Blueprint Auctions for LLM Interfaces: Efficient Mechanisms When the Platform Chooses the Generated Narrative

Liz Lemma Future Detective

January 16, 2026

Abstract

Generative assistants in 2026 do not display ads in a fixed list of slots; instead, the platform chooses how to generate the narrative (tone, disclosure intensity, and where insertions can appear), which endogenizes the set and quality of ad positions. We formalize this by introducing a blueprint parameter θ that determines (i) the candidate positions embedded in the generated content and (ii) a context-dependent matrix of standalone click-through rates $p_{ij}(\theta, c)$ for advertiser–position pairs, estimated by LLM-era predictors. Building on the winner-determination and mechanism-design reductions in Balseiro et al. (2025), we treat blueprint choice as an outer optimization and allocation/order choice as an inner optimization. Our first result is an ε-menu restriction theorem: under a low-dimensional and Lipschitz stability condition, there exists a finite menu Θ_ε of size $\mathrm{poly}(1/\varepsilon)$ (for constant dimension) such that optimizing over Θ_ε is within ε of optimizing over the full blueprint space. Our second result shows that under the MNL model the inner winner-determination problem can be solved exactly for each blueprint via the linear-fractional-to-LP transformation of the source paper, yielding a computationally efficient DSIC/IR welfare-maximizing auction and an ε'-DSIC/IR revenue-maximizing auction. We discuss extensions to order-sensitive cascade behavior using monotone approximations and highlight how blueprint auctions provide an auditable handle for disclosure, brand-safety, and user-trust constraints.

Table of Contents

3. 3. Blueprint Winner Determination: define the global WDP over (blueprint, matching, order); monotonicity implications and mechanism-design reduction.

4. 4. ε-Menu Restriction Theorem: covering-number construction for low-dimensional blueprint spaces; additive approximation guarantees; discussion of when assumptions hold/fail.

5. 5. Efficient Mechanisms under MNL: exact inner WDP per blueprint via LP transformation; DSIC welfare mechanism (VCG) and ε'-DSIC revenue mechanism (Myerson + envelope discretization).

6. 6. Extensions to Order-Sensitive Users (Cascade): what can be salvaged with monotone approximations; where numerical methods/heuristics are needed.

7. 7. Practical Considerations: estimating $p_{ij}(\theta, c)$, validating Lipschitz stability, blueprint parameterization in real LLM systems, and auditing/disclosure constraints.

8. 8. Comparative Statics and Policy: price of disclosure intensity; welfare vs. trust; implications for transparency regulation and platform commitment.

9. 9. Conclusion and Open Problems: beyond Lipschitz (non-smooth blueprint effects), multi-parameter advertisers, and dynamic multi-turn blueprint choice.

1 Introduction: Generative interfaces, endogenous positions, and why blueprint choice is the new design lever

Search and recommendation advertising were built around a stable abstraction: there are a few clearly delineated slots, and the platform solves an assignment problem—which ads to show, and where. Generative interfaces unsettle that abstraction. In an LLM-mediated experience, the "page" is not a fixed layout but a piece of text (or multimodal content) composed on demand. Where an ad can appear, how salient it is, and even whether it is natural to insert an ad at all are no longer exogenous primitives. They are design decisions embedded in the generation process itself. As a result, the platform does not merely allocate advertisers to positions; it chooses the *template* that creates positions.

We refer to these template-level decisions as the *blueprint*. A blueprint specifies a structured set of generation and disclosure choices: ad density (how many insertion opportunities are created), disclosure strength (how explicitly sponsored content is marked), and narrative tone (how the ad is integrated into the assistant's voice), among other knobs. Importantly, these knobs are not mere cosmetic tweaks. They change users' propensity to attend to and click content, and they impose trust and compliance consequences that platforms increasingly internalize. In practice, blueprint choices show up as: whether an answer contains a dedicated "Sponsored" block; whether the assistant offers "recommended products" inline; how close sponsored content is placed to the direct answer; and how aggressively the system follows up with commercial suggestions.

The economic novelty is that blueprint selection is an *outer* decision that endogenizes the feasible set for the familiar *inner* decision of advertiser selection and ordering. In a static-slot world, we can often treat the slot set as fixed and run a mechanism on top. In a generative world, the platform chooses the slot set (or, more precisely, a distribution over insertion loci and their salience) as part of optimizing revenue or welfare. This matters both computationally and strategically. Computationally, the platform faces a joint optimization over (i) content-structure parameters and (ii) allocations to advertisers. Strategically, a mechanism that is truthful for a fixed slot auction can fail to remain truthful when the slot set itself becomes bid-dependent via blueprint optimization: a higher bid can cause the platform to select a more ad-heavy blueprint, changing click probabilities in a way that feeds back into incentives.

A second novelty is that the platform's objective is no longer well summarized by expected clicks times value. Generative assistants operate under a tighter legitimacy constraint than traditional display: users interpret the system as an advisor, not merely a publisher. This creates a real cost to ag-

gressive monetization, even when it raises short-run revenue. That cost can reflect policy constraints (e.g., disclosure requirements, sector-specific restrictions), reputational considerations (user trust and retention), and product constraints (answer quality). For this reason, we model blueprint choice as trading off monetization against a blueprint-dependent cost. In the language of mechanism design, this cost enters the platform objective like a negative welfare term, capturing that some click probability is "too expensive" once we account for disruption, compliance risk, or long-run engagement loss.

The 2026 framing is that blueprint selection is becoming the central lever by which platforms manage this tradeoff. Three forces push in that direction. First, generative systems can create high-dimensional variation in presentation; yet, in a production environment, this variation is necessarily constrained to a small number of controllable knobs (policy teams demand predictable behavior, and engineering teams demand testable surfaces). Second, regulators and app-store policies increasingly specify *how* monetization must be presented, not merely *that* it exists. Third, user behavior is sensitive not only to which ads are shown, but to the narrative context in which they appear; the same product link can be perceived as helpful recommendation or as intrusive promotion depending on blueprint.

Our approach is to treat the blueprint as a low-dimensional parameter that controls the mapping from context to click behavior and cost, and then to embed standard assignment and mechanism-design reasoning inside that outer choice. Conceptually, we want two properties. The first is *stability*: small changes in blueprint should not cause discontinuous jumps in predicted click probabilities or costs. Without stability, optimization over blueprints becomes brittle, and approximation guarantees are hard to obtain. The second is *implementability*: once a finite menu of blueprints is fixed, the induced allocation rule should preserve monotonicity in bids so that incentive-compatible payments can be computed by familiar envelope arguments.

This perspective yields a disciplined way to think about a practical question that product teams confront: should the blueprint be optimized continuously (e.g., by gradient-based tuning over a rich prompt space) or discretely (e.g., by choosing among a small set of vetted templates)? We argue that a finite menu is not merely a product convenience; it can be theoretically justified when the blueprint-response mapping is sufficiently smooth. A dense enough menu approximates the best continuous blueprint up to a controlled additive loss, while allowing the platform to run exact winner-determination within each menu item. This combination—finite menu plus exact inner optimization—is precisely what preserves the monotonicity structure required for truthfulness in single-parameter advertiser settings.

We emphasize what we are *not* claiming. We do not claim that all generative blueprints are smooth; in fact, many realistic templates induce threshold behavior (e.g., turning on a shopping carousel if the query is "commercial

enough"). Nor do we claim that predicted click-through rates are known without error. Our goal is more modest and, we think, useful: to isolate the conditions under which blueprint choice can be integrated into mechanism design without breaking incentive guarantees, and to clarify when those conditions fail and must be addressed by discrete design and empirical validation.

The model we develop is designed to illuminate the platform's core tension. On one side is allocative efficiency (or revenue) from matching advertisers to high-attention insertion loci. On the other side is a cost of disruption that is inherently blueprint-driven. Because the blueprint determines how insertion loci are created and perceived, it is the natural locus for policy constraints: disclosure strength, placement separation between organic and sponsored content, and limits on ad density can all be represented as blueprint parameters or as blueprint-dependent feasibility constraints. This makes the model relevant not only for platform design but also for regulators: many policy debates are, implicitly, about restricting the blueprint space, not merely about taxing clicks or limiting advertiser participation.

To make these ideas operational, we organize the analysis around three contributions.

- *Endogenous positions via blueprint choice.* We formalize the notion that positions are not fixed primitives in a generative interface. Blueprint parameters determine the salience and feasibility of insertion loci and therefore the effective click probabilities delivered by any allocation.

- *A menu-restriction justification.* Under a stability condition, we show that optimizing over a finite blueprint menu can approximate optimizing over a continuous blueprint space. This bridges a product reality (vetting a handful of safe templates) with an economic guarantee (bounded objective loss).

- *Mechanism-design compatibility.* We show that when the platform solves the joint problem exactly over the finite menu, the induced allocation is monotone in bids, allowing standard payment constructions for truthful welfare or revenue mechanisms in the single-parameter setting.

Two practical interpretations follow. First, the blueprint can be viewed as the platform's "ad policy" encoded in a low-dimensional object. A tighter policy regime (stronger disclosure, fewer insertion loci, more separation between organic and sponsored content) corresponds to a region of the blueprint space with higher trust cost but potentially lower immediate monetization. Second, engineering constraints that require a small set of tested prompts are not merely constraints; they can be aligned with incentive-compatibility needs. If the platform were to optimize over an unconstrained prompt space

in a bid-dependent way, it would be much harder to ensure monotonicity and compute payments reliably.

Finally, we acknowledge a limitation that motivates later discussion. Generative systems can introduce discontinuities through discrete content choices: whether the assistant decides to present a list, a comparison table, or a single narrative; whether it calls a tool that returns product cards; whether it asks a follow-up question before showing offers. Such decisions can cause abrupt changes in where ads can appear and in how users respond. In those regimes, smooth covering arguments become fragile, and the right abstraction may be a genuinely discrete family of blueprints treated as distinct products rather than points in a continuous parameter space. We see this not as a failure of the framework but as a diagnostic: when smoothness fails, the economics points toward discrete governance (hand-designed menus, policy audits, and controlled experimentation) as the appropriate complement to mechanism design.

The remainder of the paper develops a model that makes these intuitions precise. We define the context space, the blueprint parameter space, the feasible allocation and ordering decisions, and two canonical user click models that capture order-insensitive and order-sensitive attention. We then incorporate blueprint-dependent costs and show how menu restriction and exact optimization together allow us to preserve tractable computation and incentive guarantees in a setting where the very notion of an "ad slot" is endogenously chosen by the platform.

2 Model: contexts, blueprints, endogenous positions, and click behavior

We model a generative interface as a two-layer decision problem. The *outer layer* is a blueprint choice that governs how the assistant will structure its response and where sponsored content can plausibly appear. The *inner layer* is the familiar allocation problem: given a set of candidate insertion loci and predicted user responses, which advertisers should be placed where (and, when order matters, in what order). This section defines the primitives needed to make that separation precise while keeping the blueprint space close to how product teams actually ship generative templates.

2.1 Contexts, advertisers, and the single-parameter environment

A context $c \in \mathcal{C}$ summarizes the user query together with relevant conversation state (e.g., prior turns, inferred intent, and any product-mode flags). The platform observes c at the time it must decide how to render an answer. There are n advertisers indexed by $i \in [n]$. Advertiser i has a per-click

value $v_i \in [0, \bar{v}]$; we adopt the standard single-parameter model in which the advertiser cares only about expected clicks (or conversions proportional to clicks), not about the specific wording of the generated content.

Advertisers submit reports b_i, which are the numbers that enter the platform's optimization objective. For welfare analysis one may take $b_i = v_i$; for revenue analysis under regularity, one may instead take $b_i = \phi_i(v_i)$, the Myerson virtual value. We keep the model agnostic about the payment rule for now; the key object here is how (c, b) maps into expected click probabilities, and how that mapping depends on a blueprint.

2.2 Blueprints as low-dimensional template parameters

A blueprint is a parameter $\theta \in \Theta \subset \mathbb{R}^d$, where Θ is compact and the dimension d is treated as constant. The intention is that θ collects a small set of controllable knobs that are stable enough to be audited and A/B tested: examples include ad density (how many insertion opportunities are created), disclosure strength (how clearly sponsorship is labeled), separation rules (distance between organic answer and sponsored insertions), and narrative integration (whether sponsored content appears as a distinct block or as inline recommendations).

We emphasize why the low-dimensional restriction is not merely mathematical convenience. In production, templates must satisfy compliance and product-review constraints that are difficult to guarantee in an unconstrained prompt space. Treating θ as low-dimensional is a way to formalize the "safe surface" on which platforms can credibly commit to predictable behavior.

2.3 Candidate insertion loci and feasibility of allocations

Fix an upper bound m on the number of *candidate* positions the system could use for sponsored content in a given rendering. We index candidates by $j \in [m]$. The point of treating m as an upper bound is that it allows the blueprint to influence *effective* position availability and salience without requiring us to redefine the combinatorial structure each time. Intuitively, a conservative blueprint can make many candidate loci effectively unattractive (very low click propensity) or infeasible (disallowed by policy), while an aggressive blueprint can activate them.

An allocation is a matching $x \in \{0,1\}^{n \times m}$ with a cap of $K \leq m$ total insertions:

$$X := \Big\{ x \in \{0,1\}^{n \times m} : \sum_{j=1}^{m} x_{ij} \leq 1 \ \forall i, \ \sum_{i=1}^{n} x_{ij} \leq 1 \ \forall j, \ \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \leq K \Big\}.$$

Thus each advertiser can appear at most once, each position holds at most one advertiser, and at most K sponsored insertions are made. When user attention depends on the sequence in which insertions are encountered, we

also include an order $\sigma \in \Sigma$, where Σ is the set of permutations of $[m]$. The augmented allocation is then $(x, \sigma) \in X \times \Sigma$.

This abstraction is deliberately neutral about *where* a position sits in the generated answer. Position j should be read as a structured locus type (e.g., "top-of-answer sponsored block," "inline recommendation after first paragraph," "end-of-answer product card"), rather than a literal slot on a static page.

2.4 CTR primitives: standalone response to an advertiser-position pair

For each context c and blueprint θ, the platform has predicted *standalone* click-through rates

$$p_{ij}(\theta, c) \in [0, 1],$$

interpreted as the propensity to click advertiser i if placed in candidate locus j under blueprint θ, absent interactions with other insertions. In practice, $p_{ij}(\theta, c)$ is estimated from logged experiments and model-based extrapolation: it depends on the query, the ad's relevance, and blueprint-driven presentation features such as labeling and prominence.

When it is convenient, we also use the log-odds parameter

$$\rho_{ij}(\theta, c) := \log \frac{p_{ij}(\theta, c)}{1 - p_{ij}(\theta, c)},$$

which aligns naturally with multinomial logit formulations below.

The standalone p_{ij} is not the final click probability when multiple insertions are shown. The mechanism-design object is the *final* click probability $\pi_i(\theta, c; x, \sigma)$, which aggregates position effects and interaction effects implied by a user-attention model.

2.5 Two user click models: MNL and cascade

We analyze two canonical behavioral models that capture distinct product regimes.

MNL (order-insensitive) clicks. In settings where multiple sponsored items are displayed in a block and user choice is well approximated as a single discrete selection among displayed options (plus an outside option), we use a multinomial logit model. For an allocation $x \in X$, advertiser i's click probability is

$$\pi_i(\theta, c; x) = \frac{\sum_{j=1}^{m} x_{ij} \exp(\rho_{ij}(\theta, c))}{1 + \sum_{i'=1}^{n} \sum_{j=1}^{m} x_{i'j} \exp(\rho_{i'j}(\theta, c))}.$$

The outside option corresponds to "no click" and has normalized attractiveness 1. The MNL structure captures the empirically relevant feature

that adding more alternatives can cannibalize clicks from existing alternatives through the shared denominator. Importantly, this model is *order-insensitive*: only the set of displayed items matters, not their sequence.

Cascade (order-sensitive) clicks. In conversational flows, users may encounter insertions sequentially as they read the assistant's response. To capture attention decay and stopping behavior, we use a cascade model in which users scan positions in an order σ and may click at most once. For augmented allocation (x, σ), we write

$$\pi_i(\theta, c; x, \sigma) = \sum_{j=1}^{m} x_{ij} p_{ij}(\theta, c) \prod_{j':\ \sigma(j')<\sigma(j)} \Big(1 - \sum_{i'=1}^{n} x_{i'j'} p_{i'j'}(\theta, c)\Big).$$

The product term is the probability the user has not clicked earlier positions in the scan. This specification makes explicit why ordering and placement policy become first-order in generative interfaces: moving an insertion earlier can increase its own clicks while reducing downstream clicks, a tradeoff that depends on relevance and disclosure.

We view these models as complementary benchmarks. MNL is appropriate for dedicated sponsored modules or carousels; cascade is appropriate for inline insertions and sequential reading. Both can be estimated and validated empirically, and both expose the central mechanism-design issue: click probabilities depend on the platform's joint choice of blueprint and allocation (and possibly order).

2.6 Blueprint-dependent costs and constraints

Blueprints affect not only click behavior but also product quality, trust, and compliance. We capture these considerations through a blueprint-dependent cost $\kappa(\theta, c) \geq 0$, measured in the same units as welfare (so that it can be subtracted from value-weighted clicks). Conceptually, κ can represent any combination of (i) expected user harm from disruptive monetization, (ii) long-run retention loss internalized by the platform, and (iii) compliance and review costs induced by certain layouts or labeling choices.

A flexible interpretation is to decompose

$$\kappa(\theta, c) = \lambda \cdot \text{Disruption}(\theta, c),$$

where λ is a policy weight chosen by the platform (or effectively imposed by regulation or internal governance). This formulation highlights an actionable lever: increasing λ shifts the optimum toward conservative blueprints (fewer or less prominent insertions, stronger disclosure), potentially reducing short-run clicks while improving net welfare once disruption is priced in.

Constraints can enter in two equivalent ways. First, they can be encoded directly into the feasible set by ruling out allocations that violate blueprint-specific rules (e.g., disallowing certain insertion loci for sensitive contexts). Second, they can be penalized through $\kappa(\theta, c)$ to reflect "soft" constraints that are discouraged but not forbidden. For most of the analysis, we treat feasibility as fixed via X and allow blueprint effects to operate through $\pi_i(\cdot)$ and $\kappa(\cdot)$; this keeps the inner allocation structure stable and isolates the role of blueprint choice.

2.7 Stability as a modeling assumption and its practical meaning

To connect the continuous blueprint space to finite template menus, we impose a stability condition: small changes in θ should not cause large changes in induced click probabilities or in the cost. Formally, we assume there exists $L > 0$ such that for all contexts c, all augmented allocations (x, σ), and all advertisers i,

$$\big|\pi_i(\theta, c; x, \sigma) - \pi_i(\theta', c; x, \sigma)\big| \le L\|\theta - \theta'\|_2, \qquad \big|\kappa(\theta, c) - \kappa(\theta', c)\big| \le L\|\theta - \theta'\|_2.$$

This assumption is best read as a disciplined approximation: it holds when blueprint knobs correspond to continuous presentation parameters (e.g., prominence scaling, disclosure phrasing intensity, or probabilistic insertion rates) and when the generation system is engineered to avoid threshold-triggered layout changes. We also acknowledge its limits. Many real templates include discontinuous logic (e.g., turning on a shopping module only when a classifier crosses a threshold), in which case stability fails and blueprint choice must be treated as genuinely discrete. Our later results use stability to justify finite menus; when stability is implausible, the model still helps by clarifying where approximation and incentive guarantees become fragile.

With these primitives in place, the platform's decision problem can be stated cleanly: for a realized context c and bid vector b, the platform chooses a blueprint θ and an allocation (and possibly an order) to maximize a value-weighted click objective net of $\kappa(\theta, c)$. The next section formalizes this as a global winner-determination problem over (θ, x, σ) and studies the monotonicity properties that enable incentive-compatible mechanisms.

3 Blueprint winner determination: a global WDP over (θ, x, σ)

Given a realized context c and reports $b \in [0, \bar{v}]^n$, the platform faces a *single* optimization problem that jointly selects (i) how the answer will be structured and disclosed (the blueprint θ), and (ii) which advertisers to place into which insertion loci (and, when relevant, in what encounter order). To

make this joint choice explicit, we write the winner-determination problem (WDP) in two nested layers.

Inner WDP for a fixed blueprint. Fix $\theta \in \Theta$. The platform then chooses an augmented allocation $(x, \sigma) \in X \times \Sigma$ to maximize bid-weighted predicted clicks net of the blueprint cost:

$$\mathrm{Opt}(\theta, c; b) \;:=\; \max_{(x,\sigma)\in X\times\Sigma} \left\{ \sum_{i=1}^{n} b_i \, \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c) \right\}. \tag{1}$$

When we work under MNL, $\pi_i(\theta, c; x, \sigma)$ is order-insensitive and σ can be dropped; under cascade (or any sequential attention model), σ is a genuine decision variable because the same set of insertions can induce different click vectors depending on their encounter order. The key point is that $\kappa(\theta, c)$ is *blueprint-level*: it does not depend on (x, σ) and thus shifts the value of choosing θ without altering the within-blueprint ranking across allocations.

Outer WDP over blueprints. The platform's global choice then selects the best blueprint as well:

$$\mathrm{Opt}(c; b) \;:=\; \max_{\theta\in\Theta} \mathrm{Opt}(\theta, c; b) \;=\; \max_{\theta\in\Theta,(x,\sigma)\in X\times\Sigma} \left\{ \sum_{i=1}^{n} b_i \, \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c) \right\}. \tag{2}$$

We emphasize an interpretive benefit of writing the problem in the global form (2). A blueprint is not merely a cosmetic wrapper around a fixed auction; it *reshapes* the click response functions $\pi_i(\theta, c; \cdot)$ and the effective tradeoff between monetization and trust through $\kappa(\theta, c)$. At the same time, once θ is fixed the inner problem is a standard assignment-and-ordering optimization with predicted CTR primitives. This is exactly the separation product teams often implement in practice: a limited set of audited templates, each with a well-defined allocation routine.

3.1 Allocation rules induced by exact global optimization

Let a (deterministic) platform policy be a mapping that, for each (c, b), selects a maximizing triple

$$(\theta^*, x^*, \sigma^*) \in \arg \max_{\theta\in\Theta,(x,\sigma)\in X\times\Sigma} \left\{ \sum_{i=1}^{n} b_i \, \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c) \right\},$$

with a fixed tie-breaking rule (e.g., lexicographic over (θ, x, σ)). This induces an allocation rule in the mechanism-design sense: advertiser i's *service level* is the resulting click probability

$$q_i(c, b) \;:=\; \pi_i(\theta^*, c; x^*, \sigma^*).$$

In single-parameter environments, the central structural requirement for dominant-strategy incentive compatibility is monotonicity of $q_i(c,(b_i,b_{-i}))$ in b_i for each fixed (c,b_{-i}). The substantive question here is whether allowing the platform to optimize over blueprints can destroy the monotonicity that underlies familiar DSIC constructions.

Our answer is that it does not: blueprint choice enlarges the feasible set but does not change the linearity in bids.

3.2 Monotonicity with endogenous blueprints

Fix c and b_{-i}. Consider two bids for advertiser i, $b_i < b_i'$, and denote by (θ,x,σ) and (θ',x',σ') the (tie-broken) maximizers under (b_i,b_{-i}) and (b_i',b_{-i}), respectively. Exact optimality implies the pair of inequalities

$$b_i\,\pi_i(\theta,c;x,\sigma)+\sum_{k\neq i}b_k\,\pi_k(\theta,c;x,\sigma)-\kappa(\theta,c)\;\geq\;b_i\,\pi_i(\theta',c;x',\sigma')+\sum_{k\neq i}b_k\,\pi_k(\theta',c;x',\sigma')-\kappa(\theta',c) \tag{3}$$

$$b_i'\,\pi_i(\theta',c;x',\sigma')+\sum_{k\neq i}b_k\,\pi_k(\theta',c;x',\sigma')-\kappa(\theta',c)\;\geq\;b_i'\,\pi_i(\theta,c;x,\sigma)+\sum_{k\neq i}b_k\,\pi_k(\theta,c;x,\sigma)-\kappa(\theta,c). \tag{4}$$

Adding (3) and (4) cancels the terms involving other bidders and κ, yielding

$$b_i\,\pi_i(\theta,c;x,\sigma)+b_i'\,\pi_i(\theta',c;x',\sigma')\;\geq\;b_i\,\pi_i(\theta',c;x',\sigma')+b_i'\,\pi_i(\theta,c;x,\sigma).$$

Rearranging gives

$$(b_i'-b_i)\Big(\pi_i(\theta',c;x',\sigma')-\pi_i(\theta,c;x,\sigma)\Big)\geq 0,$$

and since $b_i'-b_i>0$ we obtain

$$\pi_i(\theta',c;x',\sigma')\geq\pi_i(\theta,c;x,\sigma).$$

Thus, under exact global optimization with deterministic tie-breaking, advertiser i's click probability is weakly increasing in b_i. The proof is identical in spirit to the standard monotonicity argument for assignment problems: the platform maximizes a linear objective in bids over a fixed feasible set, and enlarging the feasible set from $X\times\Sigma$ to $\Theta\times X\times\Sigma$ does not change the algebra.

Two practical caveats are worth making explicit. First, tie-breaking matters only to ensure the mapping $(c,b)\mapsto(\theta^*,x^*,\sigma^*)$ is single-valued; otherwise monotonicity is a set-valued statement. Second, monotonicity is fragile to approximation: if the platform uses a heuristic that sometimes returns a non-maximizer, then the inequalities (3)–(4) need not hold, and DSIC guarantees may fail even if the heuristic is "almost" optimal in objective value.

3.3 Mechanism-design reduction: from global WDP to DSIC payments

Once we have monotonicity, the remaining mechanism-design work is conceptually standard. The blueprint variable θ simply becomes part of the allocation outcome that the mechanism selects. What changes is not the truthfulness logic but the *domain* over which the platform must compute counterfactual objectives (e.g., leave-one-out maxima).

Welfare: VCG with blueprint choice. If we interpret reports as values ($b_i = v_i$) and the platform selects $(\theta^*, x^*, \sigma^*)$ to maximize $\sum_i v_i \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c)$, then we can apply VCG on the augmented outcome space $\Theta \times X \times \Sigma$. Let

$$\mathrm{Opt}_{-i}(c; v_{-i}) \;:=\; \max_{\theta,(x,\sigma)} \left\{ \sum_{k \neq i} v_k \, \pi_k(\theta, c; x, \sigma) - \kappa(\theta, c) \right\}$$

denote the optimal objective when advertiser i is removed. A standard VCG payment takes the form

$$t_i(v) \;=\; \mathrm{Opt}_{-i}(c; v_{-i}) - \left(\sum_{k \neq i} v_k \, \pi_k(\theta^*, c; x^*, \sigma^*) - \kappa(\theta^*, c) \right), \qquad (5)$$

i.e., i pays the externality it imposes on others *including* any blueprint-induced cost changes. This is important: removing i may cause the platform to choose a different blueprint θ with a different $\kappa(\theta, c)$, and VCG properly charges the difference. Under quasilinear utilities and exact optimization, the usual DSIC and individual rationality conclusions follow.

Revenue: virtual surplus and envelope payments. For revenue maximization under regularity, we replace values by virtual values $b_i = \phi_i(v_i)$ and maximize virtual surplus $\sum_i \phi_i(v_i)\pi_i(\cdot) - \kappa(\cdot)$. The monotonicity result above implies the resulting click-through allocation $y_i(v_i, v_{-i}) := \pi_i(\theta^*, c; x^*, \sigma^*)$ is weakly increasing in v_i (because ϕ_i is non-decreasing). Therefore, by the envelope theorem for single-parameter DSIC mechanisms, there exists a payment rule implementable via

$$t_i(v) \;=\; t_i(0, v_{-i}) + v_i \, y_i(v) - \int_0^{v_i} y_i(z, v_{-i}) \, dz,$$

with $t_i(0, v_{-i})$ chosen to satisfy individual rationality (often 0). In other words, blueprint optimization does not require a new payment theory; it requires that we can compute (or approximate) the interim click allocation as a function of the bid, which is ultimately an algorithmic question.

3.4 Why we isolate exact global WDP as the key primitive

The discussion above motivates treating $\mathrm{Opt}(c; b)$ as the fundamental object: if we can solve the global WDP exactly, then the monotonicity needed for truthful mechanisms is automatic, and familiar payment formulas apply with blueprint choice folded into the outcome. This perspective also clarifies where the real difficulty lies.

First, the feasible set $\Theta \times X \times \Sigma$ can be computationally burdensome even when each component is manageable: Θ may be continuous, and Σ may be enormous under order-sensitive models. Second, even when computation is feasible, production systems often rely on learned predictors and approximate solvers; these approximations can introduce non-monotonicities that are small in objective value but large for incentives. These are not merely technicalities: in ad auctions, slight non-monotonicities can be exploited and can create instability in bidding dynamics.

This is precisely why we next develop an ε-menu restriction for blueprints. By reducing the outer optimization from a continuous Θ to a finite Θ_ε while controlling objective loss, we make global optimization—and hence truthful implementation—closer to the operational reality of audited template menus.

3.5 ε-menu restriction for low-dimensional blueprint spaces

The computational and incentive arguments above place essentially all weight on a single primitive: the ability to solve the global optimization problem *exactly*. The obstruction, of course, is that the blueprint space Θ is typically modeled as a compact subset of $\mathbb{R}^d$ (to capture continuous knobs such as disclosure strength or ad prominence), and exact maximization over a continuous set is not what production systems implement. In practice, platforms commit ex ante to a *finite* set of audited templates—a menu of blueprints—and then optimize allocations within whichever template is chosen. We now formalize when this operational restriction is without much loss: if changing θ slightly only changes induced click probabilities and blueprint cost slightly, then an appropriately fine finite menu approximates the continuous optimum up to a small additive error.

Stability and the covering-number intuition. Fix a context c and bids b. For any fixed augmented allocation (x, σ), define the blueprint-level objective

$$F(\theta; c, b; x, \sigma) \;:=\; \sum_{i=1}^{n} b_i \, \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c).$$

Under the Lipschitz-stability assumption in the enclosing scope, $\theta \mapsto \pi_i(\theta, c; x, \sigma)$ and $\theta \mapsto \kappa(\theta, c)$ move continuously and at most linearly in $\|\theta - \theta'\|_2$. Consequently, so does F, with Lipschitz modulus proportional to the total bid mass $\sum_i b_i$. This is the core reason a finite ε-net is sufficient: we do not need

to preserve the *identity* of the optimal blueprint, only the objective value. If objective values cannot change too sharply as θ varies, then sampling Θ at a sufficiently fine resolution guarantees that at least one sampled blueprint θ' lies near the true optimizer θ^* and hence attains nearly the same objective.

A finite menu with an additive approximation guarantee. We make the preceding heuristic precise by constructing an ε-menu $\Theta_\varepsilon \subset \Theta$ using a standard covering-number argument. Because Θ is compact in $\mathbb{R}^d$, for any radius $\delta > 0$ there exists a finite δ-net $N(\delta) \subset \Theta$ such that for every $\theta \in \Theta$ there is $\theta' \in N(\delta)$ with $\|\theta - \theta'\|_2 \le \delta$. Moreover, the size of the net can be bounded by the usual volumetric estimate

$$|N(\delta)| \;\le\; \left(\frac{\operatorname{diam}(\Theta)}{\delta}\right)^d,$$

up to an absolute constant factor (which we suppress since d is treated as fixed).

Menu Restriction (Covering) Lemma. Fix any $\varepsilon > 0$. Let $\delta := \varepsilon/(2L)$, and let Θ_ε be any δ-net of Θ in ℓ_2. Then $|\Theta_\varepsilon| \le (\operatorname{diam}(\Theta) \cdot 2L/\varepsilon)^d$, and for every context c and bid vector $b \in [0, \bar{v}]^n$,

$$\max_{\theta \in \Theta} \operatorname{Opt}(\theta, c; b) \;\le\; \max_{\theta \in \Theta_\varepsilon} \operatorname{Opt}(\theta, c; b) \;+\; \varepsilon \cdot (1 + n\bar{v}). \tag{6}$$

An identical statement holds if we replace bids b by values v (welfare) or by virtual values $\phi_i(v_i)$ (virtual welfare), provided the reports are bounded in $[0, \bar{v}]$ and the same Lipschitz condition applies to π and κ.

Proof sketch and where the additive term comes from. The proof is a direct application of Lipschitz continuity plus the fact that the objective is linear in the click probabilities. Let $\theta^* \in \arg\max_{\theta \in \Theta} \operatorname{Opt}(\theta, c; b)$ be an optimizer, and choose $\hat{\theta} \in \Theta_\varepsilon$ such that $\|\hat{\theta} - \theta^*\|_2 \le \delta$. For any fixed (x, σ),

$$\begin{aligned} F(\theta^*; c, b; x, \sigma) - F(\hat{\theta}; c, b; x, \sigma) &= \sum_{i=1}^n b_i \Big(\pi_i(\theta^*, c; x, \sigma) - \pi_i(\hat{\theta}, c; x, \sigma)\Big) - \Big(\kappa(\theta^*, c) - \kappa(\hat{\theta}, c)\Big) \\ &\le \sum_{i=1}^n b_i \cdot L\|\theta^* - \hat{\theta}\|_2 + L\|\theta^* - \hat{\theta}\|_2 \\ &\le L\delta \cdot \Big(\sum_{i=1}^n b_i + 1\Big). \end{aligned}$$

Since $b_i \in [0, \bar{v}]$, we have $\sum_i b_i \le n\bar{v}$, and by the choice $\delta = \varepsilon/(2L)$ we obtain

$$F(\theta^*; c, b; x, \sigma) \le F(\hat{\theta}; c, b; x, \sigma) + \varepsilon \cdot \frac{n\bar{v} + 1}{2}.$$

Finally, letting (x^*, σ^*) be the maximizer under θ^* and taking maxima over (x, σ) under $\hat{\theta}$ can only improve the right-hand side, which yields (6) after absorbing the factor $1/2$ into a slightly looser $\varepsilon \cdot (1 + n\bar{v})$ presentation. Conceptually, the only role of the bid bound $\sum_i b_i \leq n\bar{v}$ is to translate a uniform per-advertiser Lipschitz bound into a bound on *total* objective deviation.

Interpretation: a low-dimensional "template knob" model. The lemma is easiest to interpret when d is genuinely small, so that the covering number $|\Theta_\varepsilon|$ grows moderately as ε shrinks. This corresponds to a design stance that we view as both economically and operationally natural: blueprints should be parameterized by a *small* set of interpretable knobs (e.g., number of insertion opportunities, prominence scaling, disclosure phrasing strength, or a scalar "commercial intensity" control). Under that stance, an ε-menu is not merely an existence claim; it is the formal analogue of an internal policy that restricts the product surface area to something auditable. The lemma then says that, provided user response and compliance cost are stable in those knobs, we lose at most an additive $\varepsilon \cdot (1 + n\bar{v})$ in the platform's objective by restricting to a finite set of approved designs.

When the Lipschitz assumption is reasonable. Although Lipschitz stability is a modeling assumption, it is not arbitrary. It is plausible when θ affects π through smooth, bounded transformations: for example, if θ continuously scales the attractiveness of sponsored content, adjusts disclosure intensity in a way that monotonically depresses attention to ads, or interpolates between two prompt templates via mixture weights. Under both MNL and cascade-style click models, if the underlying standalone probabilities $p_{ij}(\theta, c)$ are Lipschitz in θ and K is bounded, then the induced $\pi_i(\theta, c; x, \sigma)$ inherits a Lipschitz bound (with a constant that can worsen with K because of multiplicative continuation terms in cascade). Similarly, if $\kappa(\theta, c)$ is a smooth proxy for trust, policy risk, or long-run engagement loss, it is natural for it to vary continuously with disclosure and prominence knobs.

When stability fails: discontinuous blueprints and template cliffs. The main limitation is also the practically important one: real template systems often contain *discontinuities*. The set of candidate loci $j \in [m]$ can change abruptly with θ (turning on/off an insertion type), or the rendering logic can switch regimes (e.g., moving a sponsored unit from inline to a dedicated block once a threshold is crossed). In such cases, the map $\theta \mapsto \pi(\theta, c; x, \sigma)$ can have jumps even if user behavior is itself smooth, because the *meaning* of (x, σ) changes with the available positions. Likewise, disclosure can be categorical (a label either appears or does not), which can induce step changes in click propensities and in compliance cost. When these "template cliffs" occur, a covering argument in a continuous metric space is the wrong

tool: no finite ε-net can control objective loss if arbitrarily small parameter changes can flip the feasible layout.

Design responses: discrete families, mixtures, and smoothing. When stability fails, we should treat blueprint choice as fundamentally discrete, which is anyway closer to governance practice. One approach is to replace Θ by a finite union of smooth families (a small number of template archetypes, each with a few continuous knobs) and apply the covering argument within each family. Another is to model θ as a *randomization* parameter over a finite set of base templates, so that the induced click probabilities become continuous in mixture weights even if the base templates are discrete; this can restore Lipschitz continuity at the level of expected π. A third is to relax the notion of "position availability" so that insertion types fade in continuously (e.g., via probabilistic insertion or soft gating), which can be interpreted as smoothing the discontinuity and thus making an ε-menu guarantee meaningful again. Each response has a policy interpretation: it is a way of aligning the economic desire for exact optimization and incentive guarantees with the operational constraints of audited, version-controlled template menus.

What the theorem does *not* claim. Finally, we stress two boundaries of the result. First, the lemma is an *additive* approximation guarantee, not multiplicative; when the objective scale is small (e.g., very weak demand), additive bounds can be loose, and one may want normalization by the welfare scale or a context-dependent ε. Second, the lemma assumes the primitives $\pi_i(\theta, c; x, \sigma)$ and $\kappa(\theta, c)$ are known and stable; estimation error, non-stationarity, and strategic user adaptation can all violate the effective Lipschitz property even if the structural model is smooth. The main value of the ε-menu theorem is therefore as a *structural reduction*: it tells us that, in low-dimensional and stable blueprint spaces, we can reduce continuous blueprint choice to finite menu enumeration without sacrificing much objective value, thereby bringing exact global optimization (and hence truthful implementation) within reach of the mechanisms we study next.

3.6 Efficient mechanisms under MNL: exact optimization within each blueprint

Once we restrict attention to a finite, audited menu of blueprints, the remaining technical question is whether the platform can still perform the *exact* maximization required by the incentive arguments. Under the multinomial logit (MNL) click model, the answer is affirmative: for each fixed blueprint $\theta \in \Theta_\varepsilon$, the inner winner-determination problem (WDP) has a tractable linear programming formulation, and the global problem over the menu is solved by enumerating θ and selecting the best inner optimum. We

then obtain (i) a welfare-maximizing DSIC/IR mechanism via VCG payments on the enlarged outcome space that includes blueprint choice, and (ii) a revenue-oriented mechanism that implements Myerson's virtual-surplus objective with payments computed by an envelope formula, up to a small ε' due only to numerical integration.

The MNL inner WDP as a linear-fractional program. Fix a context c, a blueprint θ, and reports $b \in [0, \bar{v}]^n$. Under MNL, order does not matter and click probabilities depend on the set of shown ads only through their *attractiveness*. It is convenient to write

$$a_{ij}(\theta, c) := \exp(\rho_{ij}(\theta, c)) \geq 0,$$

so that, for a feasible matching $x \in X$,

$$\pi_i(\theta, c; x) = \frac{\sum_{j=1}^{m} x_{ij} a_{ij}(\theta, c)}{1 + \sum_{i'=1}^{n} \sum_{j=1}^{m} x_{i'j} a_{i'j}(\theta, c)}.$$

The blueprint-specific objective is therefore

$$\mathrm{Opt}(\theta, c; b) = \max_{x \in X} \left\{ \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} b_i \, a_{ij}(\theta, c) \, x_{ij}}{1 + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}(\theta, c) \, x_{ij}} - \kappa(\theta, c) \right\}. \tag{7}$$

The key point is that, for fixed θ and c, the dependence on x is *linear-fractional*: both numerator and denominator are linear functions of the assignment variables x_{ij}.

LP transformation and integrality. Problem (7) can be solved exactly by the standard Charnes–Cooper transformation. Introduce

$$z := \frac{1}{1 + \sum_{i,j} a_{ij}(\theta, c) x_{ij}} \qquad \text{and} \qquad y_{ij} := z \, x_{ij}.$$

Substituting into (7), the fractional term becomes $\sum_{i,j} b_i a_{ij}(\theta, c) y_{ij}$, while the denominator normalization becomes a linear constraint:

$$z + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}(\theta, c) y_{ij} = 1, \qquad z \geq 0, \qquad y_{ij} \geq 0.$$

The matching constraints $\sum_j x_{ij} \leq 1$, $\sum_i x_{ij} \leq 1$, and $\sum_{i,j} x_{ij} \leq K$ transform into

$$\sum_{j=1}^{m} y_{ij} \leq z \quad (\forall i), \qquad \sum_{i=1}^{n} y_{ij} \leq z \quad (\forall j), \qquad \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ij} \leq K z.$$

Thus, for each θ, we obtain the linear program

$$\max_{y\geq 0, z\geq 0} \sum_{i=1}^{n}\sum_{j=1}^{m} b_i a_{ij}(\theta,c) y_{ij} - \kappa(\theta,c) \quad \text{s.t.} \quad \begin{cases} \sum_j y_{ij} \leq z & \forall i, \\ \sum_i y_{ij} \leq z & \forall j, \\ \sum_{i,j} y_{ij} \leq Kz, & \\ z + \sum_{i,j} a_{ij}(\theta,c) y_{ij} = 1. & \end{cases} \tag{8}$$

Two observations matter for mechanism design. First, (8) is polynomial-time solvable in (n, m). Second, despite the original binary constraint $x_{ij} \in \{0, 1\}$, we do not lose exactness: the feasible region of the matching constraints is an assignment-type polytope, and the transformation preserves the property that extreme points correspond to scaled matchings. With deterministic tie-breaking, we can recover an optimal $x \in X$ from an optimal (y, z) without sacrificing objective value. Operationally, one may view (8) as a disciplined way to exploit the special structure of MNL: it collapses the strategic problem to a tractable exact optimizer over the menu.

Global optimization over the blueprint menu. Given Θ_ε, the platform solves (8) separately for each $\theta \in \Theta_\varepsilon$ and selects the best resulting outcome. The total running time is therefore $\text{poly}(n, m, |\Theta_\varepsilon|)$, and the only dependence on continuous blueprint choice is through the menu size. In particular, the platform's commitment to a finite set of audited templates is compatible with exact maximization, which is the enabling condition for the monotonicity and incentive results.

Welfare maximization: VCG with blueprint choice. For welfare, we set $b_i = v_i$ and choose the outcome (θ^*, x^*) maximizing

$$\sum_{i=1}^{n} v_i \, \pi_i(\theta, c; x) - \kappa(\theta, c),$$

where θ ranges over Θ_ε and x over X. This is precisely a VCG setting with an outcome space enlarged to include blueprint selection. The standard VCG payment for advertiser i is the externality imposed on others:

$$t_i(v) = \max_{\theta\in\Theta_\varepsilon, x\in X} \left\{ \sum_{\ell\neq i} v_\ell \, \pi_\ell(\theta, c; x) - \kappa(\theta, c) \right\} - \left\{ \sum_{\ell\neq i} v_\ell \, \pi_\ell(\theta^*, c; x^*) - \kappa(\theta^*, c) \right\}. \tag{9}$$

Because $\kappa(\theta, c)$ is part of the platform's objective and does not depend on any single advertiser's report, it enters (9) exactly as any other common-value term would in VCG: removing advertiser i can change the optimal blueprint, and the payment correctly internalizes that effect. DSIC follows from the VCG theorem (the mechanism selects an exact welfare maximizer),

while individual rationality follows because an advertiser can always report 0 and thereby guarantee non-negative utility under quasilinear preferences.

A practical implication of (9) is computational: computing all payments requires n additional global solves (one "leave-one-out" problem per advertiser), each of which is again $\text{poly}(n, m, |\Theta_\varepsilon|)$ under MNL. This is not conceptually problematic—it is the standard price of VCG—and it is often operationally acceptable because the solves are embarrassingly parallel across advertisers and across θ.

Revenue maximization: Myerson virtual surplus and envelope payments. For revenue, we move to the Myerson objective under the usual regularity condition that each $\phi_i(\cdot)$ is weakly increasing. The mechanism selects (θ^*, x^*) maximizing virtual surplus,

$$\sum_{i=1}^{n} \phi_i(v_i)\, \pi_i(\theta, c; x) - \kappa(\theta, c),$$

again over $\Theta_\varepsilon \times X$. Exact optimization implies the induced allocation rule is monotone in each v_i (holding v_{-i} fixed), which is the implementability condition in single-parameter environments. Thus, there exists a payment rule achieving DSIC and IR in principle, and it can be written via the envelope formula. Let

$$y_i(v_i, v_{-i}) := \pi_i(\theta^*(v), c; x^*(v))$$

denote advertiser i's realized click probability under truthful reports v. Then the DSIC payment can be expressed as

$$t_i(v) = t_i(0, v_{-i}) + v_i\, y_i(v) - \int_0^{v_i} y_i(z, v_{-i})\, dz. \tag{10}$$

In this environment the integral is rarely available in closed form because $y_i(\cdot, v_{-i})$ is induced by an optimization that may switch between different matchings and different blueprints as v_i varies. The computational remedy is to approximate the integral numerically.

Discretization and ε'-DSIC. A simple implementation is to discretize values on a grid $\{0, \eta, 2\eta, \ldots, \bar{v}\}$ and approximate the integral in (10) by a Riemann sum:

$$\int_0^{v_i} y_i(z, v_{-i})\, dz \approx \eta \sum_{k=0}^{\lfloor v_i/\eta \rfloor - 1} y_i(k\eta, v_{-i}).$$

Each term $y_i(k\eta, v_{-i})$ is obtained by running the same exact global optimizer with bidder i's report set to $k\eta$. As $\eta \to 0$, the numerical approximation converges, and the resulting mechanism is ε'-DSIC with ε' controlled by

the discretization granularity (and the boundedness of $y_i \in [0,1]$). Importantly, the ε' term is conceptually separate from the ε-menu approximation: ε controls how close we are to the best *continuous* blueprint, whereas ε' controls how accurately we compute payments for the (exactly optimized) menu-based allocation rule.

Limitations and policy-facing interpretation. Two caveats are worth keeping in view. First, the efficiency statements rely on the MNL structure, which yields the linear-fractional objective and makes exact optimization compatible with polynomial time. When user models become order-sensitive or involve richer externalities, exact inner optimization may cease to be tractable, and the monotonicity that underpins payments becomes fragile. Second, even under MNL, the mechanism inherits whatever modeling error is present in $p_{ij}(\theta, c)$ (or $\rho_{ij}(\theta, c)$) and in $\kappa(\theta, c)$. From a governance perspective, the appeal of the menu-based approach is that it aligns with auditing: we can certify a finite set of templates, solve each exactly, and then apply canonical truthful mechanisms on top. The economic content is that, under MNL, the platform can simultaneously (i) optimize across an auditable blueprint menu, (ii) preserve exactness of allocation within each blueprint, and (iii) obtain incentive guarantees with only controlled, explicitly parameterized approximations coming from discretizing payments rather than from heuristic optimization.

3.7 Extensions to order-sensitive users (cascade): what can be salvaged, and when we need heuristics

The MNL formulation is attractive not because it is behaviorally perfect, but because it turns winner determination into an exact polynomial-time optimization problem, which is exactly the condition we need to invoke canonical incentive arguments. Once we move to an order-sensitive user model, the economic logic does not disappear, but the computational and incentive conclusions become more conditional: we can still obtain truthful mechanisms *if* we can optimize exactly over a suitably chosen outcome range, yet for general cascade interactions we should expect to rely on approximations, numerical methods, and carefully designed restrictions on what blueprints are allowed to do.

Cascade clicks and the source of hardness. Under the cascade model, the platform's ordering decision matters because early insertions reduce attention (or available click probability mass) for later ones. Using the notation from the global context, for an augmented allocation $(x, \sigma) \in X \times \Sigma$ we have

$$\pi_i(\theta, c; x, \sigma) = \sum_{j=1}^{m} x_{ij} p_{ij}(\theta, c) \prod_{j' : \sigma(j') < \sigma(j)} \Big(1 - \sum_{i'} x_{i'j'} p_{i'j'}(\theta, c)\Big).$$

If we write the realized sequence of shown ads in display order as $\{(i_t, j_t)\}_{t=1}^{T}$ (with $T \leq K$), then the virtual-surplus (or welfare) part of the objective can be written as

$$\sum_{t=1}^{T} b_{i_t}\, p_{i_t j_t}(\theta, c) \prod_{s<t} \big(1 - p_{i_s j_s}(\theta, c)\big),$$

which is a *nonlinear* function of the chosen set *and* its order. This is qualitatively different from MNL: there is no linear-fractional reformulation of the same simplicity, and the induced optimization problem resembles sequencing problems with assignment constraints and multiplicative externalities. In broad generality (arbitrary $p_{ij}(\theta, c)$ and free choice of σ), one should expect NP-hardness by reduction from well-known ordering/assignment problems: the platform is effectively choosing a permutation to manage negative externalities across positions, while simultaneously solving a matching problem.

This has an immediate mechanism-design consequence. Our earlier monotonicity logic continues to apply *if* the platform can solve

$$(\theta^*, x^*, \sigma^*) \in \arg \max_{\theta \in \Theta_\varepsilon, (x,\sigma) \in X \times \Sigma} \sum_i b_i \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c)$$

exactly with deterministic tie-breaking: exact maximization over a fixed feasible set yields a monotone allocation rule in each bid coordinate. The problem is therefore not conceptual; it is computational.

Tractable structure: when exact optimization is still plausible. There are several practically relevant restrictions under which cascade winner determination can be solved exactly (or nearly exactly) and thus remains compatible with DSIC.

First, if the blueprint fixes an *order rule* $\sigma = \sigma(\theta, c)$ independently of bids, then the platform only chooses $x \in X$. This does not remove the nonlinearity, but it eliminates one combinatorial dimension and often makes the remaining problem amenable to dynamic programming when K is small. For example, when K is a small constant (as is often the case in conversational interfaces where inserting many ads is infeasible), we can optimize over the top K insertion loci by enumerating which subset of positions is filled and running a DP over the resulting short sequence. The running time is still exponential in K, but polynomial in n, m, which can be acceptable when K is truly small and treated as a design constraint.

Second, if predicted CTRs are *separable* in advertiser and position, e.g.,

$$p_{ij}(\theta, c) = \alpha_j(\theta, c) \cdot q_i(\theta, c),$$

then the cascade objective takes a more structured form. In the simplest case with a fixed sequence of positions (say σ is the natural order), the ordering problem becomes closer to classic results on optimal sequencing

under multiplicative survival. Pairwise swap arguments imply that, holding positions fixed and considering two advertisers r and s in two successive opportunities with the same α, advertiser r should precede s if

$$b_r\, q_r + (1 - q_r)\, b_s\, q_s \;\geq\; b_s\, q_s + (1 - q_s)\, b_r\, q_r,$$

equivalently,

$$\frac{b_r\, q_r}{1 - q_r} \;\geq\; \frac{b_s\, q_s}{1 - q_s}.$$

Thus, an "adjusted value" index can determine the efficient order within a block. While real blueprints will rarely satisfy perfect separability, even approximate separability (capturing a strong position effect times an advertiser effect) can justify designing Θ_ε so that the inner optimization is much closer to sorting than to general integer programming.

The broader lesson is blueprint-facing: when we allow arbitrary order-sensitive interactions, we should not be surprised to lose exact tractability. If we want DSIC guarantees, we may need to *co-design* the blueprint menu so that each template induces an inner problem with known exact algorithms.

Maximal-in-range as the main salvage operation. When full optimization is hard, the cleanest way to preserve incentive guarantees is to restrict the platform to a range of outcomes over which it *can* optimize exactly. Concretely, fix a (possibly blueprint-dependent) subset $\mathcal{R} \subseteq X \times \Sigma$ of augmented allocations, and define the mechanism to pick

$$(\theta^*, x^*, \sigma^*) \in \arg \max_{\theta \in \Theta_\varepsilon, (x,\sigma) \in \mathcal{R}} \; \sum_i b_i \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c).$$

Because the mechanism is now an *exact* maximizer over a fixed range, it is maximal-in-range (MIR). For welfare, VCG payments on the restricted outcome space yield DSIC/IR *within that range*. For revenue (regular case), exact optimization still yields monotonicity, so envelope payments remain valid (with the same numerical-integration caveat as before). The cost is purely allocative: we only compete against the best outcome in $\Theta_\varepsilon \times \mathcal{R}$, not against the unrestricted optimum.

Designing $\mathcal{R}$ is therefore the substantive modeling choice. Examples that are natural in conversational settings include: (i) restricting to a small set of allowed insertion loci and a fixed narrative-consistent order; (ii) allowing only "one ad per segment" layouts so that interference is bounded; or (iii) permitting only a few canonical orderings σ per blueprint, chosen for readability and disclosure compliance. Each of these makes the platform's commitment more auditable, and simultaneously makes exact optimization more plausible.

Monotone surrogate objectives and calibrated weights. A second approach is to optimize a *surrogate* objective that is linear in x (hence easily optimized and compatible with VCG), while treating the cascade effect as a correction absorbed into predicted weights. One simple family is

$$\widehat{\mathrm{Obj}}(\theta, c; x) = \sum_{i,j} b_i \, w_j(\theta, c) \, p_{ij}(\theta, c) \, x_{ij} - \kappa(\theta, c),$$

where $w_j(\theta, c) \in [0, 1]$ is interpreted as an ex ante attention weight for position j under blueprint θ and context c. If w_j is fixed independently of bids, then maximizing $\widehat{\mathrm{Obj}}$ over X is a weighted matching problem, and the induced allocation rule is monotone under exact optimization. Payments can then be computed exactly (VCG) for welfare with respect to $\widehat{\mathrm{Obj}}$, yielding a truthful mechanism for the *proxy* environment.

What we gain is strong incentive compatibility and computational tractability; what we lose is exact alignment with the true cascade welfare. This loss can be bounded in regimes where cascade interference is modest (e.g., small K, small p_{ij}, or when earlier positions have limited effect on later ones). In practice, one can set $w_j(\theta, c)$ by calibration: estimate expected survival probabilities from historical data under the blueprint's typical fill pattern, and update them as the system learns. Economically, this is a controlled form of model misspecification: we preserve truthful bidding while accepting that the platform optimizes a smoothed approximation to user attention.

Where greedy and local search run into incentive problems. A natural impulse under cascade is to use greedy selection: add the next (i, j) that maximizes the current marginal gain

$$\Delta(i, j \mid \text{prefix}) = b_i \, p_{ij}(\theta, c) \prod_{\text{earlier } (i', j')} \big(1 - p_{i'j'}(\theta, c)\big),$$

and then continue. Greedy can be effective as a heuristic, but from a mechanism-design standpoint it is treacherous: because the "prefix" depends on bids, increasing b_i can reshuffle earlier choices and reduce i's eventual chance of being placed, violating monotonicity. The resulting allocation rule need not be implementable with any payment rule satisfying DSIC/IR. Similar warnings apply to local search, simulated annealing, and mixed-integer solvers with early stopping: they can be excellent optimizers, but absent additional structure they will not be monotone algorithms.

This is the practical boundary line. If we insist on strict DSIC, we should either (i) optimize exactly over a fixed range (MIR/MIDR-style thinking), or (ii) use an algorithm whose monotonicity can be proved for the induced allocation rule. Otherwise, we should be explicit that we are implementing an approximately incentive compatible system and quantify the deviation.

Numerical methods and ε-IC in the fully general case. When the platform's cascade WDP is genuinely hard, the realistic alternative is to accept approximation in the allocation and then reason about approximate incentives. One operational route is to use an approximate optimizer that returns an α-approximate solution to the virtual-surplus objective. Approximation *alone* does not imply ε-DSIC; the key object is still how far the induced allocation rule deviates from monotonicity. In practice, one can (a) monitor monotonicity empirically by perturbing bids and measuring allocation changes, and (b) smooth the optimization numerically (e.g., randomized tie-breaking, or regularization of the objective) to reduce discontinuous allocation switches that create profitable bid manipulations. These steps do not restore theorem-level DSIC, but they can bound incentives in a way that is meaningful for governance: we can report an empirical ε such that no bidder gains more than ε by unilateral deviation within a tested bid range.

The policy-facing interpretation is that order-sensitive user behavior forces an explicit tradeoff between allocative optimality and incentive guarantees. If we want auditable, incentive-aligned ad insertion into LLM responses, we should view "which cascade effects we model" and "which blueprint behaviors we permit" as joint design choices, not independent engineering details.

3.8 Practical considerations: estimating $p_{ij}(\theta, c)$, validating Lipschitz stability, blueprint parameterization, and auditing/disclosure constraints

The formal results above treat the primitives $\{p_{ij}(\theta, c)\}_{i,j}$ and $\kappa(\theta, c)$ as given and common knowledge. In a real LLM system, they are neither: they must be estimated from interaction data, stress-tested for robustness, and embedded in a blueprint parameterization that is simple enough to optimize and to audit. We can read the theory as a *design specification*: if we can (i) build stable prediction and cost models and (ii) restrict blueprints to a low-dimensional, well-behaved family, then the mechanism logic becomes operational rather than aspirational.

What exactly is $p_{ij}(\theta, c)$ in an LLM interface? In the model, $p_{ij}(\theta, c)$ is a *standalone* click-through probability: the probability advertiser i would be clicked if it were shown at candidate locus j under blueprint θ in context c, abstracting away from interference created by other inserted ads. In an LLM response, a "position" may correspond to qualitatively different insertion types (inline sentence, card after a paragraph, sidebar module, etc.), so j is best interpreted as a *candidate insertion locus* in a blueprint-specific rendering plan. Practically, it is useful to define

$$p_{ij}(\theta, c) \equiv \Pr(\text{click on } i \mid i \text{ is shown at locus } j \text{ under blueprint } \theta \text{ in context } c, \text{ no other ads shown}),$$

even if we never literally run "no other ads shown" for every observation. This definition clarifies that p_{ij} is a causal object, not merely a predictive one: it is the effect of showing i at j, holding fixed the user and conversational context.

Estimation and identification: from logs to causal CTRs. If we naïvely regress clicks on features of (i, j, θ, c) using historical data, we inherit selection bias because the platform historically chose (θ, x, σ) in a bid- and prediction-dependent way. The usual remedy is to log the platform's *propensity* to select each outcome and apply off-policy estimation. Concretely, let a denote the platform action (blueprint plus allocation/order), and let $\mu(a \mid c)$ be the logging policy. For any candidate policy π, we can estimate expected click outcomes using inverse propensity scoring (IPS) or doubly robust (DR) estimators. In the simplest IPS form for a scalar outcome Y,

$$\widehat{\mathbb{E}}_{\pi}[Y] = \frac{1}{T} \sum_{t=1}^{T} \frac{\mathbf{1}\{a_t = \pi(c_t)\}}{\mu(a_t \mid c_t)} Y_t,$$

and in practice we prefer DR estimators that combine a click model with propensity weighting to reduce variance. The key operational point is that identifying $p_{ij}(\theta, c)$ at fine granularity requires *exploration*: every (θ, j) pair that might be used by the mechanism must receive nontrivial probability under μ, at least within neighborhoods of contexts where it could be optimal. Otherwise, the menu restriction lemma is moot because the mechanism will compare blueprints whose CTRs are extrapolated rather than learned.

A second identification issue is that in LLM systems, the *content* surrounding an insertion is itself influenced by θ: disclosure phrasing, narrative tone, and placement can change user attention even holding i and j fixed. This is not a bug; it is precisely why θ is economically meaningful. But it implies that $p_{ij}(\theta, c)$ should be estimated in a way that respects blueprint-induced distribution shift. A practical compromise is to fit a model $p_{ij}(\theta, c) = g(\psi(i), \zeta(j), h_\theta(c))$ where $\psi(i)$ is an advertiser embedding, $\zeta(j)$ is a locus embedding, and $h_\theta(c)$ is a blueprint-conditioned context representation. This keeps statistical strength while preserving the causal interpretation under randomized blueprint assignment.

Estimation error and conservative optimization. Our theoretical objectives use point predictions. In deployment, uncertainty matters: if $\widehat{p}_{ij}$ is noisy, exact maximization of a plug-in objective can overfit to estimation error. A standard fix is to use lower confidence bounds,

$$\widetilde{p}_{ij}(\theta, c) = \widehat{p}_{ij}(\theta, c) - \beta\, \widehat{\mathrm{se}}_{ij}(\theta, c),$$

and optimize using $\widetilde{p}_{ij}$ (and similarly for κ). This converts winner determination into a risk-adjusted problem that is more stable and typically easier

to justify in audits, at the cost of some foregone surplus. The same idea can be applied to the Lipschitz constant: we can treat L as a *high-probability* sensitivity bound rather than a deterministic one.

Validating Lipschitz stability: what we can test and what we cannot. The Lipschitz hypothesis is the mathematical hinge behind finite menus: small changes in θ should not cause large changes in π_i or κ. In an LLM system, there are two distinct failure modes.

(i) Behavioral discontinuities. Small changes in disclosure wording, layout spacing, or "ad density" can trigger sharp shifts in user trust and attention. These are genuine non-smooth preferences, not merely modeling artifacts.

(ii) Rendering discontinuities. A blueprint parameter may flip a discrete switch (e.g., enabling a new insertion type), changing the feasible set of loci $J(\theta, c)$ and thus altering π and κ even if the user response function is smooth.

We cannot prove away these discontinuities; we can only *engineer* against them. A workable approach is to treat Lipschitzness as an empirical contract: choose a parameterization in which small perturbations of θ correspond to small, localized rendering changes, and then validate with randomized experiments. For example, for sampled contexts c and small perturbations δ, we can estimate local sensitivities

$$\widehat{S}(\theta, c; \delta) = \max_{(x,\sigma)\in\mathcal{A}(c)} \max_{i\in[n]} \frac{\left|\widehat{\pi}_i(\theta+\delta, c; x, \sigma) - \widehat{\pi}_i(\theta, c; x, \sigma)\right|}{\|\delta\|_2},$$

where $\mathcal{A}(c)$ is a test set of representative allocations/orders. If $\widehat{S}$ is frequently large, we should not respond by inflating L and accepting a huge menu; we should instead redesign Θ (or accept that the outer problem is intrinsically discrete, as discussed earlier).

A practical trick that often restores approximate smoothness is *generation smoothing*: randomize over a small set of equivalent renderings (e.g., paraphrases of disclosure text) with fixed mixture weights, so that the mapping $\theta \mapsto \pi$ becomes an average over variants and hence less discontinuous. This improves stability but raises governance questions, since randomness must be logged and reproducible for audits.

Blueprint parameterization: keeping d small and interpretable. The menu restriction bound is only meaningful when d is genuinely small. This pushes us toward a blueprint representation that captures the main economic levers without encoding the full generative policy of the LLM. In practice, we want θ to be a vector of a few interpretable "knobs" that product teams already reason about, such as:

- *Ad density:* expected K or a soft penalty for additional insertions;

- *Prominence:* visual size, distance from the user's requested content, or salience of the module;
- *Disclosure strength:* label wording, prefixing, or explicit separation from organic content;
- *Narrative integration:* whether ads are presented as cards, inline suggestions, or appended recommendations;
- *Eligibility filters:* sensitive-topic exclusions, advertiser-category constraints, or jurisdictional rules.

We emphasize that interpretability is not merely aesthetic: it is what makes $\kappa(\theta, c)$ defensible and what makes commitment credible. If θ is an opaque embedding controlling many latent behaviors, then even if the mechanism is DSIC in theory, it will be difficult to persuade advertisers or regulators that the platform is not implicitly conditioning on bids through hidden channels.

Modeling and measuring $\kappa(\theta, c)$: from norms to welfare units. The cost $\kappa(\theta, c)$ is where trust, compliance, and long-run platform value enter the formal objective. The main practical difficulty is units: clicks and advertiser values are monetizable, while trust and compliance are not directly. A typical implementation treats κ as a weighted sum of measurable proxies,

$$\kappa(\theta, c) = \lambda_{\text{discl}} \cdot D(\theta, c) + \lambda_{\text{sat}} \cdot S(\theta, c) + \lambda_{\text{risk}} \cdot \text{Risk}(\theta, c),$$

where D might capture disclosure weakness (lower is better), S might capture predicted user dissatisfaction or abandonment, and Risk might capture policy-violation probability. The economics here is straightforward but important: these weights λ are policy parameters. They encode the platform's willingness to trade revenue for trust and regulatory safety, and they are therefore natural objects for internal governance (and, potentially, for regulatory scrutiny). This is also where comparative statics in the next section become actionable.

Auditing, disclosure constraints, and "mechanism compliance" as invariants. Because blueprints modify an LLM response, we should treat disclosure and separation rules as *hard constraints* whenever feasible, not as soft penalties. Concretely, let $\mathcal{F}(c) \subseteq \Theta \times X \times \Sigma$ denote outcomes that satisfy invariants such as: disclosure text present; label font size above a threshold; ad modules separated from organic content; no ads in sensitive conversational segments; and no advertiser shown more than once. Then winner determination should be solved over $\mathcal{F}(c)$, not merely penalized by κ. This has two benefits: (i) it simplifies auditing, because compliance reduces to verifying membership in $\mathcal{F}(c)$; and (ii) it prevents pathological optima

where the platform "pays" for non-compliance in κ but still chooses it when bids are high.

Auditability also requires *replay*: given logged (c, b), the platform should be able to reconstruct $(\theta^*, x^*, \sigma^*)$ and payments. Deterministic tie-breaking is not just a proof convenience; it is what makes the allocation rule well-defined and dispute-resilient. If any stochasticity is used (e.g., smoothing), its random seeds and distributions must be logged as part of the mechanism state.

Payments in practice: discretization, monotonicity checks, and numerical robustness. Even when the allocation is monotone in theory (under exact optimization), payments can be fragile numerically. For welfare/VCG, the issue is computational: we must compute counterfactual optima with advertiser i removed, which multiplies runtime by n (times $|\Theta_\varepsilon|$). For envelope-based payments (virtual surplus), the issue is approximation: we discretize the integral

$$t_i(v) = t_i(0, v_{-i}) + v_i y_i(v) - \int_0^{v_i} y_i(z, v_{-i})\, dz,$$

and any discretization error becomes an incentive issue. A practical safeguard is to pair the payment computation with *monotonicity regression* (or ironing) on the empirically estimated allocation curve $y_i(\cdot, v_{-i})$, ensuring the numerical implementation respects weak monotonicity even when solver tolerances or estimation noise would otherwise induce small violations.

Finally, we emphasize a limitation that is easy to miss: if the system uses early-stopped mixed-integer solvers, heuristic search, or non-deterministic caching in production, then "exact optimization" may fail silently, and with it the DSIC guarantee. Operationally, this argues for mechanism implementations that are simple enough to be solved to certified optimality (or optimized over a restricted range), rather than relying on best-effort solvers whose failure modes are hard to audit.

Putting it together: a deployable workflow. A coherent engineering interpretation of the model is a loop: (i) design a low-dimensional, interpretable Θ and a compliant feasible set $\mathcal{F}(c)$; (ii) explore to estimate $p_{ij}(\theta, c)$ and $\kappa(\theta, c)$ with logged propensities; (iii) empirically validate local stability and redesign Θ if discontinuities are common; (iv) construct a finite menu Θ_ε aligned with the validated smoothness; and (v) run exact (or certified) optimization with deterministic tie-breaking and auditable payment computation. This workflow makes clear why the next section's comparative statics are not merely theoretical: the policy parameters embedded in κ and the granularity ε directly govern the platform's observable behavior, and hence the practical tradeoff between monetization and trust.

3.9 Comparative statics and policy: the price of disclosure intensity, welfare–trust tradeoffs, and transparency regulation

A useful feature of the blueprint-augmented mechanism is that it makes several policy-relevant tradeoffs explicit. The platform is not merely choosing a matching x (and possibly an order σ); it is also choosing θ, which governs the disclosure style, prominence, density, and other presentation choices that affect both user behavior $\pi_i(\theta, c; x, \sigma)$ and the non-revenue cost $\kappa(\theta, c)$. This section uses the objective as a lens for comparative statics: how does the chosen blueprint move when we change (i) the weight placed on trust/compliance costs, (ii) the granularity of the allowable blueprint menu, and (iii) the degree of user attention scarcity (outside-option strength)? We then connect these movements to questions of transparency regulation and platform commitment.

A parametric cost of disruption and its envelope implications. To sharpen the discussion, it is convenient to separate a *measurable disruption index* from its welfare weight. Let

$$\kappa(\theta, c) = \lambda \cdot D(\theta, c),$$

where $D(\theta, c) \geq 0$ aggregates factors such as disclosure weakness, visual intrusion, policy-risk, or predicted dissatisfaction, and $\lambda \geq 0$ is a governance parameter translating disruption into welfare units. For a fixed context c and bid vector b, define the optimized value

$$V(\lambda; c, b) = \max_{\theta \in \Theta_\varepsilon} \max_{(x,\sigma) \in X \times \Sigma} \Big\{ \sum_{i=1}^{n} b_i \, \pi_i(\theta, c; x, \sigma) - \lambda D(\theta, c) \Big\}.$$

Two general facts follow from standard envelope reasoning. First, $V(\lambda; c, b)$ is weakly decreasing and convex in λ (a pointwise supremum of affine functions of λ). Second, whenever the maximizer $(\theta^*_\lambda, x^*_\lambda, \sigma^*_\lambda)$ is essentially unique (or more generally, selecting a measurable maximizer with deterministic tie-breaking), the right and left derivatives satisfy

$$\frac{\partial V}{\partial \lambda^+}(\lambda; c, b) = -D(\theta^*_\lambda, c), \qquad \frac{\partial V}{\partial \lambda^-}(\lambda; c, b) = -D(\theta^*_{\lambda^-}, c),$$

so the realized disruption at the chosen blueprint is a (sub)gradient of the platform's value with respect to the trust/compliance weight. Operationally, this turns λ into a dial: increasing λ monotonically increases the platform's willingness to sacrifice click surplus in order to reduce disruption. The comparative static is transparent because the cost is separable in λ.

This representation also suggests a principled way to interpret internal debates about "how strict disclosure should be." If we calibrate λ so that a

unit reduction in D corresponds to an empirically estimated long-run benefit (retention, complaint reduction, enforcement risk reduction), then the mechanism's chosen blueprint is the *efficient* one under that valuation. If instead λ reflects only short-run or private costs to the platform, then the resulting blueprint is privately optimal but may under-provide disclosure relative to a social optimum.

The "price of disclosure intensity." Many governance discussions are phrased in terms of constraints on disclosure strength rather than a continuous penalty. We can connect these views by introducing a scalar disclosure parameter s embedded in θ, where higher s means more explicit labeling or stronger separation from organic content. Suppose $\theta = (s, \eta)$, where η collects other blueprint knobs. A regulator who strengthens disclosure effectively imposes a lower bound $s \geq \underline{s}$, shrinking the feasible set from Θ_ε to $\Theta_\varepsilon(\underline{s}) = \{\theta \in \Theta_\varepsilon : s(\theta) \geq \underline{s}\}$. Let

$$V(\underline{s}; c, b) = \max_{\theta \in \Theta_\varepsilon(\underline{s})} \max_{(x,\sigma) \in X \times \Sigma} \Big\{ \sum_i b_i \, \pi_i(\theta, c; x, \sigma) - \kappa(\theta, c) \Big\}.$$

Then $V(\underline{s}; c, b)$ is weakly decreasing in $\underline{s}$, and the welfare impact of raising disclosure standards is precisely the loss in optimized objective from restricting the blueprint set. This loss is a direct analogue of the "price" of disclosure intensity: it is the opportunity cost in foregone virtual surplus (or welfare, depending on whether $b_i = v_i$ or $b_i = \phi_i(v_i)$) needed to meet the stricter disclosure requirement. In contexts where users strongly discount ads once disclosed, the slope of V in $\underline{s}$ will be steep; in contexts where disclosure does not materially reduce attention (or improves trust enough to offset attention loss), the slope may be small or even effectively zero over relevant ranges.

One can also define a local price when disclosure enters linearly into the disruption index, e.g., $D(\theta, c) = D_0(\eta, c) + \alpha(c) \cdot s$, with $\alpha(c) > 0$. Then increasing λ raises the shadow price of s: holding other dimensions fixed, the mechanism trades off an incremental change in click surplus against $\lambda\alpha(c)$. Even when s is not separable, the envelope relation $\partial V / \partial \lambda = -D(\theta^*_\lambda, c)$ implies that stronger disclosure (to the extent it reduces D) is chosen precisely when the induced click-loss is dominated by the weighted reduction in disruption.

Welfare versus revenue objectives: where the tradeoff bites. The welfare mechanism (with $b_i = v_i$) and the revenue mechanism (with $b_i = \phi_i(v_i)$ under regularity) can select different blueprints even holding λ fixed. Intuitively, virtual values reweight clicks toward advertisers on the steep part of the revenue curve, so revenue optimization is more willing to sacrifice low-virtual-value clicks for a modest increase in high-virtual-value clicks. This matters for disclosure because disclosure changes the *composition* of

clicks: it can disproportionately reduce clicks on low-quality or low-relevance ads (which users would have clicked under confusion) while leaving high-relevance ads relatively intact. In such cases, stronger disclosure can lower total click volume but improve welfare (by reducing misclicks) and might even increase revenue if it shifts attention toward high virtual value advertisers. Conversely, if disclosure reduces attention uniformly, then both welfare and revenue objectives will typically move toward less intrusive blueprints as λ increases, but revenue may remain relatively more aggressive at a given λ because it places higher marginal value on the remaining clicks.

This observation has a policy corollary: if regulators care about allocative efficiency and consumer protection, it is not enough to ask whether disclosure reduces revenue. The relevant question is how disclosure changes $\pi_i(\theta, c; x, \sigma)$ across advertisers and positions—that is, whether it removes primarily low-value, potentially misleading engagement or whether it suppresses high-value matches. The blueprint framework forces this heterogeneity into the primitives $p_{ij}(\theta, c)$ (and hence π_i) rather than treating disclosure as an undifferentiated tax.

Menu granularity ε as a governance tool, not only a computational knob. The ε-menu restriction is usually motivated by tractability: smaller ε yields a denser menu Θ_ε and hence better approximation but higher runtime. There is also a governance interpretation. A very fine menu allows the platform to micro-optimize θ to each context c, potentially producing blueprints that are hard to describe, audit, or explain, even if each knob is nominally interpretable. A coarser menu (larger ε) forces the platform to choose among a small number of discrete, documented templates. The additive approximation loss can then be interpreted as the cost of *standardization* and *predictability*.

This is especially salient for transparency regulation. If the platform publicly commits to a finite menu and deterministic tie-breaking, then third parties can (at least in principle) replicate decisions ex post. The menu restriction lemma provides an economic argument for why such commitment need not be too costly when the system is stable in θ: the welfare/revenue loss scales with $\varepsilon(1 + n\bar{v})$. Conversely, if achieving high revenue requires an extremely fine menu, that is indirect evidence that the blueprint space is effectively high-dimensional or unstable, and thus that any claim of simple, auditable ad presentation is likely to be fragile.

Dimensionality d, interpretability, and the limits of "transparent optimization." The covering-number bound makes explicit the curse of dimensionality: $|\Theta_\varepsilon|$ grows on the order of ε^{-d}. Practically, this is not merely a computational warning; it is a transparency warning. A high-dimensional θ makes it difficult to articulate what the platform is committing

to. Even if the mechanism is DSIC under exact optimization over Θ_ε, a large and complex menu can undermine perceived fairness and can frustrate auditing, because small undisclosed changes in θ may be hard to detect yet economically meaningful.

From a policy standpoint, this suggests a substantive interpretation of "meaningful disclosure of advertising practices": it is not only the presence of an "Ad" label, but also the restriction of the platform's policy space to a manageable number of verifiable blueprint variants. In that sense, limiting d (or committing to a coarse Θ_ε) is a form of *mechanism transparency*. It reduces the degrees of freedom through which the platform could implicitly tailor user experience in bid-dependent ways, even when the formal mechanism is bid-monotone.

Outside-option strength and endogenous ad load. Under the MNL specification, the outside option (the baseline propensity to not click anything) enters through the denominator. When the outside option is strong, incremental changes in attractiveness $\exp(\rho_{ij}(\theta, c))$ have smaller effects on click probabilities, and the platform's objective exhibits sharper diminishing returns to filling additional loci. In such regimes the optimized solution often chooses fewer than K ads even when K is allowed: leaving positions empty can be optimal because additional ads mostly cannibalize attention rather than expand total clicks.

This interacts with disclosure in a subtle way. Stronger disclosure can be modeled either as (i) reducing ρ_{ij} for ads (users treat them as less attractive), or (ii) increasing the outside option (users prefer to continue reading without clicking). Both channels push toward concentration on the highest-quality matches and away from aggressive ad density. Thus, when user attention is scarce, the marginal revenue benefit of weaker disclosure is often small, while the trust/compliance benefit (via reduced $D(\theta, c)$) may remain substantial. The framework therefore predicts a pattern that aligns with practice: in high-stakes or high-friction contexts (e.g., sensitive topics, complex multi-turn tasks), optimal blueprints should be conservative—fewer insertions and stronger disclosure—even absent hard constraints.

Implications for transparency regulation and credible commitment. We can view transparency regulation as acting on three distinct layers of the model. First, it can impose *hard feasibility constraints* on $\mathcal{F}(c)$, such as mandatory disclosure text or forbidden insertion loci. Second, it can effectively increase λ by raising expected penalties for disruptive behavior, converting external enforcement into an internalized cost. Third, it can require *commitment and auditability*: public documentation of the menu Θ_ε, deterministic tie-breaking, and reproducible logging sufficient to reconstruct $(\theta^*, x^*, \sigma^*)$ and payments.

The blueprint mechanism clarifies why commitment matters. Even if the platform promises to be "transparent," advertisers and users may worry about time-varying or context-dependent presentation that is hard to observe. A committed finite menu is a concrete object that can be audited: the regulator can test whether the deployed blueprint is one of the declared $\theta \in \Theta_\varepsilon$ and whether the declared disclosure parameters match what is rendered. Moreover, because the winner determination is an explicit maximization over a fixed feasible set, deviations (e.g., switching to an unannounced blueprint when bids spike) are, in principle, detectable.

At the same time, the model highlights a limitation: regulation that only mandates disclosure text but leaves the blueprint space otherwise unconstrained may not achieve robust transparency if other dimensions of θ can subtly shift salience or placement. In our language, that is regulation targeting a single coordinate s while allowing the platform to optimize freely over η. Effective policy likely requires either broader constraints on Θ (e.g., restrictions on loci or prominence) or an explicit accounting of those dimensions in $D(\theta, c)$ with sufficiently large λ.

Summary: comparative statics as a design and policy checklist. The comparative statics are not merely qualitative. They tell us what to measure and what to govern. Estimating how π_i and D move with disclosure, prominence, and density directly identifies the "price" of stronger transparency. Choosing λ (or imposing constraints on Θ) determines how aggressively the platform monetizes attention in the face of trust and compliance costs. Finally, ε and d determine whether commitment to an auditable menu is feasible without large surplus loss. These are precisely the levers that product teams and regulators can manipulate; the model's contribution is to place them in a single optimization problem where the tradeoffs are legible and, when stability holds, computationally implementable.

3.10 Conclusion and open problems: beyond Lipschitz, richer advertiser types, and dynamic blueprint choice

We have treated "blueprints" as a low-dimensional, economically meaningful policy instrument: a vector of design choices that shapes both the mapping from bids to click probabilities and the platform-side cost of disruption, trust loss, or compliance risk. The key modeling move is to bring blueprint choice *inside* the winner determination problem, so that the platform optimizes jointly over θ and the augmented allocation (x, σ). Once we do so, familiar mechanism-design logic largely survives: with exact optimization over a fixed feasible set, monotonicity in bids is preserved, and DSIC/IR mechanisms are available (VCG for welfare; Myerson-style virtual surplus with envelope payments under regularity). The ε-menu restriction lemma then gives a concrete tractability story: if the system is stable in θ, a finite, auditable

menu can approximate the best continuous blueprint with a transparent additive loss.

At the same time, the analysis makes clear where the fragility lies. Our clean approximation and implementability results rely on a strong smoothness hypothesis: Lipschitz stability of $\pi_i(\theta, c; x, \sigma)$ and $\kappa(\theta, c)$ in θ, uniformly over contexts and allocations. This assumption is plausible when θ modulates *intensities* (e.g., disclosure strength, prominence, density) within a fixed layout family, and when user response varies continuously with those intensities. But real systems often exhibit step changes: a small change in a template parameter can create or eliminate an insertion locus, flip a disclosure label from "below the fold" to "above the fold," trigger a policy classifier, or switch generation into a qualitatively different style. These discontinuities are not a technical nuisance; they are a substantive feature of LLM-mediated presentation. Understanding mechanism design in the presence of such non-smooth blueprint effects is, in our view, the first open problem.

Beyond Lipschitz: non-smooth blueprints, discrete families, and smoothing. When θ changes the feasible position set $J(\theta, c)$ or the rendering grammar, the map $\theta \mapsto \pi(\theta, c; x, \sigma)$ may fail to be continuous, and the covering argument behind Θ_ε can collapse. There are several directions one might pursue. A conservative approach is to abandon the pretense of continuity and treat blueprint choice as a *discrete* design problem: a hand-curated family $\{\theta^1, \ldots, \theta^M\}$ justified by product constraints, policy requirements, and auditability. The outer optimization is then simply a finite maximization, and the main theoretical task becomes to characterize how large M must be to compete with richer, less interpretable policies.

A more structural approach is to recover a form of stability via *randomization or smoothing*. For instance, one could allow the platform to sample a blueprint θ from a distribution q over a finite set and evaluate expected clicks and costs. If π_i and κ are discontinuous pointwise but well-behaved in expectation under small perturbations, then the relevant object becomes

$$\bar{\pi}_i(q, c; x, \sigma) = \mathbb{E}_{\theta \sim q}[\pi_i(\theta, c; x, \sigma)], \qquad \bar{\kappa}(q, c) = \mathbb{E}_{\theta \sim q}[\kappa(\theta, c)],$$

which may admit Lipschitz-like control in an appropriate metric on distributions q (e.g., total variation or Wasserstein). This suggests an engineering interpretation: injecting controlled randomness into rendering (within acceptable UX bounds) can make optimization more stable and, paradoxically, more amenable to transparent approximation. A central open question is whether such smoothing can be done while retaining incentive guarantees (truthfulness is delicate when allocations are randomized) and while respecting regulatory demands that disclosures be consistent rather than stochastic.

Approximate optimization and the re-emergence of incentives. Even under smoothness, DSIC in our discussion leans on *exact* maximization (with

deterministic tie-breaking). In practice, platforms use heuristics, approximate solvers, and learned ranking models. Approximation can break monotonicity and hence invite strategic bidding. Designing algorithms that are simultaneously (i) computationally efficient, (ii) approximately optimal over $\Theta_\varepsilon \times X \times \Sigma$, and (iii) approximately truthful with *quantifiable* incentive loss remains an important open problem. One promising route is to combine menu restriction with *monotone algorithm design*: restrict attention to algorithmic families that are monotone by construction, even if they are not globally optimal, and then bound the welfare/revenue gap relative to the true optimum. Another route is to characterize conditions under which small optimization errors imply small deviations from monotonicity, thereby yielding explicit ε-DSIC bounds tied to solver tolerances. The difficulty is that monotonicity is a global property in bids, while optimization error is typically local.

Learning and uncertainty: when $p_{ij}(\theta, c)$ is estimated, not known. Our primitives treat predicted CTRs $p_{ij}(\theta, c)$ (or ρ_{ij}) and costs $\kappa(\theta, c)$ as given. In deployed systems, these are estimated with error, subject to nonstationarity, and potentially manipulable. This raises both statistical and strategic questions. Statistically, the platform's objective is computed using $\widehat{\pi}_i$ and $\widehat{\kappa}$, but welfare and payments depend on realized clicks; the gap between these two can produce systematic distortions in blueprint choice (e.g., underestimating the trust penalty pushes toward aggressive θ). Strategically, advertisers may attempt to influence predictions via creatives or landing pages, effectively making the click model endogenous.

A natural research direction is *robust blueprint optimization*: replace the point estimate with an uncertainty set $\mathcal{U}(\theta, c)$ and maximize worst-case or risk-adjusted value, e.g.,

$$\max_{\theta,(x,\sigma)} \min_{(\pi,\kappa)\in\mathcal{U}(\theta,c)} \Big\{ \sum_i b_i \pi_i - \kappa \Big\}.$$

How this interacts with truthfulness is not obvious: robust objectives can introduce non-linearities in bids that complicate monotonicity. Another direction is to integrate exploration directly into the mechanism, so that the platform learns $\pi(\theta, c; \cdot)$ while running auctions. Here one confronts a classic tension: exploration changes allocations, which changes incentives, which changes the data-generating process. Understanding what forms of online learning preserve approximate DSIC (or at least BIC) in the blueprint-augmented setting is largely open.

Multi-parameter advertisers and the limits of single-parameter reductions. We have assumed single-parameter values per click. Many advertising environments are not: advertisers can have values that depend

on conversion quality, user type, position externalities, frequency, or cross-campaign interactions; they also face budgets and ROI constraints. Once types become multi-dimensional, DSIC mechanisms require *cyclic monotonicity* (Rochet), and payment formulas are no longer one-dimensional envelopes. The blueprint layer makes this harder because θ can change the mapping from type reports to outcomes in a high-dimensional way.

A pragmatic question is whether blueprint choice can be separated from multi-dimensional private information by imposing structure. For example, if advertisers submit a scalar bid but the platform predicts heterogeneous conversion rates, one can interpret the bid as a value per predicted conversion and treat the rest as public signals; this restores a single-parameter form but invites misreporting if advertisers can influence signals. Alternatively, one could aim for Bayesian incentive compatibility with distributional assumptions and design mechanisms that are approximately optimal and approximately truthful. The open theoretical problem is to identify economically reasonable conditions under which blueprint-augmented allocation remains implementable with tractable payment rules when advertiser preferences are richer than per-click values.

Dynamic, multi-turn blueprint choice: state dependence, commitment, and long-run welfare. Our model is static: c is realized, the platform chooses θ and (x, σ), clicks happen, and the interaction ends. LLM-mediated products are inherently dynamic. The conversation evolves, user trust accumulates or decays, and ad exposure today affects engagement tomorrow. A natural extension is to treat the interaction as a controlled Markov process with state s_t (including conversation history, inferred intent, and trust proxies), action $(\theta_t, x_t, \sigma_t)$, and per-period payoff

$$\sum_i v_i \pi_i(\theta_t, s_t; x_t, \sigma_t) - \kappa(\theta_t, s_t),$$

aggregated with discount $\delta \in (0, 1)$. Blueprint choice then becomes an intertemporal policy, and the temptation to exploit users in the short run (weak disclosure, high density) must be balanced against future state deterioration (reduced retention, higher complaint probability, tighter regulatory scrutiny).

Dynamic mechanism design in such environments poses two intertwined open problems. First, even without private information, optimal control with large state spaces forces approximation; the resulting policy may again violate monotonicity in bids. Second, with private advertiser values (and possibly private user signals), truthfulness over time becomes subtle: bidders may misreport early to influence future blueprint policies. Classic tools (dynamic VCG, bank-account mechanisms) may apply in principle, but they require structure that is not obviously present when the platform's action

includes presentation and disclosure choices. A central question is what *commitment* means here: is the platform committing to a stationary mapping $s \mapsto \theta(s)$, to a finite menu with fixed switching rules, or merely to audit logs? Each notion of commitment has different welfare and policy implications.

Externalities, fairness, and multi-stakeholder costs. The cost term $\kappa(\theta, c)$ compactly represents trust/compliance. In reality, costs are multi-stakeholder: users experience annoyance or deception, publishers and creators experience crowd-out, and society bears broader harms (misinformation amplification, discriminatory targeting). Representing these as a single scalar is analytically convenient but normatively incomplete. One open direction is to model κ as a vector of costs and study constrained optimization (e.g., minimize disruption subject to revenue floors, or maximize welfare subject to regulatory constraints), which may better match how governance is implemented. Another direction is to incorporate group-level constraints on exposure or outcomes, which interact with blueprint choice in non-trivial ways because disclosure and prominence can have heterogeneous effects across user groups. The challenge is to preserve tractability and incentive properties while respecting such constraints.

Auditing, reproducibility, and the boundary between mechanism design and governance. Finally, blueprint augmentation suggests a concrete interface between theory and practice: a platform can publish a menu Θ_ε, define $D(\theta, c)$ (or at least measurable proxies), and log the chosen θ and allocation. Yet auditing an LLM-mediated rendering pipeline is difficult: what exactly counts as the blueprint, and how do we verify it was followed? This raises an open problem that sits at the boundary of economics and systems: designing *verifiable blueprint commitments* that are expressive enough for product needs but constrained enough to be audited, and that interact cleanly with the auction logic (in particular, preventing bid-dependent, unlogged deviations in θ). Menu restriction provides an economic rationale for standardization; realizing that promise requires cryptographic, procedural, or regulatory mechanisms that bind the platform to the declared menu.

Taken together, these open problems point to a common theme. The blueprint formalism is valuable precisely because it forces us to treat presentation as a choice variable with welfare consequences. But once we acknowledge that presentation is complex, learned, and dynamic, the clean DSIC story becomes a benchmark rather than a full description. Our hope is that the framework serves as a map: it identifies which assumptions buy tractability and truthfulness, what is lost when those assumptions fail, and where future work can most productively connect mechanism design, learning, and governance in the LLM era.