count changes every value. For example, the three copies of a match the target probability exactly, so the full-group variant gives them zero advantage, although among the other five rollouts a is under-produced.

Table 2: The witness advantage and its full-group variant on a group of six rollouts with outcomes $(a, a, a, b, c, \perp)$, in exact fractions.

	a (3 copies)	b	c	$\perp$
target probability $q(x)$	$1/2$	$3/10$	$1/5$	0
witness advantage A_i (ours)	$1/5$	$3/5$	$2/5$	0
full-group variant $\hat{A}_i$	0	$4/15$	$1/15$	$-1/3$

Invalid outcomes. The example shows that a lone invalid rollout receives advantage zero. In general, a group with $m \ge 2$ invalid rollouts gives each of them $A_i = -2(m-1)/(G-1)$. The model is still pushed away from invalid outputs in expectation, because the conditional mean of the advantage at $\perp$ is $2\big(q(\perp) - \pi_\theta(\perp)\big) = -2\pi_\theta(\perp) < 0$.

C.4 Numerical Checks

The code release checks Proposition 1 numerically. It recomputes Table 2 in exact fractions and checks the expected update against the automatic gradient of $-\mathrm{MMD}^2$ (largest deviation 1.5×10^{-8}).

It also enumerates every group for $|\mathcal{X}| = 3$ and small G, which confirms Equation 4 to 3.4×10^{-15}, including an alphabet with a zero-mass outcome. The same enumeration confirms the centered full-group update of Appendix C.1. A Monte Carlo run with $40{,}000$ groups at $G = 8$ shows the bias of the full-group update. Its largest coordinate-wise deviation from $-\nabla_\theta \mathrm{MMD}^2$ is 0.016, against 0.001 for the leave-one-out update, a deviation of the size of the Monte Carlo error.

D Proofs for the Group-Scalar Reward

Section 3.3 makes two claims about the group-scalar reward, which scores a group by the negative TV between its empirical distribution and the target and gives this score to every rollout. First, with one group per prompt it gives GRPO no training signal. Second, the repaired version, which scores several subgroups and subtracts their mean score, optimizes a biased objective. This appendix states both claims formally and proves them. It ends with a simulation of when the bias starts to matter.

D.1 Setting and Statements

The notation of Appendix B remains in force. In addition:

- A batch for one prompt consists of $K \geq 2$ subgroups, each of G rollouts drawn i.i.d. from π_θ. We trained with $K = 4$ and $G = 64$.
- Subgroup ℓ has empirical distribution $\hat{p}^{(\ell)}$ and TV $T_\ell = \mathrm{TV}(\hat{p}^{(\ell)}, q)$. The mean TV of the batch is $\bar{T} = \frac{1}{K} \sum_{\ell=1}^{K} T_\ell$.
- Every rollout in subgroup ℓ receives the reward $-T_\ell$. Subtracting the mean reward over the batch gives rollout i the centered advantage $\tilde{A}_i = \bar{T} - T_{\ell(i)}$, where $\ell(i)$ is its subgroup. The outcome-level update is $U = \frac{1}{KG} \sum_i \tilde{A}_i \, s(x_i)$. This reward is used only with centering, so we write its centered update as U without the subscript c.
- $\hat{p}_G$ is the empirical distribution of G i.i.d. draws. It is the group frequency $\hat{p}$ of Appendix B, with the group size made explicit. For a distribution π on $\mathcal{X}$, the surrogate loss $F_G(\pi) = \mathbb{E}_\pi[\mathrm{TV}(\hat{p}_G, q)]$ is the expected TV of G draws from π. Its bias $B_G(\pi) = F_G(\pi) - \mathrm{TV}(\pi, q)$ is the amount by which the expected TV of G draws exceeds the TV of π itself.

The first result explains why the group-scalar reward needs subgroups.

Lemma 4 (Equal rewards give zero advantages)**.** *Let every rollout in a GRPO group receive the same reward. Then every advantage in that group is zero after the group mean is subtracted.*

Proof. If every rollout receives the reward r_0, then every centered advantage is $r_0 - r_0 = 0$. The update is therefore exactly zero on every batch. The original GRPO also divides by the standard deviation of the rewards. This standard deviation is zero, and implementations add a small constant to the denominator, so the quotient is still zero. □

In a ten-step pilot run of this reward, the logged gradient norm was zero at every step.

The second result says what the repaired reward optimizes and why this objective is biased.

Proposition 3 (The group-scalar reward optimizes a biased objective)**.** *Let a batch contain $K \geq 2$ subgroups of G rollouts, and give every rollout in subgroup ℓ the centered advantage $\bar{T} - T_\ell$. Then the following hold (proof in Appendix D.2).*

(i) The advantages take at most K distinct values per batch, and each advantage has standard deviation at most $\sqrt{(K-1)/K}\,/\,(2\sqrt{G})$.

(ii) The expected update is $\mathbb{E}[U] = -\frac{K-1}{KG}\nabla_\theta F_G(\pi_\theta)$.

(iii) Let $\mathcal{X}_1 = \{x : 1/G \leq q(x) \leq 1 - 1/G\}$ be the outcomes whose target probability G draws can resolve. At $\pi_\theta = q$,

$$\frac{1}{2\sqrt{2G}} \sum_{x \in \mathcal{X}_1} \sqrt{q(x)\big(1-q(x)\big)} \;\leq\; B_G(q) \;\leq\; \frac{1}{2\sqrt{G}} \sum_{x \in \mathcal{X}} \sqrt{q(x)\big(1-q(x)\big)}.$$

(iv) Every minimizer of F_G over all distributions on $\mathcal{X}$ is within TV $B_G(q)$ of q. For some targets, F_G is smaller at a distribution that gives zero probability to every outcome with $q(x) < 1/G$ than at q itself.

Part (i) limits how much the training signal can vary within a batch. At $K = 4$ and $G = 64$ the bound is 0.054, and Appendix G.4 measures 0.031. Part (ii) says that, in expectation, the repaired reward performs gradient descent on the surrogate loss F_G, the expected TV of G draws, and not on the model's own TV. The factor $(K-1)/(KG) \approx 0.012$ matters little, because AdamW is nearly invariant to a constant rescaling of the gradient. Parts (iii) and (iv) concern the surrogate near the target. Even a model that samples exactly from q has $F_G(q) = B_G(q) > 0$, so the surrogate penalizes the target itself by the sampling error of G draws. Suppose every outcome in the support of q has $q(x) \geq 1/G$. If the support has at least two outcomes, each of them then also has $q(x) \leq 1 - 1/G$, so the support lies in $\mathcal{X}_1$ and the two sides of part (iii) differ by a factor of $\sqrt{2}$. The bias is then of order $\frac{1}{\sqrt{G}} \sum_x \sqrt{q(x)(1-q(x))}$, the rate quoted in Section 3.3. When many outcomes have target probability below $1/G$, the surrogate can prefer to give them zero probability.

D.2 Proofs

The bound of part (i) rests on the following lemma.

Lemma 5 (The subgroup TV varies little)**.** *Let T_ℓ be the TV of a subgroup of G i.i.d. draws. Then changing one draw changes T_ℓ by at most $1/G$, and $\mathrm{Var}(T_\ell) \leq 1/(4G)$.*

Proof. Replacing one draw moves $\hat{p}^{(\ell)}$ by $-1/G$ in one coordinate and by $+1/G$ in another. Since $T_\ell = \frac{1}{2}\|\hat{p}^{(\ell)} - q\|_1$, it moves by at most $\frac{1}{2} \cdot \frac{2}{G} = \frac{1}{G}$. The Efron–Stein inequality bounds the variance of a function of independent draws by the sum over draws of the expected variance over that draw with the others fixed. With the others fixed, T_ℓ lies in an interval of length at most $1/G$, so its variance over one draw is at most $1/(4G^2)$. Summing over the G draws gives $\mathrm{Var}(T_\ell) \leq 1/(4G)$. □

Proof of part (i). The advantage $\tilde{A}_i = \bar{T} - T_{\ell(i)}$ depends on i only through its subgroup, so it takes at most K distinct values in a batch. For the bound, write

$$\bar{T} - T_\ell = \frac{1}{K} \sum_{\ell' \neq \ell} T_{\ell'} - \frac{K-1}{K}\, T_\ell.$$

The TVs of different subgroups are i.i.d., so the two parts are independent and

$$\mathrm{Var}(\bar{T} - T_\ell) = \frac{K-1}{K^2}\mathrm{Var}(T_\ell) + \frac{(K-1)^2}{K^2}\mathrm{Var}(T_\ell) = \frac{K-1}{K}\,\mathrm{Var}(T_\ell) \leq \frac{K-1}{4KG}$$

by Lemma 5. Taking the square root gives the bound.

The same number also bounds the expected spread of the advantages within one batch. The advantages have mean zero, so the expected within-batch variance is

$$\mathbb{E}\Big[\frac{1}{K}\sum_{\ell=1}^{K}(\bar{T} - T_\ell)^2\Big] = \mathrm{Var}(\bar{T} - T_\ell).$$

By Jensen's inequality, the expected within-batch standard deviation is at most the square root of this value. □

Proof of part (ii). We write the update as a sum over subgroups and show that only the correlation between a subgroup's TV and its own summed score survives. Let $S_\ell = \sum_{i:\,\ell(i)=\ell} s(x_i)$ be the summed score of subgroup ℓ. Three facts are needed.

1. $\mathbb{E}[S_\ell] = 0$, by Lemma 2(a) applied to each draw.
2. $\mathbb{E}[T_{\ell'}\, S_\ell] = \mathbb{E}[T_{\ell'}]\,\mathbb{E}[S_\ell] = 0$ for $\ell' \neq \ell$, because different subgroups are independent.
3. $\mathbb{E}[T_\ell\, S_\ell] = \nabla_\theta\, \mathbb{E}[T_\ell] = \nabla_\theta F_G(\pi_\theta)$. This is the score-function identity for a function of G i.i.d. draws, obtained by differentiating the finite sum $\mathbb{E}[T_\ell] = \sum_{x_1,\ldots,x_G} T_\ell(x_1,\ldots,x_G)\prod_i \pi_\theta(x_i)$.

Grouping the update by subgroup and substituting $\tilde{A}_i = \bar{T} - T_{\ell(i)}$,

$$\mathbb{E}[U] = \frac{1}{KG}\sum_{\ell=1}^{K}\mathbb{E}\big[(\bar{T}-T_\ell)\,S_\ell\big] = \frac{1}{KG}\sum_{\ell=1}^{K}\Big(\frac{1}{K}\sum_{\ell'=1}^{K}\mathbb{E}[T_{\ell'}\,S_\ell] - \mathbb{E}[T_\ell\,S_\ell]\Big)$$
$$= -\frac{1}{KG}\cdot\frac{K-1}{K}\sum_{\ell=1}^{K}\mathbb{E}[T_\ell\,S_\ell],$$

where the last step uses fact 2 to keep only the term $\ell' = \ell$ of the inner sum. By fact 3, each of the K terms of the remaining sum equals $\nabla_\theta F_G(\pi_\theta)$, so the sum is $K\,\nabla_\theta F_G(\pi_\theta)$. Substituting,

$$\mathbb{E}[U] = -\frac{K-1}{KG}\,\nabla_\theta F_G(\pi_\theta).$$

□

Proof of part (iii). Both bounds rest on the fact that $G\hat{p}_G(x)$ is a binomial count with G trials and success probability $\pi_\theta(x)$.

For the upper bound, fix any π_θ. For each outcome x, the triangle inequality and then the Cauchy–Schwarz inequality give

$$\mathbb{E}\big[|\hat{p}_G(x)-q(x)|\big] \le \big|\pi_\theta(x)-q(x)\big| + \mathbb{E}\big[|\hat{p}_G(x)-\pi_\theta(x)|\big] \le \big|\pi_\theta(x)-q(x)\big| + \sqrt{\frac{\pi_\theta(x)\big(1-\pi_\theta(x)\big)}{G}},$$

where the second step uses that the binomial count has variance $G\pi_\theta(x)(1-\pi_\theta(x))$. Summing over x and halving gives

$$B_G(\pi_\theta) \le \frac{1}{2\sqrt{G}}\sum_x \sqrt{\pi_\theta(x)\big(1-\pi_\theta(x)\big)},$$

which at $\pi_\theta = q$ is the upper bound.

For the lower bound, take $\pi_\theta = q$. Since $\mathrm{TV}(q,q) = 0$,

$$B_G(q) = F_G(q) = \frac{1}{2G}\sum_x \mathbb{E}\big[|G\hat{p}_G(x) - Gq(x)|\big],$$

where $G\hat{p}_G(x)$ is binomial with G trials and success probability $q(x)$. For $G \ge 2$, a binomial with G trials and success probability $v \in [1/G,\, 1-1/G]$ has mean absolute deviation at least $\sqrt{Gv(1-v)/2}$ (Berend & Kontorovich, 2013, Theorem 1). Applying this bound with $v = q(x)$ to each outcome in $\mathcal{X}_1$ and dropping the other outcomes, whose terms are nonnegative, gives the lower bound. □

Proof of part (iv). Let $\pi^\dagger$ minimize F_G over all distributions on $\mathcal{X}$. Two inequalities bound its distance to q. First, $\mathrm{TV}(\cdot, q)$ is convex and $\mathbb{E}[\hat{p}_G] = \pi^\dagger$ when the draws come from $\pi^\dagger$, so Jensen's inequality gives $F_G(\pi^\dagger) \ge \mathrm{TV}(\pi^\dagger, q)$. Second, $\pi^\dagger$ is a minimizer, so $F_G(\pi^\dagger) \le F_G(q) = B_G(q)$. Chaining the two,

$$\mathrm{TV}(\pi^\dagger, q) \le F_G(\pi^\dagger) \le B_G(q).$$

For the second claim, we give a target for which deleting the rare outcomes beats sampling from the target. Take $G = 64$ and a target with one head outcome of probability $1-\beta$ and M tail outcomes of probability η each,

$$q = \big(1-\beta,\ \eta,\ \ldots,\ \eta\big), \qquad \beta = 0.1, \quad M = 64, \quad \eta = \beta/M,$$

so every tail outcome has probability below $1/G$. We compare two distributions.

1. The point mass δ_0 on the head outcome gives zero probability to every tail outcome. Its empirical distribution is always δ_0 itself, so $F_G(\delta_0) = \mathrm{TV}(\delta_0, q) = \beta = 0.1$ exactly.
2. The target q has $F_G(q) = B_G(q)$, and we bound $B_G(q)$ from below one outcome at a time. Each tail outcome x contributes to $\mathbb{E}\big[\|\hat{p}_G - q\|_1\big]$ the amount

$$\mathbb{E}\big[|\hat{p}_G(x) - \eta|\big] = 2\,\mathbb{E}\big[(\eta - \hat{p}_G(x))^+\big] \geq 2\eta(1-\eta)^G.$$

The equality holds because $\hat{p}_G(x) - \eta$ has mean zero. The inequality keeps the event that x is never drawn, which has probability $(1-\eta)^G$. It is in fact an equality here, because $G\eta < 1$, so $\hat{p}_G(x)$ falls below η only when x is never drawn. The head outcome contributes at least $\sqrt{\beta(1-\beta)/(2G)}$ by the binomial bound of part (iii), since $1-\beta \in [1/G, 1-1/G]$. Summing over the M tail outcomes with $M\eta = \beta$, adding the head, and halving,

$$B_G(q) \;\geq\; \beta(1-\eta)^G + \frac{1}{2\sqrt{2}}\sqrt{\frac{\beta(1-\beta)}{G}} \;=\; 0.0905 + 0.0133 \;>\; 0.1.$$

Hence $F_G(q) > 0.1 = F_G(\delta_0)$, and the surrogate strictly prefers the distribution without the tail to the target itself. An exact binomial computation gives $B_G(q) = 0.1056$, so the margin is 0.0056. The slack in the bound above comes from the head term, for which the binomial bound gives 0.0133 against an exact 0.0151. Outcomes rarer than $1/G$ cost the surrogate more in sampling fluctuation than their deletion costs in TV. □

D.3 When the Bias Matters

Parts (ii) and (iii) predict that the expected update of the group-scalar reward shrinks once the model's TV to the target falls to the size of the bias. By part (ii) and the definition of B_G,

$$\mathbb{E}[U] = -\frac{K-1}{KG}\Big(\nabla_\theta \mathrm{TV}(\pi_\theta, q) + \nabla_\theta B_G(\pi_\theta)\Big).$$

Where the gradient of the bias is negligible, the expected update is a scaled negative gradient of the model's own TV. We do not bound $\nabla_\theta B_G$. We instead locate by simulation where this approximation breaks.

Setup. We take a uniform target q on d outcomes and a model parameterized by a softmax over outcomes, with logits θ. The model moves along the path

$$\pi = (1-t)\,\pi_0 + t\,q, \qquad t \in [0,1],$$

where π_0 is uniform on half of the outcomes and has TV $1/2$ to q. We use $K = 4$, $G \in \{8, 64, 256\}$, and $d \in \{2, 10, 100\}$, nine settings in all. At 15 points of the path, with TV to q spaced geometrically between 0.45 and 0.001, we estimate the expected update by Monte Carlo and subtract the Monte Carlo variance from its squared norm. We then divide its norm by the norm of $\frac{K-1}{KG}\nabla_\theta \mathrm{TV}(\pi_\theta, q)$. This ratio is close to one where the approximation holds. For a uniform target, the bias scale of part (iii) is $\frac{1}{\sqrt{G}}\sum_x \sqrt{q(x)(1-q(x))} = \sqrt{(d-1)/G}$. Scanning from the far end of the path, we read off the TV at which the ratio first falls below $1/2$, by linear interpolation in $\log \mathrm{TV}$ between the two neighboring points. We write this crossing as $\kappa\sqrt{(d-1)/G}$.

Result. The crossing is proportional to the bias scale. In the five settings where the ratio is at least 0.75 at the two largest simulated distances, κ lies between 0.334 and 0.342, and a log-log fit of the crossings against $\sqrt{(d-1)/G}$ has slope 1.01. In two more settings the ratio is below 0.75 at one of the two largest simulated distances, so the bias already matters at the start of the path. They give $\kappa = 0.340$ and 0.370 if the crossing is read off anyway. In the remaining two settings, $(G, d) = (8, 100)$ and $(64, 100)$, the ratio is below $1/2$ along the whole path. Our wide-support families at $G = 64$ are in this regime. Wherever the norm of the Monte Carlo estimate exceeds five times its standard error, the direction of the expected update has cosine similarity at least 0.98 with $-\nabla_\theta \mathrm{TV}(\pi_\theta, q)$, so only its size shrinks. Extending the scaling from uniform to general targets is an extrapolation based on part (iii), not a separate measurement.

D.4 Numerical Checks

The code release checks Lemma 4, the count of distinct advantages, and the bound of part (i) on sampled batches, and checks part (ii) against exact enumeration at $|\mathcal{X}| = 2$. It also computes the bias of part (iii), the example of part (iv), and the simulation above. The released simulation script reports the crossings against $\sqrt{d/G}$. The values of κ and the slope quoted above are recomputed from its stored output against $\sqrt{(d-1)/G}$, the scale of part (iii), by the release script that prints the paper's numbers.

E Proofs for the Sign Witness

Proposition 2 states that the sign witness trains the model toward the target except on low-mass outcomes. We restate it and prove the three parts in order. We then simulate what the proof leaves open, and relate the sign witness to the property-ratio reward of Huang et al. (2026).

The notation of Appendix B remains in force. Rollout i receives the sign witness $A_i = \operatorname{sign}\big(q(x_i) - \hat{p}_{-i}(x_i)\big)$, with $\operatorname{sign}(0) = 0$, so a lone invalid rollout again receives zero. The population sign of outcome x is $\operatorname{sign}\big(q(x) - \pi_\theta(x)\big)$, and an outcome is low-mass if $0 < q(x) < 1/(G-1)$. We write $\bar{a}(x) = \mathbb{E}[A_i \mid x_i = x]$ for the expected advantage of a rollout with outcome x, which does not depend on i. In the proposition, the expected update is $\mathbb{E}\big[\frac{1}{G}\sum_i A_i\, s(x_i)\big]$ with the advantages before centering. The paragraph on centering in Appendix E.3 shows that the conclusions of part (iii) also hold for the update that GRPO applies, $\mathbb{E}\big[\frac{1}{G}\sum_i \tilde{A}_i\, s(x_i)\big]$, when $G \geq 3$.

Proposition 2 (The sign witness follows the negative gradient of total variation)**.** *Let every rollout in a group of $G \geq 2$ rollouts receive the sign witness. Then the following hold.*

(i) The population sign is the negative of the witness of total variation. If $q(x) \neq \pi_\theta(x)$ for every outcome x, the expected update with the population sign is $-\nabla_\theta\, 2\mathrm{TV}(\pi_\theta, q)$.

(ii) Given that a rollout has an outcome x with $\delta = q(x) - \pi_\theta(x) \neq 0$, its sign witness and the population sign differ with probability at most $\exp(-2(G-1)\delta^2)$.

(iii) A rollout with a low-mass outcome x receives expected advantage $2(1-\pi_\theta(x))^{G-1} - 1$, independent of $q(x)$. If the model can assign any probability to each outcome, every stationary point of the expected update therefore gives the same probability to all low-mass outcomes that it does not set to zero. In particular, $\pi_\theta = q$ is not a stationary point when two low-mass outcomes have different target probabilities.

Proof idea. Part (i) combines the dual form of total variation with the score identity. Part (ii) holds because, given the outcome of rollout i, the leave-one-out count is binomial and concentrates around its mean. Part (iii) uses the fact that the leave-one-out frequency takes values on a grid with spacing $1/(G-1)$. For a low-mass outcome, it is either 0 or above the target, so the sign only records whether the outcome appeared among the other rollouts.

E.1 Part (i): The Population Sign Is the Witness of Total Variation

For distributions π and q and any function f with $|f(x)| \leq 1$ for all x,

$$\sum_x f(x)\big(\pi(x) - q(x)\big) \leq \sum_x \big|\pi(x) - q(x)\big| = 2\,\mathrm{TV}(\pi, q),$$

with equality if and only if $f(x) = \operatorname{sign}(\pi(x) - q(x))$ wherever $\pi(x) \neq q(x)$. This is the dual representation of total variation (Müller, 1997; Sriperumbudur et al., 2012). Its maximizer, the witness of total variation, is $\operatorname{sign}(\pi - q)$, the negative of the population sign. This is the first claim of part (i).

For the expected update, assume $\pi_\theta(x) \neq q(x)$ for every x. Then each term $|\pi_\theta(x) - q(x)|$ is differentiable, with gradient $\operatorname{sign}(\pi_\theta(x) - q(x))\,\nabla_\theta \pi_\theta(x)$. The population sign of rollout i is a function of x_i alone, so Lemma 1 with $N = G$ applies, and each of the G terms of the update has the same expectation. Lemma 2(b) with $f = \operatorname{sign}(q - \pi_\theta)$ then gives

$$\mathbb{E}_{x\sim\pi_\theta}\Big[\operatorname{sign}\big(q(x) - \pi_\theta(x)\big)\, s(x)\Big] = -\sum_x \operatorname{sign}\big(\pi_\theta(x) - q(x)\big)\,\nabla_\theta \pi_\theta(x) = -2\,\nabla_\theta \mathrm{TV}(\pi_\theta, q).$$

This is the second claim of part (i). □

E.2 Part (ii): The Sign Computed From the Group Rarely Differs

Condition on $x_i = x$. The leave-one-out count excludes rollout i, and the other $G-1$ draws are independent of x_i, so $(G-1)\,\hat{p}_{-i}(x)$ is exactly binomial with $G-1$ trials and success probability $\pi_\theta(x)$. Suppose $\delta = q(x) - \pi_\theta(x) > 0$, so the population sign is $+1$. The sign computed from the group is not $+1$ only if $\hat{p}_{-i}(x) \geq q(x)$, that is, only if $\hat{p}_{-i}(x) - \pi_\theta(x) \geq \delta$. This event includes the tie $\hat{p}_{-i}(x) = q(x)$, where the sign is 0. Hoeffding's inequality for the mean of $G-1$ independent Bernoulli draws bounds its probability by $\exp(-2(G-1)\delta^2)$. The case $\delta < 0$ is symmetric. □

E.3 Part (iii): Low-Mass Outcomes

The expected advantage of a low-mass outcome. Fix a low-mass outcome x, so $0 < q(x) < 1/(G-1)$. The leave-one-out frequency $\hat{p}_{-i}(x)$ takes values in $\{0, \frac{1}{G-1}, \frac{2}{G-1}, \dots\}$, so it is never strictly between 0 and $q(x)$ and never equal to $q(x)$. The sign is therefore $+1$ exactly when none of the other $G-1$ rollouts produced x, and -1 otherwise. Given $x_i = x$, the expected advantage is

$$\bar{a}(x) = (+1)\cdot\big(1-\pi_\theta(x)\big)^{G-1} + (-1)\cdot\Big(1-\big(1-\pi_\theta(x)\big)^{G-1}\Big) = 2\big(1-\pi_\theta(x)\big)^{G-1} - 1. \quad (12)$$

The target probability $q(x)$ does not appear. The expected advantage is positive exactly when $\pi_\theta(x) < 1 - 2^{-1/(G-1)}$, which is 0.0109 at $G = 64$ (close to $\ln 2/(G-1)$), whether or not $\pi_\theta(x)$ already exceeds its target.

Stationary points of the expected update. Suppose that the model can assign any probability to each outcome. We parameterize it by a softmax over outcomes with logits $\theta = (\theta_x)_{x\in\mathcal{X}}$, so $\pi_\theta(x) \propto \exp(\theta_x)$ and

$$\frac{\partial \pi_\theta(x)}{\partial \theta_{x'}} = \pi_\theta(x)\big(\mathbb{1}[x = x'] - \pi_\theta(x')\big).$$

Conditioning on x_i and applying Lemma 2(b) with $f = \bar{a}$ gives

$$\mathbb{E}[A_i\, s(x_i)] = \sum_x \bar{a}(x)\,\nabla_\theta \pi_\theta(x).$$

Substituting the softmax derivative, the component of this update along $\theta_{x'}$ is

$$\sum_x \bar{a}(x)\,\pi_\theta(x)\big(\mathbb{1}[x = x'] - \pi_\theta(x')\big) = \pi_\theta(x')\Big(\bar{a}(x') - \sum_x \pi_\theta(x)\,\bar{a}(x)\Big).$$

For this paragraph we drop the positivity assumption of Appendix B and read the right-hand side as a vector field on the closed simplex, where outcomes may have probability zero. There $\bar{a}(x')$ is given by its binomial formula, such as Equation 12 for a low-mass x', and the component along $\theta_{x'}$ vanishes whenever $\pi(x') = 0$. We call a distribution π a stationary point when every component vanishes, that is, when $\bar{a}$ takes a common value λ on the support of π. Outcomes outside the support, which are the outcomes that the stationary point sets to zero, impose no condition. The condition does not depend on the parameterization. In the interior of the simplex, $\sum_x \bar{a}(x)\,\nabla_\theta \pi_\theta(x) = 0$ holds for any parameterization that can move π in every direction of the simplex exactly when $\bar{a}$ is constant.

At a stationary point, every low-mass outcome x in the support satisfies $2(1-\pi(x))^{G-1} - 1 = \lambda$ by Equation 12. The map $v \mapsto 2(1-v)^{G-1} - 1$ is strictly decreasing on $[0,1]$, so all these outcomes have the same probability, $1 - \big((1+\lambda)/2\big)^{1/(G-1)}$. Their total probability is this value times their number. It depends on how many low-mass outcomes the support contains and on the common level λ, but not on their target probabilities.

When the target is not stationary. At $\pi = q$, the support is the support of q. Suppose two low-mass outcomes x and x' have $q(x) \neq q(x')$. Equation 12 gives them the expected advantages $2(1-q(x))^{G-1} - 1$ and $2(1-q(x'))^{G-1} - 1$, which differ because the map above is strictly decreasing. The expected advantage is therefore not constant on the support, and q is not a stationary

point. A Zipf target gives a different probability to each outcome, so it satisfies this condition as soon as it has two low-mass outcomes.

The requirement of two low-mass outcomes with different target probabilities cannot be dropped. Consider the binary target with outcomes a and b and

$$q(a) = 1 - 2^{-1/(G-1)}, \qquad q(b) = 2^{-1/(G-1)}.$$

For every $G \geq 2$ this target is stationary, because both outcomes have expected advantage 0. Outcome a is low-mass, and Equation 12 gives it the expected advantage $2 \cdot 2^{-1} - 1 = 0$. For $G \geq 3$, outcome b is not low-mass, so we compute its expected advantage directly. Since $q(b) > (G-2)/(G-1)$, the only value of the leave-one-out frequency at or above $q(b)$ is 1. The sign of b is therefore -1 exactly when all $G-1$ other rollouts produce b, which has probability $q(b)^{G-1} = 1/2$, and $+1$ otherwise. Its expected advantage is $\frac{1}{2} - \frac{1}{2} = 0$. For $G = 2$, both outcomes are low-mass and have expected advantage 0 by Equation 12.

Centering does not change these conclusions. GRPO trains on the centered advantage $\tilde{A}_i = A_i - \bar{A}$. For $G \geq 3$, the two conclusions about stationary points also hold for the expected centered update, with $G-2$ in place of $G-1$ in the exponent. To show this, we compute the expected centered advantage of a low-mass outcome in three steps. It turns out to be a strictly decreasing function of $\pi_\theta(x)$ plus a constant, as in Equation 12.

Step 1: separate the rollout's own advantage. By the same conditioning argument as above, the expected centered update is $\sum_x \bar{a}_{\mathrm{c}}(x)\, \nabla_\theta \pi_\theta(x)$. Since $\bar{A}$ contains A_i/G, the expected centered advantage is

$$\bar{a}_{\mathrm{c}}(x) = \mathbb{E}[\tilde{A}_i \mid x_i = x] = \frac{G-1}{G}\, \bar{a}(x) - \frac{1}{G} \sum_{j \neq i} \mathbb{E}[A_j \mid x_i = x].$$

Step 2: the advantage of another rollout. Fix $j \neq i$ and condition also on $x_j = x'$. Let c be the number of the $G-2$ rollouts other than i and j whose outcome is x', a binomial count with $G-2$ trials and success probability $\pi_\theta(x')$. Define

$$h(x') = \mathbb{E}\Big[\operatorname{sign}\Big(q(x') - \frac{c}{G-1}\Big)\Big], \qquad h^+(x') = \mathbb{E}\Big[\operatorname{sign}\Big(q(x') - \frac{c+1}{G-1}\Big)\Big].$$

If $x' \neq x$, rollout i does not count toward $\hat{p}_{-j}(x')$, so $\mathbb{E}[A_j \mid x_i = x, x_j = x'] = h(x')$. If $x' = x$, rollout i adds one to the count, and the conditional expectation is $h^+(x)$. Averaging over x',

$$\mathbb{E}[A_j \mid x_i = x] = \sum_{x'} \pi_\theta(x')\, h(x') + \pi_\theta(x)\big(h^+(x) - h(x)\big).$$

The first term does not depend on x. The expression is the same for every $j \neq i$, so the sum in Step 1 is $G-1$ times it.

Step 3: evaluate at a low-mass outcome. For a low-mass x, the grid argument gives $h(x) = 2(1-\pi_\theta(x))^{G-2} - 1$ and $h^+(x) = -1$, because a count of at least one puts the leave-one-out frequency at $1/(G-1) > q(x)$ or above. Write $v = \pi_\theta(x)$. Substituting Step 2 and Equation 12 into Step 1,

$$\begin{aligned}\bar{a}_{\mathrm{c}}(x) &= \frac{G-1}{G}\Big(2(1-v)^{G-1} - 1 + 2v(1-v)^{G-2}\Big) + C \\ &= \frac{G-1}{G}\Big(2(1-v)^{G-2}\big[(1-v)+v\big] - 1\Big) + C = \frac{G-1}{G}\Big(2(1-v)^{G-2} - 1\Big) + C,\end{aligned}$$

where $C = -\frac{G-1}{G} \sum_{x'} \pi_\theta(x')\, h(x')$ does not depend on x. The first line uses $-v\big(h^+(x) - h(x)\big) = 2v(1-v)^{G-2}$.

For $G \geq 3$, $\bar{a}_{\mathrm{c}}$ is therefore a constant plus a strictly decreasing function of $\pi_\theta(x)$ alone. At a stationary point of the centered update, all low-mass outcomes in the support again share one probability, and $\pi = q$ is again not stationary when two low-mass outcomes have different target probabilities. The constant C moves the point where $\bar{a}_{\mathrm{c}}$ changes sign away from $1 - 2^{-1/(G-1)}$, and the binary example above is stationary only for the uncentered update.

E.4 Simulating the Sign Witness on Zipf Targets

The proof shows that the sign witness cannot hold low-mass outcomes with different target probabilities at their targets. It does not say whether their total probability ends too high or too low. We therefore simulate the replicator flow of the expected update,

$$\frac{d}{dt}\log \pi_t(x) = \bar{a}(x) - \sum_{x'} \pi_t(x')\,\bar{a}(x'),$$

which has the same stationary points. Here the subscript t is time, and $\bar{a}$ is evaluated at π_t. We use $G = 64$ and three Zipf targets, with 200 outcomes and exponent 1.2, 1,000 outcomes and exponent 1.5, and 50 outcomes and exponent 2.0. The flow starts from $\pi_0(x) \propto x^{-2.5}$ on the outcomes $x = 1, 2, \ldots$ of the target. The sign flow raises the total probability of the low-mass outcomes to 1.5 to 2.4 times their target probability and stops at TV 0.13 to 0.47 from the target. The same flow with the witness advantage reaches the target, with TV below 10^{-4}.

E.5 Connection to Property-Ratio Alignment

Huang et al. (2026) train with a per-group $0/1$ reward that pushes the fraction of rollouts with a binary property toward a target ratio r_t (their notation). Let $x_i \in \{0, 1\}$ be the property value of rollout i, let $\hat{r}$ be the group's empirical ratio, and let $\epsilon \geq 0$ be a tolerance. Their reward is

$$R_i = \begin{cases} x_i & \text{if } \hat{r} < r_t - \epsilon, \\ 1 - x_i & \text{if } \hat{r} > r_t + \epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

and training stops once the ratio lies inside the band. Set $q = \text{Bernoulli}(r_t)$ on $\{0, 1\}$. The sign witness computed from the full group is

$$\sigma_i = \operatorname{sign}\big(q(x_i) - \hat{p}(x_i)\big) = (2x_i - 1)\operatorname{sign}(r_t - \hat{r}),$$

since $q(1) - \hat{p}(1) = r_t - \hat{r}$ and $q(0) - \hat{p}(0) = \hat{r} - r_t$. On a group whose ratio lies outside the band, checking the two cases shows $R_i = (1 + \sigma_i)/2$. Constants cancel under centering, so $R_i - \bar{R} = (\sigma_i - \bar{\sigma})/2$. If the group also contains both property values, both standard deviations are positive and the scale cancels too, so their standardized GRPO advantage equals the standardized sign witness. Inside the band, their reward is zero for every rollout, whereas σ_i is nonzero unless $\hat{r} = r_t$. Huang et al. (2026) also give a multi-class variant. It rewards rollouts from under-represented attributes with 1 and those from over-represented attributes with 0, and removes attributes whose ratio lies inside the band. On a finite alphabet, this is the full-group sign witness shifted and scaled as above, restricted to outcomes outside the band. A binary target with moderate r_t has no low-mass outcomes, so the regime of part (iii) does not arise in their binary setting.

F Experimental Setup

This appendix gives the details behind Section 4.1: the prompts, the parser, the choice of targets, the training settings, and the evaluation and statistics.

F.1 Prompts

Every draw prompt for a synthetic target uses the system message below. The training targets, the unseen-parameter targets, and the hypergeometric targets of Table 6 use the original format. The user message in this format states the family and its parameters and asks for one outcome. For the geometric, Poisson, and Zipf families it also gives the PMF formula. The biased-coin prompt gives both probabilities. The binomial prompt gives the parameters and the range of outcomes, and the hypergeometric prompt gives only the parameters. Two examples, for the target of Figure 1a and for a geometric target:

> **System.** You simulate random draws from probability distributions. When asked for a draw, you output exactly one outcome and nothing else.
> **User.** A biased coin lands on Heads with probability 0.005 and on Tails with probability 0.995. Flip the coin once and report the single outcome. Respond with exactly one word -- either 'Heads' or 'Tails' -- and nothing else.
> **User.** A geometric distribution has success probability p = 0.551. Its outcomes are the positive integers 1, 2, 3, ... (the number of independent trials up to and including the first success), with probability P(k) = (1-p)^(k-1) * p. Draw a single sample from this distribution and report the single integer outcome. Respond with only the integer and nothing else.

The unseen-family prompts use the evaluation format, a second template with the same system message. The retrain on the evaluation format (Section 4.2) uses this template for its training targets as well. The template states the family and its parameters, gives a PMF formula for the maximum of dice and the logarithmic series and a description in words for the other families, and adds a sentence that names the valid outcomes. An example for the maximum of dice:

> **User.** A maximum of dice distribution has the maximum of k = 3 independent dice, each with m = 4 sides. The probability that the maximum equals x is (x/m)^k - ((x-1)/m)^k. The valid outcomes are the integers from 1 to 4 inclusive. Draw one random sample from this distribution. Respond with only the outcome and nothing else.

Stating prompts. To test whether the model knows the target, a separate prompt asks it to name the distribution and state its parameters, under the system message "You are a precise probability assistant. Answer concisely in the exact format requested." There is one template per family. The geometric template reads as follows.

> **User.** Consider a random process of repeated independent trials, each succeeding with probability 0.551, where you count the number of trials up to and including the first success. Name the probability distribution this follows and state its parameter value. Answer in exactly this format: '<distribution name>, p=<number>'.

The coin template asks instead for the probability of each outcome. An answer is scored correct when the family keyword appears (for every family except the coin) and every parameter matches: probabilities within 0.0015, integers exactly, and the Poisson rate within 0.5% or 0.01, whichever is larger.

F.2 Parser and Invalid Outputs

The parser is strict. It strips surrounding whitespace, one layer of matching quotes or backticks, and one trailing period, and the rest of the response must be exactly one outcome. For the biased coin, the outcome must be Heads or Tails, in any capitalization. For the integer families, it must be an integer inside the support of the target. Everything else maps to $\perp$. The common cases are extra words ("The outcome is 3"), more than one outcome, a non-integer, and an integer outside the support, such as 0 for a geometric target or 9 for a binomial target with 8 trials. Training and evaluation use the same parser. In training, $\perp$ is an outcome with target probability 0, so an invalid rollout receives the witness advantage $-2\hat{p}_{-i}(\perp)$ before centering, which is negative whenever another rollout of the group is also invalid. In evaluation, invalid responses are left out of the empirical distribution and their rate is reported separately.

F.3 Targets

For each training family, we draw candidate parameter settings from the Spectrum Suite generators (Sorensen et al., 2026). We keep a candidate if a perfect sampler's expected TV at $5{,}000$ draws is at most 0.10, estimated from $1{,}000$ Monte Carlo replicates. This screen removes targets with more outcomes than a few thousand draws can resolve. From the kept candidates, we select 20 settings spaced by quantiles of the primary parameter, such as p for the geometric family. We sort them by this parameter and number them from 0. Ranks 2, 7, 12, and 17 form the unseen-parameter set, and the other 16 are training targets. Each target is also assigned the smallest $n \in \{500, 2{,}000, 5{,}000\}$ at which the perfect sampler's expected TV is at most 0.10. This n is the number of evaluation draws for unseen-parameter targets (Appendix F.5). The five unseen families are screened the same way,

with 20 targets each. Hypergeometric targets come from the Spectrum Suite generator, and the other four families from our own samplers. In the rest of the appendix, we write occupancy for the number of empty boxes, triangular for the discrete triangular, and log-series for the logarithmic series.

F.4 Training

Table 3 lists the settings shared by all GRPO runs. We follow Liu et al. (2025) in two choices. First, the loss normalizes by a constant length instead of each response's length. Second, the advantages are centered by the group mean but not divided by the group's standard deviation of rewards. The original GRPO (Shao et al., 2024) divides by this standard deviation. We do not, because the size of the witness advantage measures how far the model is from the target (Section 3.1), and dividing would remove this information and change the expected update derived in Appendix C.

Table 3: Settings shared by all GRPO runs. Exceptions are listed in the text.

setting	value
model	Qwen2.5-1.5B-Instruct (Qwen et al., 2024), full fine-tuning, bfloat16 mixed precision
implementation	GRPO in TRL (von Werra et al., 2020), generation with vLLM (Kwon et al., 2023) on the same GPU
group size G	64
prompts per step	4 (256 rollouts)
steps	600, one gradient step per generation batch (on-policy)
optimizer	AdamW, constant learning rate 2×10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, no weight decay
gradient clipping	1.0
PPO clip range ϵ	0.2
KL weight toward the initial model	0.02
loss normalization	constant length (Liu et al., 2025)
advantage	reward minus group mean, no division by the standard deviation
decoding in training	temperature 1, at most 24 response tokens
hardware and time	one 48 GB GPU; about one GPU-hour per main run on the synthetic distributions, and 1.5 to 1.7 GPU-hours per run of the group-size sweep

Exceptions. The group-scalar reward uses one prompt of 256 rollouts per step, split into four subgroups of 64, and is otherwise identical. The group-size sweep keeps 256 rollouts per step. The witness runs use $256/G$ prompts per step, from 32 at $G = 8$ to 1 at $G = 256$, and the group-scalar runs use one prompt split into $256/G$ subgroups. The entropy-bonus baseline adds TRL's entropy bonus with coefficient 0.01 to a reward for valid output. Its mean valid rate during training was 0.95. The opinion-distribution runs use 1,600 training prompts (800 urn and 800 GlobalOpinionQA) and take about 2.5 GPU-hours. The sibling-discovery runs use 400 steps and take 1.3 to 2.4 GPU-hours, and their evaluation allows up to 32 response tokens.

Cross-entropy baseline. The supervised baseline minimizes, for each training prompt, $-\sum_x q(x) \log P_\theta(y_x)$, where y_x is the exact response string for outcome x followed by the end-of-turn token. This loss replaces the outcome probability $\pi_\theta(x)$ of Section 4.1 by the probability of one canonical response, which is a lower bound on $\pi_\theta(x)$. The sum runs over a finite set of outcomes of each training target, on which q is renormalized. For the coin it has both outcomes. For the other families except Zipf it has the outcomes between the 10^{-4} and $1 - 10^{-4}$ quantiles of the target, and for Zipf targets the most likely outcomes that cover 99% of the probability, at most 2,000. We fully fine-tune the same model with AdamW (constant learning rate 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, gradient clipping 1.0) on minibatches of 8 prompts. A time cap of 36 minutes stopped training after 11 epochs over the 80 training targets (110 steps, about 0.6 GPU-hours), and we evaluate the final checkpoint. We did not tune this recipe to preserve capability. Unlike our GRPO runs, it has no KL term toward the initial model, and its learning rate is five times the GRPO learning rate. Zhang et al. (2024) instead tune low-rank adapters for at most 50 steps with early stopping, and

report little change on MT-Bench. Our full fine-tune costs 10 MMLU points and 23 IFEval points (Section 4.2). This cost therefore reflects our recipe as well as the objective.

F.5 Evaluation and Statistics

Sampling error. Training targets and unseen families are evaluated with $n = 500$ draws per target. Unseen-parameter targets and the targets of Figure 1b use the n assigned above. Excess TV subtracts the expected TV of a perfect sampler at the same n. We estimate this quantity once for each target from 2,000 Monte Carlo replicates. A reduction in excess TV compares medians over the targets of a set, $1 - \mathrm{median}(\text{excess TV}_{\text{trained}})/\mathrm{median}(\text{excess TV}_{\text{untrained}})$.

Capability. MMLU uses the full test set (14,042 questions) with greedy decoding and counts an answer as correct when the first letter from A to D in the response is the right one. IFEval uses its 541 prompts and reports instruction-level strict accuracy. The noise band of each benchmark is the 95% bootstrap interval of the untrained model's score, from 2,000 replicates. Its half-width (0.008 for MMLU and 0.033 for IFEval) was fixed before training.

Intervals. Unless stated otherwise, an interval is a 95% paired bootstrap interval over the targets of a set, from 2,000 replicates. Rows that pool seeds resample targets within each seed. Capability intervals resample benchmark items, clustered by prompt for IFEval. On GlobalOpinionQA, the intervals of the reduction resample question stems.

G Additional Results on Synthetic Distributions

This appendix adds detail to Section 4.2. We give per-family results, the full comparison of the three rewards, the training-free baselines, a split of the remaining error, a test of whether the model relies on the listed outcomes, and a study of how transfer depends on the variety of training targets.

G.1 Results per Family

Table 4 reports all three synthetic evaluation sets for the untrained model, supervised cross-entropy, and the witness advantage. Table 5 gives the median excess TV per family, for the three rewards on the training families and for the two witness checkpoints on the unseen families. The witness advantage lowers excess TV on every training family, and its largest remaining errors are on Poisson and binomial targets (Appendix G.4). On the unseen families, the checkpoint trained on the original prompts improves four families and leaves occupancy about the same (0.615 to 0.623). The retrain on prompts in the evaluation format improves all five. Supervised cross-entropy, trained on the original prompts, removes 31% of the excess TV on the unseen families, against 36% for the witness advantage trained on the same prompts.

Table 4: Sampling error on the three synthetic evaluation sets and general capabilities. Supervised cross-entropy fits the training targets and unseen parameters more closely than the witness advantage but loses 10 MMLU and 23 IFEval points. Sampling error is the median excess TV on the 64 training and 16 unseen-parameter targets outside the Zipf family and on the 100 unseen-family targets. Capability is accuracy, with 95% bootstrap half-widths of 0.008 (MMLU) and 0.033 (IFEval).

		untrained	cross-entropy	witness (ours)
sampling error	training targets	0.477	0.053	0.101
	unseen parameters	0.483	0.094	0.161
	unseen families	0.589	0.406	0.376
capability	MMLU	0.584	0.482	0.583
	IFEval	0.512	0.281	0.528

G.2 Comparing the Three Rewards

Table 6 gives the paired comparisons on unseen targets behind Section 4.2. The witness advantage beats the group-scalar reward on unseen parameters in each of three seeds. The sign witness is

Table 5: Median excess TV per family. Top: training families after 600 steps, with the lowest value per family in bold. Bottom: unseen families and all 100 unseen-family targets pooled, for the untrained model and for the witness advantage trained on the original prompts or on prompts in the evaluation format. All values use $n = 500$ draws per target. Figure 1b uses the n assigned to each target, so its untrained values differ from those in the top block.

training family	untrained	witness (ours)	group-scalar	sign witness
biased coin	0.361	0.034	0.440	**0.021**
geometric	0.372	0.050	0.309	**0.028**
binomial	0.516	**0.176**	0.511	0.255
Poisson	0.691	0.421	0.696	**0.263**
Zipf	0.563	**0.024**	0.447	0.523

		witness advantage (ours)	
unseen family	untrained	original prompts	evaluation-format prompts
log-series	0.606	0.139	0.197
triangular	0.299	0.221	0.184
maximum of dice	0.662	0.436	0.522
hypergeometric	0.654	0.564	0.545
occupancy	0.615	0.623	0.547
pooled (100 targets)	0.589	0.376	0.447

indistinguishable from the witness advantage on 36 unseen targets outside the Zipf family. These are the 16 unseen-parameter targets of the other four training families and the 20 hypergeometric targets, evaluated with prompts in the original format. On the Zipf training targets the two differ (Figure 2b).

Table 6: Paired per-target differences in TV on unseen targets outside the Zipf family. The top block uses the 16 unseen-parameter targets, and the bottom block adds 20 hypergeometric targets. Each difference is the TV of the second method minus the TV of the first, so a positive value favors the first. The superscript gives the larger side of the 95% bootstrap interval.

		median difference
witness advantage vs. group-scalar reward, 16 unseen-parameter targets, no Zipf	seed 0	$0.248^{\pm 0.131}$
	seed 1	$0.153^{\pm 0.137}$
	seed 2	$0.223^{\pm 0.129}$
	pooled	$0.211^{\pm 0.073}$
sign witness vs. 36 unseen targets	group-scalar reward	$0.075^{\pm 0.123}$
	witness advantage	$0.014^{\pm 0.023}$

On the unseen families, the witness advantage and the group-scalar reward are closer. Table 7 varies the group size at a fixed 256 rollouts per step, and the intervals of the two rewards overlap at every group size. Two further sets of witness runs have no group-scalar counterpart. At $G = 256$, the witness advantage removes 31.1% of the excess TV (interval $[22.1, 40.6]$). A second seed at $G = 8$, 16, 32, and 128 removes 38.2%, 40.3%, 36.9%, and 32.0%.

A reward for diversity alone. Rewarding diversity without reference to the target makes sampling worse than no training. Outside the Zipf family, the entropy-bonus run (Appendix F.4) reaches a median excess TV of 0.663 on the training targets, against 0.477 for the untrained model, and 0.730 on the unseen parameters, against 0.483.

G.3 Training-Free Baselines

Table 8 compares the training-free baselines of Section 4.2 on the unseen families. None of them removes more than 19% of the excess TV, against 36% for training with the witness advantage. All settings use the untrained Qwen2.5-1.5B-Instruct and 500 draws per target unless stated otherwise.

Table 7: Group-size sweep with 256 rollouts per step, seed 0. Each entry is the reduction in excess TV on the 100 unseen-family targets, in percent, and the superscript gives the larger side of the 95% paired bootstrap interval. All runs were trained on the original prompts and evaluated on the unseen-family prompts, so they compare with the 36% of the main run, not with the 24% of the retrain on the evaluation format. The $G = 64$ witness run is separate from the main run.

group size G	witness advantage (ours)	group-scalar reward
8	$39.4^{\pm 12.9}$	$30.8^{\pm 10.4}$
16	$36.2^{\pm 13.8}$	$30.3^{\pm 10.7}$
32	$35.8^{\pm 10.8}$	$28.6^{\pm 8.7}$
64	$34.4^{\pm 11.9}$	$29.8^{\pm 12.9}$
128	$35.8^{\pm 9.5}$	$24.6^{\pm 12.7}$

Settings that apply a method on top of the witness-trained model use a reproduction of the main run, trained with the same recipe. It reaches a pooled excess TV of 0.380, against 0.376 for the main run, and their paired per-target difference is indistinguishable from zero (interval $[-0.002, 0.002]$).

Table 8: Training-free methods on the 100 unseen-family targets, applied to the untrained Qwen2.5-1.5B-Instruct, whose median excess TV is 0.589. Reduction is relative to the untrained model. Both temperatures are chosen with knowledge of the targets.

method	excess TV	reduction
renormalization	0.600	−2%
seed conditioning	0.546	7%
verbalized sampling (best case)	0.518	12%
global temperature (2.76)	0.514	13%
per-target oracle temperature	0.478	19%
witness advantage (ours)	**0.376**	**36%**

Verbalized sampling. Verbalized sampling (Zhang et al., 2025a) asks the model to list outcomes with probabilities and samples from the parsed list. Our prompt asks for five outcomes with probabilities. We parse the list with the same parser, renormalize it, and draw 500 outcomes from it. With one list per target, most lists do not parse. The published prompt, a variant that also names the support, and the published XML form give pooled excess TV between 0.945 and 0.959, with a parse rate of 0.05 to 0.45. With the published prompt, 46 of 100 targets yield lists whose probabilities are all zero, and 14 more list only outcomes outside the support.

Table 8 reports the best case, which pools ten verbalizations at temperature 1 with the support-naming prompt (0.518). At least one of the ten lists parses on every target. Even the lists that parse are wrong. Their median TV to the target is 0.71, and 0.78 for greedy lists. Applied on top of the witness-trained model, verbalized sampling is equally poor (0.957 with the support-naming prompt). Zhang et al. (2025a) evaluate verbalized sampling on frontier API models and on open models with 70B parameters or more, and report that its gains grow with model scale, so our results describe it only at 1.5B.

Seed conditioning. Seed conditioning (Nagarajan et al., 2025) adds a random string to the start of the prompt, here 16 characters. With greedy decoding, three seed formats, one of them with 64 characters, give an excess TV of 0.834 to 0.847, against 0.904 for greedy decoding without a seed, and a median of two to four distinct outcomes per target. At temperature 1, the seeded prompt reaches 0.546 and the same prompt with a constant seed reaches 0.542 (paired difference 0.0001, interval $[-0.003, 0.005]$), so the randomness of the seed contributes nothing. Nagarajan et al. (2025) add seeds during both training and sampling, and we test only the sampling part.

Renormalization. Renormalization sets the probability of every invalid outcome to zero and rescales the rest to sum to one. Applied to the model's sequence-level probabilities over the valid outcomes, it gives 0.600 (paired difference from plain sampling $+0.007$, interval $[0.002, 0.013]$). Taking the most likely valid outcome gives 0.903. Editing the logits of a single token is possible on

only 29 of the 100 targets, where it reaches 0.725. On top of the witness-trained model, renormalization changes little (0.373 against 0.380).

Temperature. A temperature chosen for each target to minimize the exact TV to q gives 0.478. The single best temperature for all targets, 2.76, gives 0.514. Both are tuned with knowledge of the targets, so they bound what temperature scaling can do. Replacing the model's distribution by q itself gives 0.001, the excess TV of a perfect sampler, which is zero up to Monte Carlo error. On the 64 training targets outside the Zipf family and the 36 unseen targets of Table 6, temperatures 1.3 and 1.5 reduce the excess TV by at most 9.5% on the training targets and 6.5% on the 36 unseen targets, and min-p sampling and an explicit instruction to sample at random increase it.

Larger models. Scaling the untrained model does not help. On the unseen families, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Qwen et al., 2024), Llama-3.2-3B-Instruct, and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) all have a higher median excess TV than the 1.5B model, 0.63 to 0.72 against 0.59 (Figure 1c).

G.4 Where the Remaining Error Is

Figure 4 reports two diagnostics. Panel (a) checks part (i) of Proposition 3 in training. Panel (b) splits the remaining error of the witness advantage on its two hardest families.

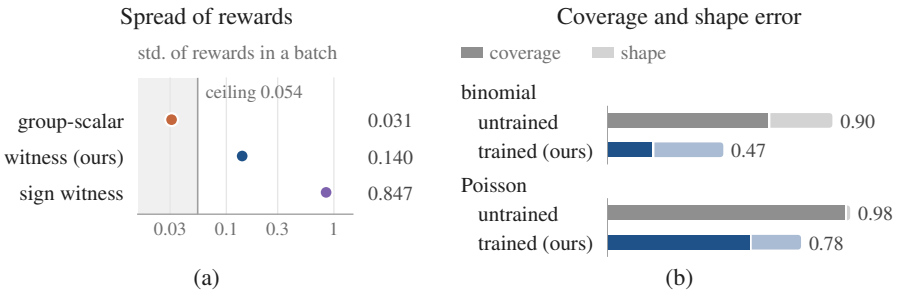

Figure 4: (a) Standard deviation of the rewards in a batch, averaged over training, on a log scale. For the group-scalar reward, it equals the standard deviation of the advantages, and it lies below the ceiling of Proposition 3(i) (vertical line). (b) TV on binomial and Poisson training targets, split into a coverage term (dark) and a shape term (light), for the untrained model and after training with the witness advantage. Each bar stacks the medians of the two terms over four targets, so its total can differ from the median TV. Training mainly reduces the coverage term.

Spread of the rewards. For each run we log the standard deviation of the 256 rewards in a batch and average it over the 600 steps. It is 0.031 for the group-scalar reward, 0.140 for the witness advantage, and 0.847 for the sign witness. For the group-scalar reward, all rewards in a batch come from one prompt and differ from the advantages by a constant, so 0.031 is also the spread of its advantages. It lies below the ceiling $\sqrt{3/4}/(2\sqrt{64}) = 0.054$ of Proposition 3(i). For the other two rewards, a batch contains four prompts and GRPO subtracts a separate mean for each, so their logged spread also includes differences between prompts. The group-scalar reward gives at most four distinct advantages per batch, whereas the witness advantage can give 256.

Coverage and shape. On binomial and Poisson targets, the untrained model puts much of its probability on outcomes that the target almost never produces. To separate this error from errors in the relative probabilities, we split the outcomes of each target into its bulk, the interval $[F^{-1}(10^{-4}), F^{-1}(1-10^{-4})]$, where F is the target's cumulative distribution function, and the rest. The TV then splits exactly into two terms. The coverage term is half the absolute error on produced outcomes outside the bulk plus half the target probability of the outcomes the model never produced. The shape term is half the absolute error on produced outcomes inside the bulk. We compute both from 500 draws on the four binomial and four Poisson training targets with the highest excess TV after training.

Before training, most of the error is coverage. On binomial targets the untrained model's median TV is 0.87, and the median share of coverage is 71%. On Poisson targets the median TV is 0.98, and the median share of coverage is 98%, with a sample mean a tenth of the target mean. Training with the witness advantage lowers the coverage term from 0.65 to 0.18 on binomial targets and from 0.95 to 0.57 on Poisson targets, and raises the ratio of the Poisson sample mean to the target mean from 0.10 to 0.51. After training, the shape terms are 0.29 and 0.21. The remaining binomial error is therefore mostly shape, while most of the remaining Poisson error is still coverage. On both families the draws are much more spread out than the target. The median ratio of the standard deviation of the draws to that of the target is between 7 and 15, before and after training.

G.5 Copying the Listed Outcomes

Every unseen-family prompt names the valid outcomes. A model could lower its excess TV by copying this list without reading the rest of the prompt. We first explain why the prompts list the outcomes. We then remove the list at evaluation and find that most of the improvement remains.

Why the prompts list the outcomes. Before we fixed the unseen-family prompts, we evaluated the witness checkpoint on the 20 hypergeometric targets in the original format, whose prompts do not name the valid outcomes. Training moved excess TV only from 0.625 to 0.614, and the fraction of valid outputs fell from 0.64 to 0.40. When few items are drawn, the support is tiny. When one item is drawn, it is $\{0, 1\}$, and the model emits counts far above it. A model cannot know the support of a family it has never seen, so the unseen-family prompts name it.

Removing the list. We evaluate the untrained and the trained model with and without the sentence that names the valid outcomes, on all 100 unseen-family targets. The trained model is the reproduction of Appendix G.3. Training improves excess TV by a median of 0.176 per target (interval $[0.124, 0.227]$) with the sentence and 0.135 ($[0.120, 0.173]$) without it, so 77% of the median improvement remains when the list is removed. The median per-target difference between the two improvements is 0.057 ($[0.010, 0.103]$), so the list contributes, but most of the improvement does not depend on it.

On log-series, the improvement without the sentence is 0.226 ($[0.204, 0.247]$). On occupancy, the improvement is larger without the sentence (difference -0.054, interval excluding zero). Removing the sentence also changes the prompt format, so this test does not separate the content of the list from the change of format.

G.6 Variety of the Training Targets

Transfer to unseen families may depend on how varied the training targets are. To test this at matched compute (600 steps, same group size and decoding), we compare the five-family retrain with the witness advantage trained on three other training sets, all in the evaluation format. One has a single family (the 16 geometric targets), one a single target (a geometric target with $p = 0.551$), and one fifteen families (240 targets). Table 9 gives the results.

Table 9: Median excess TV on the 100 unseen-family targets after training on sets of different variety, all with prompts in the evaluation format and the same compute. Reduction is relative to the untrained model (0.589).

training set	excess TV	reduction
one geometric target	0.397	33%
one family (16 geometric targets)	0.462	22%
five families (80 targets)	0.447	24%
fifteen families (240 targets)	0.394	33%

Widening from five to fifteen families lowers excess TV on the typical target (median paired difference -0.038, interval $[-0.045, -0.025]$).

The single-target run reaches about the same pooled value as the fifteen-family run, 0.397 against 0.394, and a second seed of the single-target run reaches 0.383. Its gains are narrow, however.

Most come from log-series, the one unseen family that shares the geometric family's support and decreasing shape (excess TV from 0.606 to 0.119), and from occupancy (0.615 to 0.456). On hypergeometric and maximum-of-dice targets it ends worse than the untrained model (0.654 to 0.693 and 0.662 to 0.701), while the five-family run improves every unseen family. The single-target model also does not read the stated parameter. On unseen geometric parameters, the single-family model has excess TV 0.02 to 0.11 for p from 0.155 to 0.914, while the single-target model is accurate only near its training value and reaches 0.44 at $p = 0.155$.

Because compute is matched, fewer training targets also mean more visits to each prompt ($2{,}400$ group visits for the single target against 10 per target for fifteen families), so variety and repetition change together in this comparison.

H Opinion Distributions and Structured Outputs

This appendix gives the details of Section 4.3. Table 10 lists the results on both tasks.

Table 10: Stated opinion distributions (top) and structured outputs (bottom), in median excess TV. Trained values are means over seeds, and reductions give the range over seeds: three for the opinion tasks, two for uniform sibling targets, and one for non-uniform sibling targets. The structured-output targets are evaluated on 60 test graphs.

opinion evaluation set	Qwen2.5-1.5B	Llama-3.1-8B	trained (ours)	reduction
urn draws	0.33	0.23	0.04	80 to 91%
GlobalOpinionQA	0.26	0.10	0.03	86 to 92%
NYTimes task	0.48	0.32	0.15	61 to 72%
implicit variant	0.31	0.21	0.11	56 to 69%

structured-output target	Qwen2.5-1.5B	best decoding	trained (ours)	reduction
sibling pairs, uniform	0.568	0.540	0.358	35 to 39%
sibling pairs, non-uniform	0.533	0.458	0.280	47%

Opinion distributions. The opinion prompts state the answer options, name them as the valid outcomes, and list the target probability of each option to three decimals. The implicit variant removes the probability line. Each opinion prompt is evaluated with $n = 250$ draws instead of 500. The training set has $1{,}600$ prompts, 800 from urn draws and 800 from GlobalOpinionQA, and the test set has 200 prompts from each task. The urn prompts describe $1{,}000$ urns generated by the Spectrum Suite code, which we split at random into 800 training and 200 test urns. The split does not remove duplicates, and 4 test urns have the same description as a training urn. The GlobalOpinionQA prompts come from 75 question stems. The test prompts use 18 of them, and the training prompts use the other 57, so no test question appears in training.

Comparisons on opinion tasks. On the test prompts of both tasks and on the NYTimes task, the trained model has lower excess TV than the best prompting method. On the test prompts, it also has lower excess TV than the best temperature. Across seeds, its paired margin in excess TV is 0.14 to 0.32 over the best prompting method and 0.19 to 0.23 over the best temperature, and every interval excludes zero. A group-scalar run on the opinion tasks stopped at step 77 of 600, when the mean rate of valid output over the previous 20 steps fell below 0.5.

Structured outputs. We adapt the sibling-discovery task of Nagarajan et al. (2025). In their version, the model learns the graph during training, the prompt does not show it, and the model generates two siblings together with their parent. Our prompt instead names a parent and its children and asks for one pair of siblings, so the valid outputs are all pairs of children. We state two kinds of targets over these pairs. A uniform target asks for a pair drawn uniformly at random and does not list the pairs, so the model must build them from the list of children. A non-uniform target lists each pair with its probability. The uniform target tests whether the model finds the valid outputs, and the non-uniform target tests whether it also follows the stated probabilities over them. For uniform

targets, we train on 800 graphs with 8 to 20 children and evaluate on 60 test graphs with 8 children (28 pairs). For non-uniform targets, we train on 200 graphs with 6 children and evaluate on 60 test graphs of the same size.

On uniform targets, training removes 35% and 39% of the excess TV in two seeds, compared with 5% for the best decoding baseline and 8% for the group-scalar reward, which reaches 0.523. Training also raises the median fraction of valid pairs that the model produces at least once in 500 draws from 0.71 to 0.93 (first seed), so the trained model finds more of the valid outputs. On non-uniform targets, training removes 47% of the excess TV, compared with 14% for the best decoding baseline.