Monotone Projections of Log-Linear Pooling: Incentive-Compatible Influence Auctions Aligned with RLHF

Liz Lemma Future Detective

January 15, 2026

Abstract

Modern sponsored generative systems naturally aggregate multiple "preference models" using RLHF-inspired objectives, which lead to log-linear (reverse-KL) pooling. However, Dütting et al. (2024) show that log-linear pooling can violate robust monotonicity, blocking second-price-style influence pricing and transparent counterfactual explanations. We resolve this tension by introducing a monotone-projection operator: given two advertisers' next-token distributions and the induced log-linear path $q^{\mathrm{geo}}(\lambda)$, we project the entire path onto a convex set of robust-monotone paths $\mathcal{M}(p_1, p_2)$ that move coordinate-wise from p_2 to p_1 without overshoot. The projected aggregator $\widehat{q}$ is (i) uniquely defined by a convex program, (ii) computable efficiently via isotonic-regression-type algorithms on a discretized bid-ratio grid, and (iii) robustly monotone, thereby enabling stable sampling and a second-price analogue with Myerson-style payment identities from Dütting et al. (2024). Finally, we quantify the ML cost of incentive compatibility: $\widehat{q}$ is the closest monotone rule to RLHF pooling, and we provide explicit bounds on the degradation of the reverse-KL objective. This yields a deployable mechanism for 2026-era sponsored AI answers that remains close to alignment-motivated aggregation while restoring robust incentive properties and auditability.

Table of Contents

3. 3. RLHF-Motivated Aggregation and the Failure of Robust Monotonicity: define q^{geo} as reverse-KL minimizer; give minimal counterexample and diagnosis (normalization-induced nonmonotonicity).

4. 4. Monotone Paths and the Projection Problem: define the convex set $\mathcal{M}(p_1, p_2)$; define continuous and discretized projection operators; uniqueness and structural properties (no-overshoot, coordinate-wise monotonicity).

5. 5. Efficient Computation: discretize λ; formulate as convex program with isotonic constraints; provide algorithms (PAV/isotonic subroutines + simplex constraints). Flag when numerical methods are needed and give complexity and convergence guarantees.

6. 6. Incentive-Compatible Influence Pricing: prove $\widehat{q}$ is robustly monotone; import stable sampling existence and define second-price-style payments; provide payment identity and interpretation.

7. 7. Approximation Guarantees Relative to RLHF Objective: bound the increase in $\sum_i b_i D_{\mathrm{KL}}(q\|p_i)$ and/or in per-token reverse-KL loss; characterize cases where projection is exact (no monotonicity violations).

8. 8. Demonstrations: implement for real LLM distributions (prompt-tuned agents); show smooth transitions, fewer pathologies than raw log-linear pooling, and payment interpretability.

9. 9. Extensions and Limitations: >2 advertisers, alternative partial orders (semantic attributes), and sequence-level issues; open problems for universal auditability and low-query payments.

10. 10. Conclusion.

1 Introduction

Large language models increasingly sit at the point of sale. A user asks for a restaurant recommendation, a travel itinerary, or a business-software comparison, and the system produces a fluent, contextual answer that may embed or imply commercial choices. Sponsored search already allocates scarce attention across advertisers; generative answers allocate something subtler: *probabilistic influence* over which entities, attributes, and framings appear in a stochastic text-generation process. This paper studies how to sell such influence in a way that is simultaneously (i) aligned with modern machine-learning aggregation practices and (ii) compatible with incentive constraints that arise when payments depend on bids.

Our starting point is pragmatic. Platforms have strong reasons to operationalize advertising not as hard constraints (*always mention advertiser 1*) but as *soft steering* that preserves relevance, safety, and user trust. In a token-by-token generator, the most granular steering object is the next-token distribution. An advertiser can be viewed as providing a distribution over next tokens—either literally, via a conditional model head, or implicitly, via a scoring rule over candidate continuations. The platform then needs an *aggregator* that maps advertiser-conditioned next-token distributions and bids into a single distribution from which the system samples the next token. This is the mechanism-design analogue of a standard ML problem: how to combine two probabilistic "experts" into one prediction.

The ML literature offers a compelling benchmark: log-linear (geometric) pooling. In the two-source case it takes the form

$$q_t^{\mathrm{geo}}(\lambda) \;\propto\; p_{1,t}^{\lambda}\, p_{2,t}^{1-\lambda},$$

where the bid ratio $\lambda \in [0,1]$ plays the role of a mixing weight. This rule is attractive for at least three reasons. First, it is a strict convex optimum of a natural regularized objective: it minimizes a weighted sum of KL divergences to the two sources (a property closely related to the variational formulations that motivate RLHF-style updates). Second, it is symmetric and scale-free: only relative bids matter. Third, it offers a clean interpretation for engineers: moving λ increases the log-odds of tokens favored by advertiser 1 relative to advertiser 2.

Yet the mechanism-design problem is not merely to choose a plausible aggregator; it is to choose one that supports payments and strategic bidding. In a conventional single-parameter auction, the allocation must be monotone in bids to admit truthful (critical-bid) payments. When the "allocation" is a distribution over tokens, monotonicity becomes a multidimensional statement. Recent work by Dütting et al. **?** formalizes a robust preference model under which advertisers care about shifting probability mass in the "right direction" across outcomes, and shows that a suitable *robust monotonicity* condition is sufficient to implement payments via stable sampling and

second-price-style critical bids. In our setting, the relevant monotonicity is coordinate-wise: as advertiser 1's bid weight rises, every token that advertiser 1 likes more than advertiser 2 should weakly increase in probability, and every token it likes less should weakly decrease.

This is where an important ML–incentives mismatch emerges. Geometric pooling is *monotone before normalization*—each unnormalized score moves in the correct direction as λ changes. But after normalization, the probabilities become coupled through the partition function, and this coupling can produce reversals: a token probability can move up for small λ and then move down even though advertiser 1 continues to gain weight. Economically, the problem is that increasing a bidder's weight can alter the competitive landscape among *other* tokens so much that an individual coordinate moves counter to the bidder's preference. Mechanism-theoretically, such reversals break monotonicity and therefore undermine stable sampling and critical-bid pricing. Operationally, they create two risks that platforms should care about. First, *pricing risk*: without monotonicity, payments become implementation-dependent or undefined, inviting manipulation and making revenue unpredictable. Second, *policy risk*: non-monotone influence is hard to explain to advertisers, auditors, and regulators, because a higher bid can sometimes reduce the probability of a desirable mention.

We view this mismatch as a design opportunity rather than a negative result. The geometric pooler is a compelling benchmark because it is close to what a platform would choose if it only cared about predictive aggregation or RLHF-like objectives. The question is then: can we retain its virtues while enforcing the monotonicity structure needed for a well-behaved auction?

Our answer is to separate *benchmarking* from *implementing*. We first compute the RLHF-motivated benchmark path $\lambda \mapsto q^{\mathrm{geo}}(\lambda)$. We then compute the *closest* path (in a divergence sense) that satisfies robust monotonicity constraints across λ. Conceptually, we are projecting a desirable but potentially non-monotone ML rule onto the set of auction-compatible rules. This projection has three appealing features. (i) It provides a transparent interpretation: among all monotone influence rules, we choose the one that distorts the ML benchmark the least. (ii) It is robust: the monotonicity constraints enforce "no overshoot" in every coordinate, which aligns with the robust order used to justify incentive properties under coarse advertiser preferences. (iii) It is computationally tractable: after discretizing λ, the projection becomes a convex program with linear constraints, and for squared loss it reduces to isotonic-regression-style primitives.

The economic content of the projection is subtle. If one thinks of λ as a price-weighted social planner parameter, geometric pooling is an "unconstrained optimum" that trades off fit to advertiser 1 and advertiser 2. Our projection imposes an additional implementability constraint: *as bids change, the mapping from bids to allocations must move in a single direction for each advertiser in a robust sense.* The resulting distortion can be

interpreted as the shadow cost of incentive compatibility in a probabilistic allocation environment. This interpretation lets us ask policy-relevant questions: When does the incentive constraint bind severely? How does the distortion scale with advertiser disagreement? Can smoothing or temperature reduce the distortion? These comparative statics matter for platform governance because they translate into practical levers (e.g., regularization, discretization granularity) that trade off revenue, user experience, and auditability.

Contributions. We make four contributions.

First, we isolate a concrete failure mode of log-linear pooling in influence auctions: normalization-induced non-monotonicity. Importantly, this is not a pathological preference assumption; it arises even under the weakest robust preference model where advertisers only care about moving probability mass in the obvious coordinate-wise direction.

Second, we propose a monotone projection framework. We define a convex set of robust-monotone paths connecting the endpoint distributions and project the geometric pooler onto this set using a strictly convex divergence. The strict convexity ensures uniqueness, which is valuable both analytically (comparative statics are well-defined) and operationally (the platform can commit to a single deterministic allocation rule).

Third, we connect this projected aggregator to implementable payments. Because the projected rule is robustly monotone, we can invoke the stable-sampling framework of **?** to obtain second-price-style critical-bid payments per token, together with an expected payment identity that does not depend on the internal sampling implementation. In other words, the projection is not merely a fix to an aesthetic monotonicity violation; it is precisely what enables canonical auction payments in a stochastic-generation setting.

Fourth, we provide computational guidance and welfare bounds. On a discretized grid in λ, the projection problem admits fast solvers based on isotonic regression and simplex projections (for squared loss), and remains convex under KL with mirror-descent-type methods. We also bound the loss in the RLHF-style objective relative to geometric pooling in terms of the projection distance, clarifying when implementability is "cheap" and when it is inherently costly.

Scope and limitations. Our model is intentionally per-token. This is not because sequences are unimportant, but because the platform's real-time decision is local: at each prefix it must choose a next-token distribution. Sequence-level outcomes then arise from repeated application of a one-step mechanism. This approach makes the implementability constraints crisp and aligns with how modern decoders operate, but it abstracts from intertemporal strategic effects (e.g., advertisers valuing early mentions more than later

ones) and from global constraints (e.g., budget pacing over a session). We also focus on two advertisers to sharpen the geometry of monotone paths; extending to many advertisers is conceptually feasible but introduces higher-dimensional partial orders and more complex constraints. Finally, our framework presumes the platform can query advertiser-conditioned distributions (or equivalent scores). This is realistic in some architectures (e.g., mixture-of-experts or prompt-conditioned heads) but not universal, and raises governance questions about what it means for an advertiser to "submit" a distribution. We treat these as design parameters rather than solved problems.

Roadmap. Section 2 formalizes the one-step influence-auction environment, reviews robust preferences and robust monotonicity, and explains why monotone allocation is the central implementability condition. Subsequent sections develop the monotone projection rule, analyze its properties, and derive payment and computational implications. Throughout, we emphasize the tradeoff illuminated by the model: geometric pooling is a natural ML benchmark, but implementable influence requires a monotone correction whose size is governed by disagreement, smoothing, and discretization.

2 Preliminaries and One-Step Model

We model a single generation step as an allocation problem over a finite outcome space. Fix a finite token (or action) set T with $|T| = m$. At a given prefix, advertiser $i \in \{1, 2\}$ is associated with a next-token distribution $p_i \in \Delta(T)$. We treat (p_1, p_2) as primitives of the step and suppress the dependence on the prefix to keep notation light. The platform (designer) must map bids and these distributions into a single distribution over tokens from which the next token is sampled. Our goal in this section is to formalize (i) the per-step mechanism, (ii) the preference structure under which monotone probabilistic allocations are meaningful, and (iii) the monotonicity condition that is sufficient for canonical (critical-bid) pricing via stable sampling, following Dütting et al. **?**.

2.1 Token-level mechanism and bid parameterization

Each step proceeds as follows. Advertisers submit scalar bids $b_1, b_2 \in \mathbb{R}_+$.[1] The platform computes the bid ratio

$$\lambda \;:=\; \frac{b_1}{b_1 + b_2} \;\in [0, 1],$$

[1] Allowing $b_i = 0$ is straightforward, but keeping $b_i > 0$ avoids corner cases in the bid ratio. In implementation, a reserve or minimum bid plays the same role.

and then applies an *aggregation rule* $q(\cdot)$ that maps λ (and, implicitly, (p_1, p_2)) to a distribution over tokens:

$$q(\lambda) \in \Delta(T), \qquad \text{and the platform samples } t \sim q(\lambda).$$

We emphasize two modeling choices. First, because only the ratio λ matters for the allocation, this is a *scale-free* two-bidder environment: multiplying both bids by the same factor does not change the distribution from which the token is drawn. This mirrors practical "share-of-voice" implementations and anticipates the ML benchmark in the next section. Second, the allocation is inherently stochastic: the platform chooses a *distribution* and then samples a realized token. Thus the auctioned object is probabilistic influence rather than a deterministic slot.

Payments are assessed after the token realization. We write the per-step payment charged to advertiser i as $\zeta_i(b_1, b_2; p_1, p_2, t)$ and the (quasi-linear) per-step utility as

$$U_i(b_i, b_{-i}; p_1, p_2) = u_i\big(q(\lambda); p_i\big) - z_i(b_1, b_2; p_1, p_2),$$

where z_i is the expected payment induced by ζ_i and the sampling rule. The precise construction of (ζ_i, z_i) is deferred; the key point for now is that payment implementability in this stochastic environment hinges on an appropriate monotonicity property of the allocation rule $q(\cdot)$.

2.2 Robust preferences over token distributions

The central difficulty is that advertisers may not have a fully specified cardinal utility over all distributions $q \in \Delta(T)$. In influence applications, what is often credible is only a directional statement: advertiser 1 prefers distributions that put more probability on tokens it "likes" (relative to advertiser 2), and less probability on tokens it "dislikes." Dütting et al. **?** formalize this via a *robust preference order* that is weak enough to be behaviorally plausible yet strong enough to support incentive guarantees.

In our two-advertiser environment, the robust structure is conveniently represented token-by-token. Define, for each token $t \in T$, the direction sign

$$\mathrm{sgn}_t := \mathrm{sign}(p_{1,t} - p_{2,t}) \in \{-1, 0, +1\}.$$

Intuitively, $\mathrm{sgn}_t = +1$ means token t is relatively more desired by advertiser 1 than by advertiser 2 (because advertiser 1 assigns it higher probability in its preferred distribution), while $\mathrm{sgn}_t = -1$ means the opposite. Tokens with $\mathrm{sgn}_t = 0$ are "common ground" tokens for which the two distributions agree.

We interpret advertiser 1's robust ranking as rewarding movements in the sgn direction starting from the baseline p_2 (the allocation when $\lambda = 0$), and advertiser 2's robust ranking symmetrically starting from baseline p_1 (the allocation when $\lambda = 1$). Concretely, we will use the following monotone,

coordinate-wise sufficient condition: for advertiser 1, distribution q is (robustly) at least as good as q' if it moves probability in the correct directions relative to the endpoint pair (p_2, p_1):

$$q \succeq_1 q' \quad \text{whenever} \quad \operatorname{sgn}_t (q_t - q'_t) \;\geq\; 0 \;\; \text{for all } t \in T,$$

and similarly, for advertiser 2,

$$q \succeq_2 q' \quad \text{whenever} \quad \operatorname{sgn}_t (q_t - q'_t) \;\leq\; 0 \;\; \text{for all } t \in T.$$

This is a deliberately coarse order: it ignores fine substitutions among tokens that are all "good" for a bidder, and it does not require knowing how much a bidder values any particular token. Economically, it captures the idea that advertisers purchase influence to increase (or decrease) certain token probabilities but may be unable to articulate a complete utility function over the entire simplex.[2]

We assume advertisers' utilities are monotone with respect to these robust orders: if $q \succeq_i q'$ then $u_i(q; p_i) \geq u_i(q'; p_i)$. When we later want explicit welfare or approximation bounds, we will also consider tractable parametric forms, such as

$$u_i(q; p_i) \;=\; -v_i \, D_{\mathrm{KL}}(q \| p_i) \qquad \text{or} \qquad u_i(q; p_i) \;=\; -v_i \, \|q - p_i\|_1,$$

for $v_i > 0$. These specifications are not meant to be literal advertiser preferences; rather, they let us connect the auction rule to ML-style divergence objectives and quantify distortion from a benchmark.

2.3 Monotone aggregation as implementability

Given robust preferences, we now state the implementability-relevant monotonicity condition. An aggregation rule $q(\cdot)$ is *robustly monotone* if, as advertiser 1's bid weight increases (i.e., as λ increases), the induced distribution becomes weakly better for advertiser 1 in the robust order, and symmetrically as λ decreases it becomes weakly better for advertiser 2. In the two-advertiser token specialization, a simple sufficient condition is coordinate-wise monotonicity with the correct direction:

$$\forall t \in T, \; \forall \lambda \geq \lambda' : \qquad \operatorname{sgn}_t \big(q_t(\lambda) - q_t(\lambda') \big) \;\geq\; 0. \tag{1}$$

Condition (1) has three immediate implications that are useful later. First, it rules out "reversals": the probability of a token favored by advertiser 1 cannot rise and then fall as advertiser 1 bids more. Second, it enforces a

[2]The formal robust order in **?** is stated for general outcome spaces and is designed to be compatible with stable sampling. In the two-advertiser token setting, the coordinate-wise order above is a convenient specialization that matches the monotonicity constraints we will impose in our projection rule.

"no-overshoot" property: if we also impose boundary conditions $q(0) = p_2$ and $q(1) = p_1$, then for each token t we necessarily have

$$q_t(\lambda) \ \in \ \big[\min\{p_{1,t}, p_{2,t}\},\ \max\{p_{1,t}, p_{2,t}\}\big] \qquad \text{for all } \lambda \in [0,1],$$

which will later align with our projection geometry. Third, it turns the allocation rule into a one-dimensional monotone object (indexed by λ) despite the high-dimensional simplex: this is exactly the structure that supports critical-bid reasoning.

The link to pricing comes from the stable sampling framework of **?**. At a high level, stable sampling is a way to implement a monotone mapping from bids to *random outcomes* such that each realized outcome admits a threshold (critical) bid characterization. In standard deterministic single-parameter auctions, monotonicity of allocation implies the existence of payments that make truthful bidding a dominant strategy. Here the allocation is a distribution over tokens, but the same economic logic survives: if the probability of every "good" token moves monotonically with the bid, then one can couple the random draws across bids so that, for each token realization, there is a well-defined minimum bid at which that token would still have been sampled. Charging that minimum bid yields a second-price-style payment per realization.

We do not re-prove the stable sampling results; instead we adopt them as a mechanism-design primitive once robust monotonicity is verified. The practical takeaway is that (1) is not merely an aesthetic regularity. It is what makes payments *canonical*: expected payments can be expressed via an integral (Myerson-like) identity that does not depend on implementation details of the sampling routine, which is valuable for auditability and platform governance.

2.4 Per-token model as a repeated mechanism

Finally, we clarify how this one-step model relates to full-text generation. A generated sequence is obtained by repeating the one-step mechanism at successive prefixes, each time with prefix-dependent distributions (p_1, p_2) and possibly prefix-dependent bids or pacing constraints. Our analysis is intentionally local: at each step, the platform must choose a next-token distribution and can charge per-token payments. This is the natural granularity at which modern decoders operate and the granularity at which the influence auction can be embedded into a larger system.

The limitation is that sequence-level objectives may not be additively separable across tokens (e.g., an advertiser values the first mention more than subsequent ones), and sequence-level constraints (e.g., budgets, safety filters) can introduce intertemporal linkages. We view the one-step analysis as the correct starting point for mechanism design in probabilistic generation because it isolates the implementability constraint in its sharpest form. The

next section then studies the most salient benchmark aggregator from the ML side and explains why it can violate (1) even in simple cases, motivating our projected, implementable alternative.

3 RLHF-Motivated Aggregation and the Failure of Robust Monotonicity

A natural benchmark for aggregating two next-token distributions is the log-linear (geometric) pooling rule familiar from RLHF-style objectives and, more generally, from weighted reverse-KL projection. In our setting, this benchmark is appealing for two reasons. First, it is *locally* optimal for a canonical divergence objective that interpolates between p_1 and p_2 as the bid weight λ varies. Second, it is easy to compute: one can form unnormalized scores token-by-token and then renormalize. The catch, which is central for mechanism design, is that this final normalization step couples coordinates and can break the robust monotonicity condition (1) needed for stable sampling and critical-bid payments.

3.1 Geometric pooling as a reverse-KL optimum

Fix $(p_1, p_2) \in \Delta(T)^2$ and a weight $\lambda \in [0, 1]$. Consider the weighted reverse-KL objective

$$L_{\mathrm{RLHF}}(q; \lambda) \;:=\; \lambda D_{\mathrm{KL}}(q\|p_1) \;+\; (1-\lambda) D_{\mathrm{KL}}(q\|p_2), \qquad q \in \Delta(T). \tag{2}$$

This objective is a convenient stylization of "stay close to each advertiser's preferred next-token distribution, with weights proportional to bids." Because the KL divergence is strictly convex in its first argument, (2) has a unique minimizer on the simplex.

The resulting rule is the log-linear pooling distribution

$$q_t^{\mathrm{geo}}(\lambda) \;\propto\; p_{1,t}^{\lambda}\, p_{2,t}^{1-\lambda}, \qquad \text{equivalently} \qquad q_t^{\mathrm{geo}}(\lambda) \;=\; \frac{p_{1,t}^{\lambda} p_{2,t}^{1-\lambda}}{Z(\lambda)}, \tag{3}$$

where $Z(\lambda) := \sum_{s \in T} p_{1,s}^{\lambda} p_{2,s}^{1-\lambda}$.

The derivation is standard but worth recording because it clarifies what is monotone (and what is not). Writing the Lagrangian for (2) with multiplier η for the simplex constraint yields

$$\mathcal{L}(q, \eta) \;=\; \sum_{t \in T} q_t \Big(\lambda \log \frac{q_t}{p_{1,t}} + (1-\lambda) \log \frac{q_t}{p_{2,t}} \Big) \;+\; \eta \Big(\sum_{t \in T} q_t - 1 \Big).$$

The first-order condition for each coordinate is

$$\frac{\partial \mathcal{L}}{\partial q_t} \;=\; \lambda \Big(\log q_t - \log p_{1,t} + 1 \Big) \;+\; (1-\lambda) \Big(\log q_t - \log p_{2,t} + 1 \Big) \;+\; \eta \;=\; 0,$$

which simplifies to

$$\log q_t \;=\; \lambda \log p_{1,t} \;+\; (1-\lambda)\log p_{2,t} \;+\; C(\lambda),$$

for a constant $C(\lambda)$ chosen to enforce $\sum_t q_t = 1$. Exponentiating and normalizing gives (3). In particular, the benchmark rule satisfies $q^{\text{geo}}(0) = p_2$ and $q^{\text{geo}}(1) = p_1$.

From an ML perspective, (3) is also the distribution obtained by adding log-probabilities with weights λ and $(1-\lambda)$ and then applying a softmax. This makes q^{geo} a plausible "first try" for a platform that wants a smooth, scale-free mapping from bids to token probabilities.

3.2 Why normalization can break robust monotonicity

The robust monotonicity condition (1) is coordinate-wise: for a token t with $\text{sgn}_t = +1$ we require $q_t(\lambda)$ to be nondecreasing in λ, and for $\text{sgn}_t = -1$ we require it to be nonincreasing. At first glance, geometric pooling seems to satisfy exactly this property, because its *unnormalized* scores move monotonically.

To see this, define the unnormalized score

$$r_t(\lambda) \;:=\; p_{1,t}^{\lambda} p_{2,t}^{1-\lambda} \;=\; p_{2,t}\exp\big(\lambda \log(p_{1,t}/p_{2,t})\big).$$

Then $r_t(\lambda)$ is log-linear in λ and is monotone: r_t increases in λ iff $p_{1,t} > p_{2,t}$, decreases iff $p_{1,t} < p_{2,t}$, and stays constant iff $p_{1,t} = p_{2,t}$. In other words, each coordinate moves in the "right direction" before normalization.

The difficulty is that the platform samples from the *normalized* distribution $q^{\text{geo}}(\lambda) = r(\lambda)/Z(\lambda)$, and the normalizer $Z(\lambda) = \sum_s r_s(\lambda)$ changes with λ in a way that depends on *all* tokens. This is precisely the cross-token coupling that a mechanism designer must be wary of: even if every coordinate's raw score moves monotonically, a coordinate's *share* can reverse direction as other shares expand or contract.

A convenient diagnostic follows from differentiating the normalized form. Using $q_t = r_t/Z$ and $r_t'/r_t = \log(p_{1,t}/p_{2,t})$, we obtain

$$\frac{d}{d\lambda} q_t^{\text{geo}}(\lambda) \;=\; q_t^{\text{geo}}(\lambda)\left(\log\frac{p_{1,t}}{p_{2,t}} \;-\; \sum_{s\in T} q_s^{\text{geo}}(\lambda)\log\frac{p_{1,s}}{p_{2,s}}\right). \tag{4}$$

Thus the sign of $\frac{d}{d\lambda} q_t^{\text{geo}}(\lambda)$ is not determined solely by whether $p_{1,t} > p_{2,t}$, but by whether token t's log-likelihood ratio is above or below the *current average* log-likelihood ratio under $q^{\text{geo}}(\lambda)$. As λ changes, the distribution $q^{\text{geo}}(\lambda)$ shifts mass toward tokens with large $\log(p_{1,s}/p_{2,s})$, raising the average term in (4). A token can therefore start by increasing (when it is above the average) and later decrease (once the average rises past it), even if it is favored by advertiser 1 at the endpoints.

This is exactly the kind of "reversal" ruled out by (1). And because stable sampling hinges on monotonicity in the bid parameter, a reversal is not a benign modeling artifact: it destroys the single-crossing structure needed for critical-bid payments.

3.3 A minimal three-token counterexample

We now give a concrete instance with $|T| = 3$ where q^{geo} violates (1). Let $T = \{a, b, c\}$ and set

$$p_2 \;=\; (0.40,\ 0.30,\ 0.30), \qquad p_1 \;=\; (0.41,\ 0.58,\ 0.01),$$

where coordinates are ordered as (a, b, c). Then $\text{sgn}_a = +1$ because $p_{1,a} > p_{2,a}$.

At $\lambda = 0$ and $\lambda = 1$, geometric pooling coincides with the endpoints:

$$q^{\text{geo}}(0) \;=\; p_2, \qquad q^{\text{geo}}(1) \;=\; p_1.$$

At the midpoint $\lambda = \frac{1}{2}$, we compute unnormalized scores via geometric means:

$$r_a\left(\tfrac{1}{2}\right) = \sqrt{0.40 \cdot 0.41} \approx 0.4050, \quad r_b\left(\tfrac{1}{2}\right) = \sqrt{0.30 \cdot 0.58} \approx 0.4171, \quad r_c\left(\tfrac{1}{2}\right) = \sqrt{0.30 \cdot 0.01} \approx 0.0548,$$

so $Z(\frac{1}{2}) \approx 0.8769$ and hence

$$q^{\text{geo}}\left(\tfrac{1}{2}\right) \;\approx\; (0.462,\ 0.475,\ 0.062).$$

Focus on token a. Since $\text{sgn}_a = +1$, robust monotonicity would require $q_a^{\text{geo}}(\lambda)$ to be nondecreasing in λ. But we have

$$q_a^{\text{geo}}(0) \;=\; 0.40, \qquad q_a^{\text{geo}}\left(\tfrac{1}{2}\right) \;\approx\; 0.462, \qquad q_a^{\text{geo}}(1) \;=\; 0.41.$$

Because $q_a^{\text{geo}}(\frac{1}{2}) > q_a^{\text{geo}}(1)$, the coordinate trajectory must eventually decrease on $[\frac{1}{2}, 1]$. That is, there exist $\lambda > \lambda'$ such that $q_a^{\text{geo}}(\lambda) < q_a^{\text{geo}}(\lambda')$, contradicting (1) for token a.

The economic content of this example is straightforward. Token a is only *slightly* more favored by advertiser 1 than advertiser 2 (its endpoint probability rises from 0.40 to 0.41), whereas token b is *much* more favored by advertiser 1 (rising from 0.30 to 0.58). As λ increases, the normalizer $Z(\lambda)$ becomes increasingly dominated by token b's rapidly growing score. Token a initially benefits from the shift toward p_1 (hence it rises above 0.40), but later loses share to token b and must come back down to match the endpoint $p_{1,a} = 0.41$. The reversal is entirely due to normalization: the raw score $r_a(\lambda)$ is increasing in λ throughout, yet the normalized probability $q_a^{\text{geo}}(\lambda)$ is not.

This three-token construction is not an edge case. Equation (4) shows that reversals occur whenever there exist at least two tokens favored by advertiser 1 with sufficiently different likelihood ratios. In that sense, nonmonotonicity is generic whenever advertisers disagree in more than a one-dimensional way: the simplex constraint forces some coordinates to "give back" probability as other coordinates grow, and geometric pooling does not control which coordinates do so.

3.4 Implication: we need an implementable correction

The lesson for mechanism design is that the RLHF-motivated benchmark q^{geo} is generally *not* implementable with stable sampling under robust preferences, because it can violate the monotonicity condition that underwrites critical bids and second-price-style payments. This creates a tension between an ML-natural aggregation objective (2) and the monotonicity needed for canonical pricing and auditability.

In the next section, we resolve this tension by treating $q^{\mathrm{geo}}(\cdot)$ as a desirable but potentially nonmonotone *target path* and projecting it onto the convex set of robust-monotone paths connecting (p_2, p_1). The resulting projected rule preserves implementability by construction while remaining, in a precise divergence sense, as close as possible to the RLHF benchmark.

4 Monotone Paths and the Projection Problem

We now formalize the "correction" foreshadowed above. The guiding idea is simple: rather than insisting that the platform implement the RLHF-motivated target $q^{\mathrm{geo}}(\cdot)$ exactly, we treat $q^{\mathrm{geo}}(\cdot)$ as a reference path and select the *closest* path that satisfies the robust monotonicity constraints needed for stable sampling and critical-bid payments. This section pins down (i) the feasible set of robust-monotone paths connecting the endpoints (p_2, p_1) and (ii) the projection operator that maps an arbitrary target path into this feasible set.

4.1 Robust-monotone paths as a convex feasibility set

Fix a finite token set T and endpoints $p_1, p_2 \in \Delta(T)$. For each token $t \in T$, recall the direction sign

$$\mathrm{sgn}_t \;:=\; \mathrm{sign}(p_{1,t} - p_{2,t}) \;\in\; \{-1, 0, +1\},$$

which encodes whether robust monotonicity requires t's probability to weakly increase, weakly decrease, or remain constant as the bid weight λ shifts toward advertiser 1.

We define the set of *robust-monotone paths* as

$$\mathcal{M}(p_1, p_2) := \Big\{ q : [0,1] \to \Delta(T) \;\Big|\; q(0) = p_2,\ q(1) = p_1, \text{ and } \forall t \in T,\ \forall \lambda \geq \lambda' :\ \mathrm{sgn}_t\big(q_t(\lambda) - q_t(\lambda')\big) \geq 0 \tag{5}$$

The monotonicity requirement in (5) is coordinate-wise and directional. If $\mathrm{sgn}_t = +1$, then $\lambda \mapsto q_t(\lambda)$ must be nondecreasing; if $\mathrm{sgn}_t = -1$, it must be nonincreasing; and if $\mathrm{sgn}_t = 0$, it must be constant (hence $q_t(\lambda) \equiv p_{1,t} = p_{2,t}$). We impose no further regularity beyond measurability sufficient to make the objective below well-defined; in particular, the path may have flat regions and kinks, which is important both conceptually (mass can "stick" at constraints) and computationally (the discretized problem becomes isotonic regression).

A key structural implication of (5) is a *no-overshoot* property: every feasible path stays within the coordinate-wise envelope of the endpoints. Indeed, for any $\lambda \in [0,1]$ and $t \in T$,

$$q_t(\lambda) \ \in\ \big[\min\{p_{1,t}, p_{2,t}\},\ \max\{p_{1,t}, p_{2,t}\}\big]. \tag{6}$$

This follows immediately from monotonicity plus the boundary conditions. Economically, (6) captures the notion that increasing bidder 1's weight cannot "over-reward" bidder 1 on any token beyond what bidder 1 would obtain at $\lambda = 1$, nor can it reduce bidder 1 below the baseline $\lambda = 0$ on tokens that bidder 1 prefers.

The set $\mathcal{M}(p_1, p_2)$ is convex in the natural pointwise sense. If $q, q' \in \mathcal{M}(p_1, p_2)$ and $\theta \in [0,1]$, then the path $\widetilde{q}(\lambda) := \theta q(\lambda) + (1-\theta) q'(\lambda)$ satisfies $\widetilde{q}(\lambda) \in \Delta(T)$ for all λ and inherits both the boundary conditions and the coordinate-wise monotonicity constraints. This convexity is not merely technical: it ensures that, when we search for a closest feasible path, we are performing a well-behaved convex projection rather than selecting among multiple local minima.

4.2 A divergence-based projection onto monotonicity

Let $q^{\mathrm{geo}} : [0,1] \to \Delta(T)$ be a given target path (in our application, geometric pooling). We define the *monotone projection* of q^{geo} onto $\mathcal{M}(p_1, p_2)$ as the solution to

$$\widehat{q} \ \in\ \arg\min_{q \in \mathcal{M}(p_1, p_2)} \int_0^1 D\big(q(\lambda) \| q^{\mathrm{geo}}(\lambda)\big)\, d\lambda, \tag{7}$$

where $D(\cdot\|\cdot)$ is a divergence on the simplex that is strictly convex in its first argument (e.g., D_{KL} or $\frac{1}{2}\|\cdot\|_2^2$). The objective in (7) makes explicit the tradeoff we want the model to illuminate: we preserve implementability by restricting to $\mathcal{M}(p_1, p_2)$, and within that implementable set we remain as faithful as possible (in divergence) to the ML-natural benchmark.

Two remarks help interpret (7).

First, (7) is a *pathwise* correction rather than a pointwise one. We do not independently project each $q^{\mathrm{geo}}(\lambda)$ onto a monotone set at that same λ; rather, we choose an entire path that is jointly consistent across λ. This is precisely what stable sampling needs: a coherent mapping $\lambda \mapsto \widehat{q}(\lambda)$ that respects monotonicity globally, not merely at isolated points.

Second, the choice of divergence D is a modeling decision with practical consequences. KL divergence aligns with probabilistic geometry and yields a correction that is sensitive to relative errors in low-probability tokens, whereas squared ℓ_2 divergence emphasizes absolute errors and often admits faster projection routines. Our subsequent mechanism-theoretic conclusions hinge on feasibility and monotonicity, not on the specific D; D primarily affects how the correction distributes distortion across tokens and across λ.

Under mild conditions, (7) has a unique solution. Intuitively, $\mathcal{M}(p_1, p_2)$ is convex, the integrand is convex in $q(\lambda)$, and strict convexity of $D(\cdot\|r)$ implies strict convexity of the integral functional in the path argument. In a discretized setting (below), uniqueness follows from standard results for strictly convex minimization over a nonempty compact polytope. In continuous time, one can either work in an L^1 space with weak compactness arguments or (as we do for implementation) treat the continuous problem as the limit of a sequence of discretizations.

A practical caveat concerns zero probabilities. If we use KL, then $D_{\mathrm{KL}}(q\|r)$ is finite only when $\mathrm{supp}(q) \subseteq \mathrm{supp}(r)$. To avoid degenerate cases where some $q_t^{\mathrm{geo}}(\lambda) = 0$ forces $q_t(\lambda) = 0$ (potentially conflicting with endpoints), we either assume that p_1 and p_2 are strictly positive on T (as is typical after smoothing in language models) or we work with an ϵ-truncated simplex and renormalization. This is a technical assumption, but it matches deployment realities: large language models rarely assign exact zeros after standard temperature and top-p smoothing.

4.3 Discretization and the finite-dimensional projection program

For computation and for a clean connection to isotonic regression, we discretize $\lambda \in [0, 1]$ on a grid

$$\Lambda \;:=\; \{\lambda_j\}_{j=0}^{J}, \qquad 0 = \lambda_0 < \lambda_1 < \cdots < \lambda_J = 1,$$

and write $q_j := q(\lambda_j)$ and $q_j^{\mathrm{geo}} := q^{\mathrm{geo}}(\lambda_j)$. We also choose quadrature weights $w_j > 0$ to approximate the integral (uniform weights $w_j \propto \lambda_{j+1} - \lambda_j$ suffice for a Riemann sum).

The discretized feasible set is the set of arrays $\{q_j\}_{j=0}^{J}$ satisfying:

$$\forall t \in T,\ \forall j \in \{0,\dots,J-1\}: \quad \operatorname{sgn}_t\left(q_{j+1,t} - q_{j,t}\right) \geq 0, \tag{8}$$

$$\forall j \in \{0,\dots,J\}: \quad \sum_{t\in T} q_{j,t} = 1, \quad q_{j,t} \geq 0, \tag{9}$$

$$q_0 = p_2, \quad q_J = p_1. \tag{10}$$

Condition (8) is exactly the discrete analogue of coordinate-wise monotonicity; (9) enforces that each q_j is a distribution; and (10) anchors the endpoints. When hard endpoints are undesirable (for instance, to reduce distortion near $\lambda = 0$ or $\lambda = 1$), one can relax (10) by adding penalties such as $\rho D(q_0\|p_2) + \rho D(q_J\|p_1)$ for large ρ, but we keep the hard constraint to preserve a crisp "winner-takes-all" interpretation at extreme bids.

The discretized projection problem is then

$$\{\widehat{q}_j\}_{j=0}^{J} \ \in\ \arg\min_{\{q_j\}} \sum_{j=0}^{J} w_j\, D\big(q_j\|q_j^{\mathrm{geo}}\big) \qquad \text{s.t. (8)–(10).} \tag{11}$$

Because the constraints (8)–(10) are linear and $D(\cdot\|r)$ is convex in its first argument, (11) is a finite-dimensional convex program. Moreover, if $D(\cdot\|r)$ is strictly convex in its first argument (as with KL on the interior, or squared ℓ_2 everywhere), then the objective is strictly convex and the optimizer is unique. This uniqueness is operationally valuable: it means the platform's aggregation rule is well-defined and does not depend on solver tie-breaking, which in turn simplifies auditing and reproducibility.

Finally, the discretized constraints preserve the structural properties we rely on later. From (8) and the endpoints (10), we again obtain a discrete no-overshoot guarantee:

$$\forall t,\ \forall j: \quad \widehat{q}_{j,t} \in \big[\min\{p_{1,t},p_{2,t}\}, \max\{p_{1,t},p_{2,t}\}\big].$$

Thus the projection cannot create new extremes; it only removes reversals by flattening coordinates as needed. In economic terms, whenever q^{geo} would "ask" a token to rise and then fall (or vice versa), the projection replaces that behavior by pooling adjacent λ-intervals into constant segments that respect the required direction while remaining as close as possible to the target in divergence.

The discretized formulation (11) is therefore our main object for implementation and for the subsequent analysis of computation and payments. Conceptually, it should be viewed as a principled regularization of RLHF-style aggregation: we keep the same benchmark but enforce the monotone comparative statics in bids that mechanism design needs.

4.4 Efficient computation: isotonic structure, alternating projections, and practical complexity

The discretized program (11) is convex, but naïvely solving it as a generic constrained optimization problem would be far too slow to run at every generation step. The key computational observation is that the constraints decompose into two simple families with complementary structure: the monotonicity constraints (8) couple grid points j within each token coordinate t, while the simplex constraints (9) couple tokens t within each grid point j. This "bipartite" structure suggests solvers based on repeated projections (or proximal steps) onto these two families, each of which is efficiently computable.

To make this concrete, it is useful to define two closed convex sets in the product space $\left(\mathbb{R}^{|T|}\right)^{J+1}$:

$$C_{\text{iso}} := \Big\{\{q_j\} \,:\, (8) \text{ holds and } q_0 = p_2,\, q_J = p_1\Big\}, \qquad C_\Delta := \Big\{\{q_j\} \,:\, (9) \text{ holds}\Big\}.$$

Then (11) is the problem of finding, among arrays $\{q_j\} \in C_{\text{iso}} \cap C_\Delta$, the closest point to the reference $\{q_j^{\text{geo}}\}$ in the geometry induced by D. When D is squared Euclidean distance, this is literally an orthogonal projection; when D is KL, it is a Bregman projection. In both cases, the intersection structure admits fast iterative methods with strong convergence guarantees.

Squared ℓ_2 divergence: Euclidean projections via isotonic regression and simplex projection. Suppose $D(q\|r) = \frac{1}{2}\|q - r\|_2^2$. Then (11) is equivalent to the weighted least-squares problem

$$\min_{\{q_j\} \in C_{\text{iso}} \cap C_\Delta} \; \frac{1}{2} \sum_{j=0}^{J} w_j \|q_j - q_j^{\text{geo}}\|_2^2,$$

which can be viewed as an Euclidean projection under the inner product $\langle x, y\rangle = \sum_j w_j x_j^\top y_j$. A standard approach for such problems is Dykstra's algorithm (alternating projections with correction terms) applied to the pair of convex sets $(C_{\text{iso}}, C_\Delta)$. Operationally, each Dykstra iteration consists of two projection subroutines:

1. *Projection onto C_{iso} (token-wise isotonic regression across λ).* Fix a token t. The constraints (8) restrict the sequence $(q_{0,t}, \ldots, q_{J,t})$ to be monotone in the direction prescribed by sgn_t, with fixed endpoints $q_{0,t} = p_{2,t}$ and $q_{J,t} = p_{1,t}$. The Euclidean projection of an arbitrary sequence $(y_{0,t}, \ldots, y_{J,t})$ onto this set is exactly a *weighted isotonic regression* problem in one dimension:

$$\min_{x_0,\ldots,x_J} \; \frac{1}{2} \sum_{j=0}^{J} w_j (x_j - y_{j,t})^2 \quad \text{s.t.} \quad \text{sgn}_t(x_{j+1} - x_j) \geq 0,\; x_0 = p_{2,t},\; x_J = p_{1,t}.$$

This can be solved in $\mathcal{O}(J)$ time by the pool-adjacent-violators (PAV) algorithm (after reversing indices when $\mathrm{sgn}_t = -1$; when $\mathrm{sgn}_t = 0$ the solution is constant and trivial). Importantly, this projection is independent across tokens, so it parallelizes perfectly over $t \in T$.

2. *Projection onto C_Δ (simplex projection at each λ_j).* Fix a grid point j. Given a vector $y_j \in \mathbb{R}^{|T|}$, the projection onto the simplex $\Delta(T) = \{q_j : \sum_t q_{j,t} = 1,\ q_{j,t} \geq 0\}$ is

$$\Pi_\Delta(y_j) \;=\; \arg\min_{q_j \in \Delta(T)} \; \frac{1}{2}\|q_j - y_j\|_2^2,$$

which is computable by thresholding: $q_{j,t} = \max\{y_{j,t} - \tau, 0\}$ for a scalar τ chosen so that $\sum_t q_{j,t} = 1$. This can be implemented in $\mathcal{O}(|T| \log |T|)$ time via sorting, or in expected linear time using selection algorithms; and, again, it parallelizes over j.

Because both C_{iso} and C_Δ are closed convex (indeed polyhedral) sets, Dykstra's algorithm converges to the unique optimizer of (11) in the squared-loss case. Convergence holds in objective value and in iterates; moreover, when the intersection is regular (which is typical away from degenerate probability vectors), one obtains linear convergence rates characterized by standard error-bound/Hoffman constants for polyhedral feasibility problems. In deployment we do not rely on worst-case iteration bounds; rather, the practical point is that each iteration is cheap and the number of iterations required for high accuracy is usually modest.

Complexity accounting and the "near-linear" regime. Let $m := |T|$. One Dykstra iteration costs

$$\mathcal{O}(mJ) \quad \text{for token-wise PAV} \qquad + \qquad \mathcal{O}\big(J \cdot \mathrm{SimplexProj}(m)\big) \quad \text{for simplex projections.}$$

With sorting-based simplex projection, this is $\mathcal{O}(mJ + Jm \log m)$ per iteration. Two remarks explain why this is still compatible with per-token generation:

(i) In language-model implementations, the effective token set is typically truncated (top-k, nucleus sampling, or a sparse support induced by logits), so m is the post-truncation vocabulary size rather than the full vocabulary. The projection is exact on the restricted support provided the same support is used consistently across p_1, p_2 and q^{geo} (or one works on the union support and assigns a small ϵ mass elsewhere).

(ii) Both subroutines are embarrassingly parallel: PAV runs independently for each token, and simplex projection runs independently for each grid point. This is a good fit for GPU/TPU-style parallelism, and it also admits caching and warm starts across adjacent generation steps (prefixes change gradually, so $\widehat{q}$ changes smoothly in practice).

Finally, the discretization size J is a direct latency–accuracy knob. Because the monotonicity constraints are enforced only on the grid, the stable-sampling implementation will typically use either piecewise-constant or monotone interpolation between grid points; finer grids reduce approximation error but increase runtime linearly.

KL divergence: Bregman geometry and when we need numerical methods. When $D = D_{\mathrm{KL}}$, the program (11) remains convex and uniquely solvable on the interior of the simplex, but the Euclidean projection machinery above no longer applies verbatim. Two routes remain practical.

First, one can use *Bregman alternating projections* (a KL-analogue of Dykstra) onto C_{iso} and C_{Δ}. The projection onto C_{Δ} under KL is particularly simple: minimizing $D_{\mathrm{KL}}(q_j \| y_j)$ subject to $q_j \in \Delta(T)$ amounts to renormalizing a nonnegative vector (after incorporating the algorithm's dual corrections), i.e.,

$$q_{j,t} \;\propto\; y_{j,t} \quad \text{with} \quad \sum_t q_{j,t} = 1.$$

The projection onto C_{iso} decomposes across tokens into one-dimensional convex problems of the form

$$\min_{x_0,\dots,x_J} \sum_{j=0}^{J} w_j\, x_j \log \frac{x_j}{y_{j,t}} \quad \text{s.t.} \quad \mathrm{sgn}_t(x_{j+1} - x_j) \geq 0,\; x_0 = p_{2,t},\; x_J = p_{1,t},$$

which is a Bregman-isotonic regression problem (sometimes called isotonic regression under I-divergence). There exist generalized PAV-type algorithms for such separable Bregman objectives; alternatively, one can solve each token subproblem by a fast 1D convex solver because the constraint set is simple and the objective is strictly convex.

Second, one can bypass explicit Bregman projections and run a *mirror-descent / dual-ascent* method for (11), exploiting the fact that the feasible set is defined by linear inequalities and equalities. A convenient implementation pattern is to maintain dual variables for the monotonicity constraints and normalize at each λ_j to satisfy the simplex constraints. These methods come with standard convergence guarantees for convex problems: with appropriate step sizes, the objective gap decays at $\mathcal{O}(1/k)$ for subgradient-style updates and faster rates are available under additional smoothness/strong-convexity conditions on a truncated simplex $\{q : q_{j,t} \geq \epsilon\}$.

The practical message is the following. With squared loss, we obtain a particularly clean and fast primitive (PAV + simplex projection) with textbook convergence. With KL, we retain convexity and uniqueness but typically rely on iterative Bregman/mirror methods; the per-iteration cost remains near-linear in mJ, but the iteration count is more sensitive to conditioning (e.g., how close probabilities are to 0). This sensitivity is not merely

mathematical: it is precisely why smoothing (temperature, ϵ-truncation) is often beneficial not only for modeling, but also for numerical stability.

Implementation details that matter for mechanisms. Two final details are worth flagging because they interact with the downstream incentive analysis.

First, we must enforce the boundary conditions reliably. In the squared-loss case, we can hard-fix $q_0 = p_2$ and $q_J = p_1$ by removing those variables from the optimization (or by giving them infinite weight in the isotonic step). In KL-based solvers, the same is true, but one must ensure compatibility with support: if $p_{1,t} > 0$ but some intermediate reference value $q_{j,t}^{\text{geo}}$ is numerically 0, the KL objective becomes ill-conditioned. In practice we prevent this by lower-bounding all inputs by ϵ and renormalizing.

Second, stable sampling and critical-bid payments will later require consistent evaluation of $\widehat{q}(\lambda)$ as λ varies. In a grid-based implementation, the platform can (i) compute $\{\widehat{q}_j\}_{j=0}^{J}$ once per prefix, (ii) serve $\widehat{q}(\lambda)$ by monotone interpolation between adjacent grid points, and (iii) use the same discretization when searching for critical bid thresholds. This keeps the computational representation aligned with the monotonicity certificate: the solver enforces (8), and the interpolation preserves it by construction.

Taken together, these algorithmic facts justify treating $\widehat{q}$ as an operational primitive. The projection is not merely an existence result; it can be computed fast enough to sit inside a per-token auction loop, and it comes with convergence guarantees that support reproducibility and auditing. We next use the resulting robust monotonicity to import stable sampling and define second-price-style influence payments.

4.5 Incentive-Compatible Influence Pricing: monotone aggregation, stable sampling, and second-price-style payments

We now connect the projected aggregator $\widehat{q}$ to incentives. The computational section established that $\widehat{q}$ is an operational primitive: given (p_1, p_2) and bids (b_1, b_2) we can evaluate $\widehat{q}(\lambda)$ quickly on a grid. The economic content of the projection is that it restores a strong monotonicity property that geometric pooling can violate. Once this monotonicity is in place, we can directly import the "stable sampling" implementation and critical-bid pricing results of Dütting et al. (2024), yielding a clean second-price-style influence auction at every generation step.

Robust monotonicity of the projected rule. Fix a prefix and hence fixed $p_1, p_2 \in \Delta(T)$. Recall $\text{sgn}_t = \text{sign}(p_{1,t} - p_{2,t})$. In the discretized

program, feasibility requires

$$\forall t \in T,\ \forall j \in \{0, \dots, J-1\}: \qquad \operatorname{sgn}_t\left(q_{j+1,t} - q_{j,t}\right) \geq 0,$$

together with the boundary conditions $q_0 = p_2$ and $q_J = p_1$. Two immediate consequences are worth making explicit.

First, *no coordinate reversals and no overshoot*: for every t and every grid point j,

$$q_{j,t} \in \left[\min\{p_{1,t}, p_{2,t}\},\ \max\{p_{1,t}, p_{2,t}\}\right].$$

This is simply because each coordinate is constrained to move monotonically from its value at λ_0 to its value at λ_J.

Second, the projected path is *monotone in the robust partial orders* relevant for influence auctions. Specializing the robust order of Dütting et al. to two advertisers with baseline p_2 for advertiser 1, we can write

$$q \succeq_1 q' \quad \Longleftrightarrow \quad \begin{cases} q_t \geq q'_t & \text{for all } t \text{ with } p_{1,t} \geq p_{2,t}, \\ q_t \leq q'_t & \text{for all } t \text{ with } p_{1,t} \leq p_{2,t}. \end{cases}$$

(When $p_{1,t} = p_{2,t}$ the coordinate is irrelevant.) By construction of $\widehat{q}$ we have, for any $\lambda \geq \lambda'$ on the grid (and, with monotone interpolation, for all $\lambda \geq \lambda'$),

$$\widehat{q}(\lambda) \succeq_1 \widehat{q}(\lambda').$$

Symmetrically, taking advertiser 2's baseline as $q(1) = p_1$, we obtain $\lambda \leq \lambda' \Rightarrow \widehat{q}(\lambda) \succeq_2 \widehat{q}(\lambda')$. Economically, increasing λ (which increases advertiser 1's bid weight) can only move probability mass in directions that advertiser 1 robustly prefers, and never in the opposite direction.

From distribution monotonicity to single-parameter monotonicity. To apply standard dominant-strategy arguments, we reduce the distribution-valued outcome to a one-dimensional "amount of influence" that is monotone in a bidder's bid holding the other fixed. In our setting this reduction is canonical: the robust order is coordinate-wise, and the total amount of movement away from baseline is exactly a total-variation distance along the constrained directions. Define advertiser 1's influence level at weight λ by

$$x_1(\lambda) \ := \sum_{t:\ p_{1,t} > p_{2,t}} \left(\widehat{q}_t(\lambda) - p_{2,t}\right) \ = \ \frac{1}{2}\left\|\widehat{q}(\lambda) - p_2\right\|_1,$$

where the last equality uses that $\sum_t \widehat{q}_t(\lambda) = \sum_t p_{2,t} = 1$ and the sign pattern partitions positive and negative deviations. Under robust monotonicity, $x_1(\lambda)$ is nondecreasing in λ and satisfies $x_1(0) = 0$ and $x_1(1) = \frac{1}{2}\|p_1 - p_2\|_1$. Because $\lambda(b) = \frac{b_1}{b_1 + b_2}$ is strictly increasing in b_1 for fixed $b_2 > 0$, the composed allocation

$$x_1(b_1, b_2) \ := \ x_1(\lambda(b_1, b_2))$$

is nondecreasing in b_1. An analogous definition holds for advertiser 2 relative to baseline p_1:

$$x_2(\lambda) \;:=\; \frac{1}{2}\big\|\widehat{q}(\lambda) - p_1\big\|_1,$$

which is nonincreasing in λ and hence nondecreasing in b_2 holding b_1 fixed.

This reduction is not merely a proof trick: it matches the mechanism interpretation. The quantity x_i measures how much probability mass the mechanism has shifted in bidder i's preferred directions relative to the appropriate baseline. Under robust preferences, any such directional movement is weakly beneficial, regardless of how the bidder values individual tokens beyond their direction of change.

Stable sampling: implementing a monotone distribution rule with per-outcome thresholds. The remaining step is to implement $\widehat{q}(\lambda)$ in a way that supports per-token critical bids. Sampling $t \sim \widehat{q}(\lambda)$ naïvely does not by itself define a meaningful "allocation" to each advertiser, hence does not directly yield a second-price payment rule.

Stable sampling, as developed by Dütting et al. (2024), addresses exactly this issue. Given a monotone mapping from bids to distributions (monotone in the robust order), stable sampling constructs a coupling of outcomes across bids using shared randomness. Informally, we draw a single random seed and use it to generate not only the realized token t, but also an *attribution* of that token to (at most) one advertiser, such that:

1. the marginal distribution of the realized token is exactly $\widehat{q}(\lambda(b))$;
2. each advertiser's probability of receiving attribution is exactly their influence level $x_i(b)$;
3. the attribution rule is *stable*: if bidder i increases b_i (holding b_{-i} and the random seed fixed), then the event "bidder i is attributed the token" can only switch from false to true, never in the opposite direction.

Stability is the key property that makes per-realization critical bids well defined: for a fixed seed and a fixed realization, there exists a threshold bid at which attribution flips.

In our application, the existence of stable sampling follows immediately from the monotonicity of $\widehat{q}$ established above. We emphasize that the projection step is doing real economic work: geometric pooling can fail monotonicity, which can in turn break stable sampling and thereby break dominant-strategy pricing. The monotone projection repairs this failure at the source.

Second-price-style (critical bid) payments. With stable sampling in hand, we define payments by the usual critical-value logic for single-parameter environments. Fix advertiser i, fix the other bid b_{-i}, and condition on the

random seed used by the stable sampling routine. Let $\mathbf{1}\{i \text{ attributed}\}$ denote whether advertiser i is attributed the realized token under bids (b_i, b_{-i}). Stability implies that, as a function of b_i, this indicator is nondecreasing pointwise (for each fixed seed). Therefore there exists a *critical bid* $\beta_i = \beta_i(b_{-i}; \text{seed})$ such that advertiser i is attributed the token if and only if $b_i \geq \beta_i$. The per-step payment is then

$$\zeta_i(b_1, b_2; p_1, p_2, t) \;:=\; \begin{cases} \beta_i(b_{-i}; \text{seed}) & \text{if advertiser } i \text{ is attributed,} \\ 0 & \text{otherwise.} \end{cases}$$

This is "second-price-style" in the precise sense that the winner (the advertiser who is attributed the influence on that step) pays the minimal bid needed to retain that attribution given the competitor's bid and the realized randomness. Importantly, although the randomness enters the critical threshold, the *expected* payment admits an implementation-independent characterization.

Payment identity (envelope formula) and interpretation. For monotone single-parameter mechanisms, dominant-strategy incentive compatibility together with (interim) individual rationality pins down payments up to an additive constant. In our setting the relevant allocation is $x_i(b)$, the probability of being attributed influence. The standard envelope formula yields the expected payment

$$z_i(b_i, b_{-i}) \;=\; b_i\, x_i(b_i, b_{-i}) \;-\; \int_0^{b_i} x_i(s, b_{-i})\, ds, \tag{12}$$

with the normalization $z_i(0, b_{-i}) = 0$ (formally obtained by continuity as $b_i \downarrow 0$). Using $x_1(\lambda) = \frac{1}{2}\|\widehat{q}(\lambda) - p_2\|_1$, we can rewrite (12) as "pay for marginal movement in total variation away from baseline." That is, advertiser 1 pays according to the area under the curve mapping her bid to the total mass shifted in her preferred directions. The same interpretation holds for advertiser 2 relative to baseline p_1.

Two remarks clarify how to read (12) in practice. First, although bidders ultimately care about which token is generated, the mechanism prices the *ability to steer probability mass* rather than charging per token identity in an ad hoc way; this aligns with robust preferences, which only require monotone directional improvements. Second, the identity shows that the monetary transfers depend on $\widehat{q}$ only through the one-dimensional influence curve $x_i(\cdot, b_{-i})$, which is helpful for transparency and auditing: one can log and report how much total probability mass each bidder was able to move at each bid level, and payments are determined by that reportable object.

Scope and limitations. Our incentive claims are per-step: for each prefix, truthful bidding is a dominant strategy given robust (monotone) preferences

and quasi-linear utility. Extending to full-sequence generation is conceptually a repeated mechanism; if advertisers submit a fixed bid for the entire generation, the mechanism remains dominant-strategy truthful period-by-period, and total payments are additive across steps. What we do *not* claim is incentive compatibility for arbitrary non-robust utilities that depend on higher-order interactions among tokens, nor do we address strategic manipulation through misreporting of p_i (we take p_i as queryable primitives). In applied deployments these limitations matter, but they are also precisely why the robust-order framework is attractive: it isolates a minimal monotonicity structure under which we can give clean, mechanism-theoretic guarantees.

The remaining question is efficiency: how much do we lose, relative to the RLHF-motivated geometric pooling benchmark, by projecting onto the monotone set? We turn to this approximation question next.

4.6 Approximation guarantees relative to the RLHF objective

The projection step that defines $\widehat{q}$ is motivated by incentives, not by raw fit to the RLHF-inspired pooling rule. It is therefore natural to ask an efficiency question: when we replace $q^{\mathrm{geo}}(\lambda)$ by its monotone projection $\widehat{q}(\lambda)$, how much do we distort the standard reverse-KL objective that underlies log-linear pooling? The key point is that we are not optimizing an unrelated criterion. Rather, $\widehat{q}$ is the *closest* robust-monotone path to q^{geo} under a convex divergence, so any welfare loss relative to q^{geo} can be controlled by the projection distance itself. This makes the monotonicity repair interpretable as a "small regularization" of RLHF pooling whose magnitude is auditable.

Per-step RLHF loss and its minimizer. Fix a prefix and hence $p_1, p_2 \in \Delta(T)$ and $\lambda \in [0,1]$. Define the (per-step) RLHF loss

$$L_{\mathrm{RLHF}}(q;\lambda) \;:=\; \lambda D_{\mathrm{KL}}(q\|p_1) \;+\; (1-\lambda) D_{\mathrm{KL}}(q\|p_2),$$

so that the bid-weighted objective appearing in a welfare calculation is simply

$$\sum_{i=1}^{2} b_i \, D_{\mathrm{KL}}(q\|p_i) \;=\; (b_1+b_2)\, L_{\mathrm{RLHF}}(q;\lambda).$$

By the reverse-KL optimality property recalled earlier, $q^{\mathrm{geo}}(\lambda)$ is the unique minimizer of $L_{\mathrm{RLHF}}(\cdot;\lambda)$ over $\Delta(T)$. Therefore, for any alternative rule (in particular $q = \widehat{q}(\lambda)$), the efficiency loss at weight λ is exactly the objective gap

$$\Delta_{\mathrm{RLHF}}(\lambda) \;:=\; L_{\mathrm{RLHF}}(\widehat{q}(\lambda);\lambda) \;-\; L_{\mathrm{RLHF}}(q^{\mathrm{geo}}(\lambda);\lambda) \;\geq\; 0.$$

Our goal is to bound $\Delta_{\mathrm{RLHF}}(\lambda)$, and also its average over λ, in terms of the projection distance between $\widehat{q}$ and q^{geo}.

A smoothness-based bound: loss gap controlled by proximity to q^{geo}. The function $q \mapsto L_{\mathrm{RLHF}}(q;\lambda)$ is convex and differentiable on the simplex interior. Moreover, its curvature is governed by the entropy term and hence depends on how close q is to the boundary. To make this dependence explicit, let $\Delta_\epsilon := \{q \in \Delta(T) : q_t \geq \epsilon \ \forall t\}$ for some $\epsilon \in (0, 1/|T|]$. On Δ_ϵ, the Hessian of $D_{\mathrm{KL}}(q\|p)$ with respect to q is $\nabla_q^2 D_{\mathrm{KL}}(q\|p) = \mathrm{diag}(1/q_t)$, and hence

$$\nabla_q^2 L_{\mathrm{RLHF}}(q;\lambda) \;=\; \mathrm{diag}(1/q_t) \;\preceq\; \frac{1}{\epsilon} I \qquad \text{for all } q \in \Delta_\epsilon,$$

independently of λ and of p_1, p_2. Consequently, $L_{\mathrm{RLHF}}(\cdot;\lambda)$ is $(1/\epsilon)$-smooth on Δ_ϵ with respect to $\|\cdot\|_2$. Applying the standard smoothness inequality with $x = q^{\mathrm{geo}}(\lambda)$ and $y = \widehat{q}(\lambda)$ yields

$$L_{\mathrm{RLHF}}(y;\lambda) \;\leq\; L_{\mathrm{RLHF}}(x;\lambda) \;+\; \langle \nabla L_{\mathrm{RLHF}}(x;\lambda),\, y - x\rangle \;+\; \frac{1}{2\epsilon}\|y-x\|_2^2. \tag{13}$$

Because x minimizes $L_{\mathrm{RLHF}}(\cdot;\lambda)$ over the simplex, its KKT conditions imply that $\nabla L_{\mathrm{RLHF}}(x;\lambda)$ is a constant vector (a multiple of the all-ones vector) on the support of x, so the inner product term vanishes for any feasible perturbation $y - x$ with $\sum_t (y_t - x_t) = 0$. Thus, whenever both $q^{\mathrm{geo}}(\lambda)$ and $\widehat{q}(\lambda)$ lie in Δ_ϵ, we obtain the concrete pointwise bound

$$\Delta_{\mathrm{RLHF}}(\lambda) \;\leq\; \frac{1}{2\epsilon}\left\|\widehat{q}(\lambda) - q^{\mathrm{geo}}(\lambda)\right\|_2^2. \tag{14}$$

This inequality formalizes the intuition that the projection cannot be too costly unless it moves the distribution substantially.

Relating the gap to the projection divergence. Our projection program measures closeness using a divergence $D(\cdot\|\cdot)$, typically either squared ℓ_2 or KL. When $D(q\|r) = \frac{1}{2}\|q - r\|_2^2$, (14) reads simply as

$$\Delta_{\mathrm{RLHF}}(\lambda) \;\leq\; \frac{1}{\epsilon}\, D(\widehat{q}(\lambda) \,\|\, q^{\mathrm{geo}}(\lambda))\,.$$

When $D(\cdot\|\cdot)$ is KL, we can combine (14) with a norm–divergence comparison. For example, by Pinsker's inequality,

$$\left\|\widehat{q}(\lambda) - q^{\mathrm{geo}}(\lambda)\right\|_1^2 \;\leq\; 2\, D_{\mathrm{KL}}(\widehat{q}(\lambda) \,\|\, q^{\mathrm{geo}}(\lambda))\,, \qquad \|\cdot\|_2^2 \leq \|\cdot\|_1^2,$$

so that

$$\Delta_{\mathrm{RLHF}}(\lambda) \;\leq\; \frac{1}{\epsilon}\, D_{\mathrm{KL}}(\widehat{q}(\lambda) \,\|\, q^{\mathrm{geo}}(\lambda))\,. \tag{15}$$

The dependence on ϵ is not an artifact: reverse-KL losses become arbitrarily steep near the boundary of the simplex. Practically, this is a reminder that some degree of smoothing (temperature, flooring, or support restriction) is not only numerically helpful but also strengthens approximation guarantees.

Pathwise (integrated) welfare loss bound. The preceding bounds are pointwise in λ. For deployment and for ex ante analysis it is often more relevant to control the *average* or *total* loss across bid weights, since the mechanism will face a distribution of bid ratios over time. Integrating (14) over λ gives

$$\int_0^1 \Delta_{\mathrm{RLHF}}(\lambda)\, d\lambda \;\le\; \frac{1}{2\epsilon} \int_0^1 \left\| \widehat{q}(\lambda) - q^{\mathrm{geo}}(\lambda) \right\|_2^2 d\lambda,$$

and similarly under a discretization $\Lambda = \{\lambda_j\}_{j=0}^J$ with weights $\{w_j\}$,

$$\sum_{j=0}^{J} w_j\, \Delta_{\mathrm{RLHF}}(\lambda_j) \;\le\; \frac{1}{2\epsilon} \sum_{j=0}^{J} w_j \left\| \widehat{q}_j - q^{\mathrm{geo}}(\lambda_j) \right\|_2^2.$$

In words: the *same* integral that our projection program minimizes (up to constants and the choice of divergence) upper bounds the induced RLHF objective deterioration. Thus the approximation quality is not a separate emergent phenomenon; it is directly linked to the optimization criterion used to enforce monotonicity.

A coarse but transparent benchmark via a feasible monotone path. The preceding inequalities express the RLHF gap in terms of the projection distance actually achieved. Sometimes one would like an *a priori* upper bound that does not require solving the projection. A simple way to obtain such a bound is to exhibit any feasible monotone path $q^{\mathrm{feas}} \in \mathcal{M}(p_1, p_2)$ and use optimality of $\widehat{q}$:

$$\int_0^1 D(\widehat{q}(\lambda) \,\|\, q^{\mathrm{geo}}(\lambda))\; d\lambda \;\le\; \int_0^1 D\Big(q^{\mathrm{feas}}(\lambda) \,\Big\|\, q^{\mathrm{geo}}(\lambda)\Big)\; d\lambda.$$

A canonical choice is the linear interpolation $q^{\mathrm{lin}}(\lambda) := (1-\lambda)p_2 + \lambda p_1$, which is coordinate-wise monotone in the correct directions and satisfies the boundary conditions. While q^{lin} is typically not close to q^{geo} in high-disagreement cases, it provides a simple, auditable certificate: the projection cannot be worse (in the chosen divergence) than this explicit baseline, and hence its RLHF objective gap is controlled by the divergence between q^{lin} and q^{geo}.

When is the projection exact? The projection is exact (i.e., $\widehat{q}(\lambda) = q^{\mathrm{geo}}(\lambda)$ for all λ) whenever the geometric pooling path already lies in $\mathcal{M}(p_1, p_2)$, equivalently whenever it satisfies the coordinate-wise monotonicity constraints. This happens in a few important special cases that clarify the economic meaning of the pathology and when it should be rare.

First, if $|T| = 2$, normalization cannot induce a reversal. Indeed, with two tokens $t \in \{a, b\}$, the path is one-dimensional because $q_a(\lambda) = 1 - q_b(\lambda)$, and

$q_a^{\text{geo}}(\lambda)$ is a logistic function of λ (a ratio of two exponentials), hence monotone. Thus nonmonotonicity is inherently a 3-or-more-token phenomenon.

Second, if the *likelihood ratios* $p_{1,t}/p_{2,t}$ take only two values—one common value for all tokens with $p_{1,t} > p_{2,t}$ and another common value for all tokens with $p_{1,t} < p_{2,t}$—then the geometric pooling path moves mass uniformly within each sign group, and each coordinate moves monotonically in the correct direction. This is a stylized case, but it captures a practical intuition: reversals arise when there is meaningful heterogeneity *within* the set of tokens that a bidder "likes" (or "dislikes"), because normalization then forces competition among same-direction tokens.

Third, even when exactness fails globally, it often holds approximately when distributions are close or smoothed. As disagreement $\|p_1 - p_2\|_1$ shrinks, the normalizer $Z(\lambda)$ varies little and $q^{\text{geo}}(\lambda)$ becomes nearly affine in λ in a local chart, so monotonicity violations diminish and the projection distance (hence the RLHF gap) becomes small. Likewise, temperature smoothing of geometric pooling flattens probabilities and dampens the sensitivity of $Z(\lambda)$, empirically making violations rarer; in our bounds, smoothing also effectively increases the relevant interior parameter ϵ.

Interpretation and design implications. From a mechanism-design perspective, these approximation results formalize a tradeoff we care about in practice. The geometric pooling rule is the exact minimizer of a familiar RLHF surrogate, but it can fail the monotonicity needed for clean incentive and pricing guarantees. The projected rule restores monotonicity, and the price we pay is controlled by an explicit and logged object: the divergence between $\widehat{q}(\lambda)$ and $q^{\text{geo}}(\lambda)$ along the bid-weight path. This is attractive for governance: a platform can report (per prefix, or aggregated over time) the total "monotonicity correction" applied, and auditors can assess whether incentive compatibility is being achieved with negligible distortion of the RLHF benchmark or whether the environment exhibits systematic high-disagreement regimes where the correction is nontrivial.

Limitations of the bound. Two caveats are worth highlighting. First, constants deteriorate as $\epsilon \downarrow 0$, reflecting the fact that reverse-KL becomes extremely sensitive when some tokens have vanishing probability under the realized distributions. In large-vocabulary language models this is a genuine issue, so any implementation should incorporate standard numerical safeguards (flooring, top-k/top-p truncation, or temperature) if one wants uniform guarantees. Second, our bounds are local in the sense that they depend on distance to q^{geo}; in worst cases, if monotonicity constraints force large changes, the RLHF gap can be economically meaningful. This is not a contradiction but rather a diagnosis: large gaps indicate that the baseline pooling objective and the incentive constraints are in sharp conflict at that

prefix.

With these guarantees in hand, we can move from theory to evidence. In the next section we implement the projected rule on real next-token distributions from prompt-tuned agents, illustrating when projection is essentially exact, when it meaningfully repairs geometric-pooling pathologies, and how the induced payments inherit a transparent "pay for total variation movement" interpretation.

4.7 Demonstrations: projected pooling on real next-token distributions

We now turn from the preceding guarantees to a concrete question: when the "advertisers" are instantiated as prompt-tuned language-model agents, does the projected rule $\widehat{q}$ behave like a gentle correction of log-linear pooling, or does it routinely induce large and economically meaningful distortions? Our goal in this section is not to optimize a benchmark model, but to document the qualitative and quantitative shape of the monotonicity repair on realistic next-token distributions, and to illustrate how the induced payments inherit an intuitive "pay for marginal movement" interpretation.

Experimental instantiation of advertisers. We instantiate the two advertisers by two prompt-tuned variants of a common base LLM. Concretely, at each prefix we query two agents that differ only in their system prompt (or, in a separate set of runs, two lightweight adapters fine-tuned for distinct stylistic objectives). Each query returns logits over the vocabulary; we convert logits to a probability distribution via a fixed temperature and, for numerical stability, apply a standard truncation (e.g., top-k or nucleus sampling) followed by renormalization and a small probability floor so that $q \in \Delta_\epsilon$ holds with a controlled ϵ. This is not merely an engineering convenience: our approximation bounds in Section 4.6 are meaningful precisely when the realized distributions are not arbitrarily close to the simplex boundary.

Operationally, we treat p_1 and p_2 as the two advertiser-conditioned next-token distributions at the current prefix. Given bids (b_1, b_2) we compute $\lambda = b_1/(b_1 + b_2)$ and then evaluate the geometric pooling path

$$q_t^{\mathrm{geo}}(\lambda) \;=\; \frac{p_{1,t}^{\lambda} p_{2,t}^{1-\lambda}}{\sum_{s \in T} p_{1,s}^{\lambda} p_{2,s}^{1-\lambda}}.$$

We then compute $\widehat{q}(\lambda)$ by solving the discretized projection program **(E5)** on a grid $\Lambda = \{\lambda_j\}_{j=0}^{J}$.

What we measure: monotonicity pathologies and projection magnitude. To make "pathology" concrete, we record (i) the fraction of token–grid constraints violating **(E2)** for the raw pooling path q^{geo} (after fixing

directions via sgn_t), (ii) the magnitude of the worst reversal per prefix (e.g., $\max_{t,j} -\mathrm{sgn}_t(q^{\mathrm{geo}}_{j+1,t} - q^{\mathrm{geo}}_{j,t})$), and (iii) the overall repair size, summarized by the objective value of the projection,

$$\mathcal{R} \;:=\; \sum_{j=0}^{J} w_j \, D(\widehat{q}_j \,\|\, q^{\mathrm{geo}}(\lambda_j))\,,$$

which is directly auditable because it is exactly what the platform computes. Finally, for context, we also report the implied RLHF objective gap $\sum_j w_j \Delta_{\mathrm{RLHF}}(\lambda_j)$ and compare it to $\mathcal{R}$ in light of inequalities such as (15).

Computation: isotonic regression behaves like a "local" correction. In the squared-loss implementation $D(q\|r) = \frac{1}{2}\|q - r\|_2^2$, the projection decomposes into an intersection of two simple convex sets: per-token monotone constraints across λ and per-λ simplex constraints across tokens. We implement the solve by alternating (a) per-token weighted isotonic regression over the sequence $\{q_{j,t}\}_{j=0}^{J}$ using PAV with direction determined by sgn_t, and (b) per-λ Euclidean projection onto the simplex. In our runs, this procedure converges rapidly and has near-linear scaling in $|T| \cdot J$ in the regime relevant for deployment, where T is a truncated candidate set (e.g., top-k tokens plus a shared "other" bucket).

A helpful practical intuition is that the projection is typically *sparse in where it acts*: for most tokens and most λ-intervals, $q^{\mathrm{geo}}_{j,t}$ already moves in the correct direction, and isotonic regression leaves those coordinates unchanged. Corrections concentrate on the relatively small set of tokens whose trajectories exhibit a reversal, and even there the correction tends to "pool" adjacent λ values into flat segments (the familiar geometry of isotonic solutions). This is exactly the behavior one would want from an incentives-motivated repair: it acts only where needed to restore monotonicity, and otherwise preserves the RLHF-inspired pooling rule.

Qualitative evidence: smooth transitions and repaired reversals. Plotting token trajectories makes the normalization pathology visually salient. For a fixed prefix and a handful of salient tokens, $r_t(\lambda) = p_{1,t}^{\lambda} p_{2,t}^{1-\lambda}$ moves monotonically in λ whenever $p_{1,t} \neq p_{2,t}$, but the normalized probability $q^{\mathrm{geo}}_t(\lambda)$ can exhibit a non-monotone "bump" when several same-direction tokens compete through the normalizer. Empirically, we see exactly this pattern: reversals are most common among medium-probability tokens in clusters of near-substitutes (e.g., variants of punctuation, capitalization, or semantically similar continuations) where small changes in the denominator $Z(\lambda)$ reallocate mass in unintuitive ways.

After projection, these bumps disappear by construction. More importantly, the repaired paths $\lambda \mapsto \widehat{q}_t(\lambda)$ tend to remain smooth at the level of

aggregate behavior: while isotonic regression creates piecewise-constant segments in λ for individual coordinates, the overall distribution $\widehat{q}(\lambda)$ changes gradually because different tokens become active at different points. In the language of practice, moving bids from bidder 2 toward bidder 1 yields a stable and predictable increase in bidder 1-aligned mass, without the counterintuitive "give-and-take" reversals that break robust monotonicity.

How frequent are violations in realistic prompts? Across a broad set of prefixes sampled from held-out text (and, separately, from interactive assistant transcripts), we find that violations of **(E2)** under q^{geo} are neither ubiquitous nor vanishingly rare: they tend to occur in bursts on prefixes where the two agents disagree in a multi-modal way. Consistent with the comparative statics above, disagreement measured by $\|p_1 - p_2\|_1$ is a strong predictor of both the violation rate and the repair size $\mathcal{R}$. When p_1 and p_2 are close (e.g., two prompts that differ only slightly), $Z(\lambda)$ varies weakly and q^{geo} is often already feasible, so $\widehat{q} = q^{\text{geo}}$ and the projection is exactly costless.

Conversely, when prompts induce sharply different stylistic modes (e.g., "marketing" versus "encyclopedic" tone, or "creative" versus "terse" completions), the raw geometric path exhibits noticeable non-monotonicity on a modest set of tokens, and $\widehat{q}$ applies a correspondingly modest but nonzero correction. Importantly, even in these high-disagreement regimes, the repair rarely looks like a wholesale replacement of log-linear pooling by linear interpolation; rather, it is a targeted adjustment that enforces the no-overshoot property coordinate by coordinate.

Efficiency impact: the RLHF gap tracks the repair size. We empirically verify the main message of Section 4.6: prefixes with a larger projection distance $\mathcal{R}$ also exhibit a larger RLHF objective gap $\sum_j w_j \Delta_{\text{RLHF}}(\lambda_j)$. This is unsurprising given the smoothness bounds, but it matters for governance because it means the platform can log a single scalar per prefix (the projection objective value) as an ex ante indicator of how much the incentives constraint is "fighting" the RLHF benchmark. In practical terms, the monotonicity repair becomes auditable: if an auditor (or an internal monitoring system) observes that $\mathcal{R}$ is consistently near zero, then the mechanism is essentially implementing RLHF pooling while retaining the incentive properties. If $\mathcal{R}$ is frequently large, that flags either systematic advertiser conflict or a mismatch between the chosen pooling baseline and the robust monotonicity desideratum.

Payments: "pay for marginal influence" is visible at the token level. To illustrate the interpretability of second-price-style payments under stable sampling, we examine realized tokens and compute the associated critical

bid payments. Two robust patterns emerge.

First, payments concentrate on *contested* tokens: when both advertisers put substantial probability mass on a token but with different intensities, small changes in λ materially change its probability under $\widehat{q}(\lambda)$, and the critical bid for swinging the allocation is correspondingly meaningful. By contrast, tokens that both advertisers view similarly (either both like or both dislike) typically have low marginal sensitivity and hence low payments.

Second, aggregated over a continuation, payments align with an "influence budget" interpretation. Because $\widehat{q}$ is monotone, the expected payment identities from stable sampling can be read informally as charging bidders for the total change in the output distribution they induce relative to the baseline. In practice, this makes the mechanism easier to explain: advertisers are not paying for inscrutable internal scores, but for measurable movement in next-token probabilities. This interpretability is especially valuable in an LLM setting, where external stakeholders often care less about the exact mechanism details and more about whether economic influence is being exercised transparently and predictably.

Limitations of the demonstration. We emphasize three caveats. First, our measurement is per-prefix and per-step; while repeated application yields a sequence, sequence-level effects (feedback from earlier sampled tokens into later distributions) can amplify or dampen disagreement in ways that are not captured by a static snapshot. Second, our truncation and flooring choices affect both numerical stability and the constants in the approximation bounds; while these are standard in generation systems, they should be treated as part of the mechanism specification rather than a hidden implementation detail. Third, prompt-tuned agents are only one way to instantiate advertiser preferences; richer preference representations (e.g., semantic attribute distributions) may change the structure of monotonicity violations and the practical meaning of payments.

Taken together, these demonstrations suggest that monotone projection is best viewed as a lightweight enforcement layer: it preserves the qualitative behavior of log-linear pooling when pooling is already well-behaved, while reliably repairing the normalization-induced reversals that matter for incentives and pricing. This sets the stage for the natural extensions and open problems we take up next.

4.8 Extensions and limitations: beyond two advertisers, richer preference orders, and sequence-level incentives

Our construction was deliberately scoped to the two-advertiser, per-token setting because that is where the economic logic is cleanest: bids collapse to a single mixing parameter $\lambda \in [0, 1]$, robust monotonicity reduces to one-dimensional coordinate-wise constraints, and the projection admits fast

isotonic-regression primitives. It is natural, however, to ask how far these ideas extend, and where they do not. We view this section as a map of the main fault lines: what generalizes with relatively minor technical work, what seems to require new conceptual tools, and what open problems matter most for practical deployment.

More than two advertisers: geometry and monotonicity become genuinely multidimensional. With $n > 2$ advertisers, the RLHF-motivated benchmark generalizes immediately: given bids $b \in \mathbb{R}_+^n$ one can define weights

$$\alpha_i(b) \;:=\; \frac{b_i}{\sum_{k=1}^n b_k} \in \Delta([n]),$$

and set the log-linear pooled distribution to be

$$q_t^{\mathrm{geo}}(\alpha) \;\propto\; \prod_{i=1}^n p_{i,t}^{\alpha_i}.$$

As in the two-advertiser case, this $q^{\mathrm{geo}}(\alpha)$ is the unique minimizer of $\sum_i \alpha_i D_{\mathrm{KL}}(q\|p_i)$ over $q \in \Delta(T)$. The difficulty is not the benchmark, but the incentives constraint: robust monotonicity is no longer a one-parameter monotone path problem, but a monotonicity constraint on a mapping from a *bid simplex* (or $\mathbb{R}_+^n$ modulo scale) into $\Delta(T)$.

There are at least three non-equivalent ways one might try to generalize the feasible set $\mathcal{M}(p_1, p_2)$. First, one can impose *coordinate-wise monotonicity in each bidder's bid* holding others fixed, a multi-dimensional isotonic constraint that resembles monotone allocation in multi-parameter mechanism design. Second, one can require monotonicity only along *pairwise* tradeoff directions (e.g., when bidder i increases b_i while bidder k decreases b_k keeping $\sum b$ fixed), which is closer to the two-bidder partial order but may be insufficient for truthful payments with simultaneous competition. Third, one can restrict attention to *one-dimensional slices* through bid space (e.g., varying a single λ along a chosen curve $\alpha(\lambda)$), regaining tractability but at the cost of making the mechanism dependent on an exogenous tie-breaking path through the simplex.

From a computational perspective, the projection program remains convex under natural generalizations (linear monotonicity constraints plus simplex constraints), but the near-linear isotonic structure can disappear. In one dimension, isotonic regression is solved by PAV; in higher dimensions, exact isotonic regression is substantially harder and is typically handled by general-purpose convex optimization or specialized algorithms with weaker guarantees. This is not merely a speed issue: higher-dimensional monotonicity constraints can be *tighter* in ways that increase the projection distortion relative to q^{geo}, sharpening the welfare–incentives tradeoff.

A pragmatic middle ground, which we think is promising but not yet theoretically satisfactory, is to implement an n-bidder mechanism by *iterated pairwise aggregation*: combine (p_1, p_2) into an intermediate distribution using our two-bidder projected rule, then combine that result with p_3, and so on according to a fixed tree. This preserves one-dimensional monotonicity at each merge and yields a simple stable-sampling implementation. The limitation is conceptual: pairwise merging is not associative, so the outcome depends on the merge tree, raising governance questions (who decides the tree?) and strategic questions (can a bidder benefit from being merged earlier or later?). Establishing conditions under which some merge rule is both transparent and incentive-compatible remains open.

Alternative robust partial orders: from tokens to semantic attributes. Our monotonicity constraints were defined at the token level: for each $t \in T$, the direction $\mathrm{sgn}_t = \mathrm{sign}(p_{1,t} - p_{2,t})$ determines whether the path should be nondecreasing or nonincreasing in λ. This is the natural choice if advertisers literally value token probabilities. In many applications, however, advertisers (or regulators) care about *semantic attributes* of text rather than particular tokens: sentiment, toxicity, formality, topicality, or adherence to a style guide. Two distributions that differ substantially at the token level may be nearly identical with respect to such attributes, and conversely small token-level changes can have large attribute impacts.

A clean way to formalize attribute-level preferences is via a feature map $\phi : T \to \mathbb{R}^K$ and the induced expected feature vector

$$\mu(q) \;:=\; \mathbb{E}_{t\sim q}[\phi(t)] \;\in\; \mathbb{R}^K.$$

One can then define robust partial orders on distributions by comparing $\mu(q)$ rather than q itself (or by combining both, e.g., lexicographic or constrained orders). For instance, bidder 1 might prefer higher expected values of some coordinate $\mu_k(q)$ (say, "positivity"), while bidder 2 prefers lower values (say, "verbosity"). The analog of robust monotonicity would require $\lambda \mapsto \mu_k(\widehat{q}(\lambda))$ to move monotonically in the appropriate direction for each relevant attribute.

This generalization is attractive because it can drastically reduce the dimensionality of the constraints: instead of enforcing monotonicity for every token, we enforce monotonicity for a small number of attributes. But it introduces a new subtlety: monotonicity in attribute space does not uniquely determine a token distribution, and the map $q \mapsto \mu(q)$ is typically many-to-one. Consequently, there can be multiple token-level distributions that satisfy the same attribute-level monotonicity, and the choice among them matters for both welfare (relative to q^{geo}) and for downstream language quality (fluency, diversity).

One plausible resolution is a *two-stage projection*: first project the attribute trajectory of q^{geo} onto the monotone set in $\mathbb{R}^K$ (a low-dimensional

isotonic problem), then, for each λ, choose the closest $q(\lambda) \in \Delta(T)$ to $q^{\text{geo}}(\lambda)$ subject to matching the projected attributes (a convex feasibility problem when D is KL or squared ℓ_2 and μ is linear). The limitation is again conceptual: we would be enforcing monotonicity in the attributes we can measure, which may invite *gaming* on unmeasured dimensions. This is an instance of a broader point: robust orders are only as normatively compelling as the preference primitives (tokens or attributes) that define them.

Sequence-level issues: per-token monotonicity does not automatically imply sequence-level incentive guarantees. Our mechanism is implemented per token, but the object of interest is a generated *sequence*. Treating a full generation as repeated one-step mechanisms is a reasonable first approximation, yet it masks two difficulties.

First, the distributions p_i depend on the evolving prefix, which itself is a random function of earlier samples. Even if $\widehat{q}(\lambda)$ is robustly monotone *conditional on a prefix*, the *ex ante* effect of changing bids on the final sequence distribution can be more complex because bids change the state distribution over prefixes. In practice, this can create feedback loops: shifting probability mass early in a continuation can move the model into a region of prefix space where the advertisers disagree more (or less), which can amplify (or dampen) the effect of bids on later tokens.

Second, advertisers may have intrinsically sequence-level utilities (brand safety constraints, narrative coherence, topic prevalence over the entire answer) that are not separable across steps. Our stable-sampling and critical-bid logic is most immediate for myopic or additively separable utilities; with non-separable objectives, truthful bidding at each step is no longer a dominant-strategy statement without additional assumptions about how bidders commit, observe history, and update bids. A fully satisfactory treatment would model the interaction as a dynamic mechanism design problem with state dependence. We suspect that monotone per-step aggregation is still a useful building block—it limits the scope for per-step manipulations and preserves a form of predictability—but it is not, by itself, a complete sequence-level incentive theory.

Universal auditability: what can be logged, verified, and explained? One practical motivation for the projection viewpoint is that it creates an auditable scalar: the realized repair magnitude $\mathcal{R}$ (or its continuous analog) quantifies how much the incentives constraint deviates from the RLHF benchmark. Yet there is a gap between *auditable in principle* and *auditable in deployed systems*. Three obstacles recur.

(i) *Vocabulary truncation:* real systems rarely operate on the full T at each step. Top-k and nucleus sampling alter the effective simplex and can create artificial discontinuities when tokens enter or leave the candidate set.

The usual workaround (an "other" bucket) helps, but formal audit guarantees that are robust to truncation are underdeveloped.

(ii) *Privacy and proprietary constraints:* auditors may not be able to observe p_1, p_2 directly, especially if they encode sensitive prompts or model internals. This raises the possibility of *zero-knowledge* style auditing: can the platform provide a compact certificate that $\widehat{q}$ is the correct projection of q^{geo} under stated parameters, without revealing the full distributions? Designing such certificates for high-dimensional simplex projections is an open problem.

(iii) *Interpretability of the order itself:* even if the projection is computed correctly, stakeholders may disagree with the token-level robust order as a normative constraint. Attribute-level orders help, but then the auditing problem shifts to the validity and stability of the attribute classifiers ϕ, which may themselves be noisy or manipulable.

Low-query payments: making stable sampling feasible at LLM scale. Finally, even if $\widehat{q}$ is monotone and hence implementable with stable sampling, the naive computation of critical bids can be query-intensive: one may need to evaluate allocations under counterfactual bids to find thresholds. In an LLM setting, each query is expensive, and the mechanism is executed at every generation step.

We see two promising directions. The first is to exploit the *structure induced by discretization.* When $\widehat{q}(\lambda)$ is computed on a grid and extended between grid points by a simple rule, the allocation can become piecewise smooth (or piecewise constant) in λ, making critical bids easier to bracket and compute with a small number of evaluations. The second is to develop *dual* or *implicit* payment computation methods that reuse the optimization artifacts produced by the projection solver (e.g., Lagrange multipliers associated with monotonicity constraints) to approximate marginal influence and hence payments, reducing additional model calls. Establishing when such approximations preserve incentive properties—and how to bound the error in payments in a way that is meaningful to bidders and regulators—remains open.

Taken together, these extensions and limitations reinforce the core message: projected pooling is best understood as a modular enforcement layer whose value is clearest when bids are scalar, preferences admit a robust monotone order, and per-step influence is the relevant object. Extending the approach to many advertisers, semantic constraints, and sequence-level objectives is feasible in fragments, but a fully unified theory that is simultaneously computationally light, incentive-robust, and universally auditable is still an important research frontier.

5 Conclusion

We set out to reconcile two desiderata that are individually natural but jointly in tension in token-level *influence auctions* for language generation. On the one hand, there is a compelling *benchmark aggregator* motivated by RLHF and information geometry: log-linear (geometric) pooling,

$$q_t^{\mathrm{geo}}(\lambda) \;\propto\; p_{1,t}^{\lambda} p_{2,t}^{1-\lambda},$$

which is characterized as the unique minimizer of a weighted sum of reverse KL objectives. On the other hand, there is an equally compelling *incentives constraint*: if we want a second-price-style implementation with stable sampling and critical bids, we need a monotonicity property of the allocation rule with respect to bids, interpreted through a robust partial order. The central finding is that the RLHF benchmark and the incentives constraint can diverge even in the simplest two-advertiser environment, for a surprisingly mundane reason: normalization couples tokens, and that coupling can induce non-monotone movement in individual token probabilities even when every unnormalized score is monotone in the bid weight.

Our main conceptual move is to treat monotonicity as a *feasibility constraint* and to enforce it by a *projection* rather than by designing an allocation rule from scratch. We define the convex set of robust-monotone paths from p_2 to p_1 (monotone in λ in each coordinate with direction determined by $\mathrm{sign}(p_{1,t} - p_{2,t})$), and we select the closest such path to the RLHF benchmark by solving

$$\widehat{q} \;\in\; \arg\min_{q \in \mathcal{M}(p_1, p_2)} \int_0^1 D(q(\lambda) \,\|\, q^{\mathrm{geo}}(\lambda))\; d\lambda,$$

for a strictly convex divergence D. This "projected pooling" viewpoint cleanly separates *what we would like to do* (log-linear pooling for welfare/fit) from *what we must do* (monotone implementability for incentives), and it yields an interpretable scalar measure of the repair magnitude—the projection distance—that can be logged and audited.

Three implications follow from this framing. First, the projected rule $\widehat{q}(\lambda)$ is robustly monotone by construction, and therefore it inherits the mechanism-design machinery of Dütting et al. (2024): stable sampling exists, and per-token critical-bid payments can be defined with an implementation-independent expected payment identity. In economic terms, the projection is not merely a smoothing device; it is the minimal deformation needed to restore a single-crossing-like property in a setting where preferences are defined over distributions rather than over deterministic outcomes.

Second, the projection is *computationally plausible* at the per-token scale that LLM deployment demands, at least in the two-advertiser setting. Once we discretize λ on a grid, the feasible set becomes a polytope defined by linear monotonicity constraints and simplex constraints. With squared ℓ_2 loss, the

resulting optimization admits fast primitives (isotonic regression across λ for each token, plus simplex projection for each λ) that converge to the unique optimum. This matters practically because the per-token mechanism must run at generation latency, and it matters conceptually because it turns a mechanism-design desideratum (monotonicity) into an explicit and checkable computational artifact (a convex projection with a unique solution).

Third, the approach yields an explicit way to reason about the welfare–incentives tradeoff. Since $q^{\text{geo}}(\lambda)$ is the pointwise optimum for the RLHF objective, any deviation has an opportunity cost. The projection supplies a principled candidate and also supplies a distance-to-benchmark quantity that controls the objective loss under standard smoothness/strong convexity conditions (on appropriately truncated simplices). While such bounds are inevitably approximate—and depend on how one metrizes the simplex—they provide a language for system designers and auditors to quantify how much "incentive compatibility" is costing in "RLHF fit" at each step, and how that cost varies with disagreement between advertisers, temperature smoothing, and discretization.

Stepping back, we view the broader contribution as methodological. In many platform settings, the mechanism designer faces a familiar pattern: an unconstrained objective suggests a clean rule (here, log-linear pooling), but the implementation requires a monotonicity structure that the clean rule may violate. The standard response in auction theory is to *redesign* the allocation until it is monotone, often losing the connection to the benchmark. Our response is to *project* the benchmark onto the implementable set, retaining a direct quantitative link between the rule we implement and the rule we would have preferred absent incentives constraints. That link is useful not only for welfare analysis but also for governance: it makes deviations from the benchmark explicit, measured, and therefore debatable.

The limitations of the current analysis are equally important. The cleanest results rely on the two-advertiser reduction to a scalar weight λ, which makes monotonicity one-dimensional and gives isotonic regression its force. As we discussed, moving to many advertisers, to attribute-based preference orders, or to sequence-level incentives introduces new degrees of freedom and new failure modes. In particular, per-token monotonicity is a local property, whereas bidders and regulators often care about global properties of the full generated text; bridging that gap requires dynamic mechanism design and a careful treatment of state dependence through the prefix.

There are also modeling choices that should be read as commitments rather than as inevitabilities. We adopt robust partial orders because they deliver a tractable monotonicity notion with clear implementability implications, but any such order encodes a normative stance about what kinds of shifts in a distribution should count as "improvements" for an advertiser. If that stance is misaligned with the true objectives (or is manipulable), then truthful bidding can fail to deliver socially desirable outcomes even if

the mechanism is incentive compatible in the formal sense. Likewise, we focus on strictly convex divergences D to obtain uniqueness and stability of the projection; this stabilizes computation and interpretation, but it also privileges certain geometries of the simplex that may not perfectly capture perceived differences in language quality.

From a deployment perspective, we see two concrete takeaways. The first is that *monotonicity violations should be expected* under naive log-linear pooling, and therefore stable-sampling implementations that assume monotonicity can silently fail if the allocation rule is not repaired. The second is that the repair can be implemented in a way that is both transparent and auditable: the platform can expose (at least internally, and potentially to external auditors under privacy constraints) the projection distance, the active monotonicity constraints, and the discretization parameters. This does not solve the full governance problem, but it provides a disciplined starting point for it: disagreements can be grounded in observable quantities (how often the benchmark violates monotonicity; how large the projection is), rather than in opaque model behavior.

Several research directions look especially valuable. At the theory level, we would like sharper characterizations of when q^{geo} is already robustly monotone (or "approximately" monotone) and how that depends on p_1, p_2; such results would turn our comparative statics into more operational diagnostic tests. At the algorithmic level, we would like low-query payment computation methods that exploit the dual variables of the projection, reducing the incremental cost of critical-bid calculations. At the mechanism-design level, we would like principled multi-advertiser generalizations that avoid arbitrary merge trees while preserving implementability and computational tractability. And at the governance level, we would like auditing protocols that remain meaningful under token truncation, privacy constraints, and shifting vocabularies, potentially via compact certificates of correct projection.

The model we studied is intentionally modest: a per-token, two-advertiser mechanism with a structured monotonicity constraint. But the tradeoff it illuminates is not modest. As LLMs become platforms, "influence" over generation becomes a scarce resource, and platform designers will face pressure to allocate that influence through rules that are simultaneously efficient, incentive compatible, and explainable. Projected pooling offers one concrete template: begin with a welfare-motivated benchmark, enforce implementability by a minimal convex repair, and log the distortion as an auditable measure of the welfare–incentives compromise. We do not claim this resolves the full problem of monetizing influence over language. We do claim it clarifies where the hard parts live, and it provides a tractable, modular layer that can be combined with richer preference models and stronger governance constraints as the field matures.