MIRT: TRANSFORMERS FOR TRUTHFUL GENERATIVE AUCTIONS WITH WHOLE-FEED PERMUTATION EXTERNALITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern online platforms commonly rank ads and organic content separately before blending them into a feed displayed to the user, overlooking externalities: an item's click-through rate depends on its surrounding content, not only on its own position. Recent learning-based feed generation mechanisms model some of these cross-type interactions to globally optimize for the whole feed's welfare. However, these approaches either fix the ordering of organic content, or lack exact strategyproofness guarantees for bidders. To combat these shortfalls, we introduce the *Maximal-in-Range Transformer (MIRT)* mechanism class, which uses a transformer to generate a range of candidate feeds that jointly order ads and organic content, selects the welfare-maximizing feed in the range. However, there is a tension: strategyproofness requires the generated range to be bid-independent, even though a candidate feed's welfare depends linearly on the bids. Our key technical contribution is a reinforcement learning approach that incorporates both candidate generation and bid-aware selection into training, enabling a *bid-independent* transformer to learn to generate high-welfare ranges by accounting for both individual feed quality and the collective quality of the range. Additionally, we bound the pseudo-dimension of the MIRT class under hard attention, showing that near-optimal expected welfare is learnable with sample complexity polynomial in the transformer size and only logarithmic in the range size. Empirically, MIRT outperforms the previous non-strategyproof state-of-the-art feed models while remaining exactly strategyproof. Our results show that transformer-based auctions can deliver externality-aware whole-feed optimization without sacrificing exact incentive compatibility, removing a major obstacle to their practical deployment.

1 INTRODUCTION

In modern feed-based ranking systems, organic content and sponsored ads are commonly ranked in independent pipelines and then blended into a single feed [8; 27; 46]. This architecture is tractable because it treats a post or ad's click-through-rate (CTR) as primarily a function of its own identity and position relative to other items of the same type. In practice, however, user engagement also depends on the surrounding content. For instance, a hiking post from a close friend may increase the *click-through rate (CTR)* of a nearby outdoor-equipment ad, while several similar ads may instead create fatigue and reduce CTR. This effect is referred to as *permutation externalities*: the CTR of an ad slot or organic post depends not only on its position but also on the surrounding content. When these externalities are significant, this tractability comes at a toll on of total welfare. Motivated by this, we focus on the problem of training transformer models for jointly ranking organic and ad content, taking into account permutation externalities.

However, in the advertising context, the goal is not merely to choose a good ranking but also to run an auction. Joint ranking makes it more difficult to enforce *incentive compatibility* among advertisers because an advertiser may benefit from strategic misreporting to influence the content surrounding its ad. Reusing the aforementioned example of an outdoor equipment advertisement: at a low bid, a joint feed aggregator might place a hiking post immediately before the ad, generating a high CTR for the advertiser; after the advertiser raises its bid, the aggregator might promote the ad above and

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

far from the hiking post, so the CTR of the ad decreases because it loses the favorable surrounding context despite receiving a higher slot.

More formally, a necessary condition for *dominant-strategy incentive compatibility (DSIC)*, also known as *strategyproofness*, is that realized advertiser benefits (e.g., CTR times click value) is nondecreasing in the advertiser's bid, but an unconstrained joint ranker need not satisfy this condition. For many mechanism design problems, the techniques of differentiable economics [16] have been used to design state-of-the-art strategyproof mechanisms by training neural networks. For feed generation in particular, recent methods such as CGA [47] and NGA [46] build on this approach. They train transformer models to choose rankings that take into account permutation externalities, and appropriate transfers that minimize incentive compatibility violations. However, these methods can only approximate strategyproofness. In fact, the strategyproofness gap is substantial in practice: in our experiments, we find possible gains from misreporting of more than 10% of advertiser utility (Section 5). To the best of our knowledge, no prior work jointly optimizes the entire ad-organic ordering under permutation externalities while guaranteeing exact DSIC.

To directly address incentive issues, we introduce *Maximal-in-Range Transformers (MIRT)*, a framework that combines the expressive power of transformer-based ranking for welfare-maximization under permutation externalities, with a maximal-in-range [33] allocation rule that ensures DSIC. In particular, MIRT uses a transformer to generate a set of candidate ad-organic joint rankings known as the "range", conditioned on ad and organic content, user features, and other constraints, but independent of advertiser bids. At serving time, a maximal-in-range mechanism observes the bids, selects the candidate among the range that maximizes reported welfare, and charges VCG transfers over the range. Using RL, we train the transformer to generate a small *set* of feeds that capture contextual interactions among ads and organic content, accounting for both individual feed quality and the collective quality of the range.

Our Contributions. In Section 2, we present a general model of joint ad-organic optimization. Prior work either considers permutation externalities only among ads, or models ad-organic interactions while taking the organic ordering as fixed input. In Section 3 we introduce the MIRT mechanism class for exactly strategyproof ad-organic joint optimization under permutation externalities. MIRT uses a maximal-in-range approach over a transformer-generated, bid-independent range to ensure strategyproofness. The transformer is trained via RL to generate high-welfare ranges by accounting for both individual feed quality and the collective quality of the range. To our knowledge, MIRT is the first transformer-based approach to feed optimization that provides a strategyproofness guarantee. In Section 4, we provide pseudo-dimension bounds for the welfare class induced by MIRT, and show that the sample complexity of learning a MIRT mechanism with optimal expected welfare within the mechanism class is polynomial in the transformer size and only logarithmic in the range size. Finally, in Section 5 we evaluate MIRT in synthetic environments, comparing against the previous state-of-the-art approaches on welfare, virtual welfare, and DSIC violations. and studying the effects of range size and externality strength. MIRT attains better welfare than previous SOTA across all tested environments, while also being DSIC by construction. By comparison, the previous SOTA approaches have expected regret under truthful bidding as high as 10% of advertiser utility.

1.1 Related Work

Differentiable economics A powerful approach to automated mechanism design, sometimes called *differentiable economics*, uses deep neural networks to represent and optimize economic mechanisms. The seminal work of Dütting et al. [16] introduces the RegretNet technique, in which an auction is directly represented as a neural network, which is trained to optimize virtual welfare while minimizing regret violations. There is a wide literature building on this basic idea by improving network architectures and applying the technique to new mechanism design settings [14; 12]. The core limitation of RegretNet is that the guarantee of strategyproofness is only approximate and may be hard to measure. This has motivated the design of specialized network architectures which can train high-performing auctions while ensuring exact strategyproofness by construction, for example by restricting to the class of affine maximizer auctions and learning continuously-parameterized allocation menus [11; 15], by using a more computationally heavy but more general menu-based approach [40], or by using reinforcement learning to train a sequential mechanism [34; 6]. Several recent works are of particular interest in the context of MIRT. Prior works for feed generation [46;

47] use a RegretNet-inspired training approach (further discussion of these and other works is in Section A), exhibiting large DSIC violations. Duan et al. [15] uses transformer networks to generate the parameters of an affine maximizer (a generalization of MIR mechanisms), but their mechanism design problem is classical additive auction design rather than our feed auctions, their architecture requires a hardcoded range size, and their training techniques search over a continuous space of lottery allocations.

Maximal-in-range mechanisms The Vickrey-Clarke-Groves (VCG) mechanisms choose the allocation that maximizes welfare while using transfers that ensure strategyproofness. [38; 10; 20]. If welfare maximization is only approximate, then the corresponding VCG transfers are in general not strategyproof. One important exception is the class of maximal-in-range (MIR) mechanisms, which optimize exactly over a restricted range of possible outcomes [26; 33]. As long as the range is chosen without considering bids, the resulting mechanism with VCG payments is strategyproof. Mechanism design within this class then amounts to designing the range. Some existing theoretical work studies approximately welfare maximizing maximal-in-range mechanisms in single-parameter combinatorial auction settings [18; 22] and in multi-unit auctions [13], as well as sample complexity of learning closely related classes of mechanisms [3; 4].

2 Model and Preliminaries

A platform ranks ads and organic content in a user's feed. Let $\mathcal{A} = \{a_1, \ldots, a_M\}$ and $\mathcal{O} = \{o_1, \ldots, o_{N_O}\}$ denote the sets of candidate ads and organic posts, respectively. The set of all candidate items is $\mathcal{I} = \mathcal{A} \cup \mathcal{O}$, of size $N = M + N_O$. Each advertiser $j \in [M]$ submits a single ad a_j and bid $b_j \in \mathbb{R}_+$, interpreted as its value conditional on a click, and has some underlying true value conditional on a click $v_j \in \mathbb{R}^+$. We write $B = (b_1, \ldots, b_M)$ for the bid profile. True values are drawn from some distribution with CDF $F_i(v_i)$. Each organic post $o \in \mathcal{O}$ generates welfare $h_o \geq 0$ when the user engages with it. Let $e_i \in \mathbb{R}^d$ be the embedding of item $i \in \mathcal{I}$, and let $C \in \mathcal{C}$ collect the features of all items.

A feed with horizon $T \leq M + N_O$ is a sequence $x = (x_1, \ldots, x_T) \in \mathcal{X}$. For every displayed item i, let $p_x(i) = x^{-1}(i)$ denote its position in the feed, and let $q_i(x)$ denote the rate at which the relevant engagement event occurs in feed x. This rate may depend both on the item's position and on the other items in the feed. In particular, we have $q_j(x) = \rho_j(p)\gamma_j(x, p)$, where $\rho_j(p)$ is the baseline, position-dependent click-through rate of ad a_j in the absence of externalities, and $\gamma_j(x, p)$ is an externality multiplier determined by the content around the ad.

The platform's joint ranking optimization problem. The expected welfare generated by feed x assuming truthful bidding is

$$V(x, B) = \sum_{a_j \in x \cap \mathcal{A}} b_j q_j(x) + \sum_{o \in x \cap \mathcal{O}} h_o q_o(x), \tag{1}$$

where h_o is the welfare per engagement of organic o. Thus, an ad click contributes the corresponding advertiser's bid, while engagement with an organic post contributes its platform welfare.

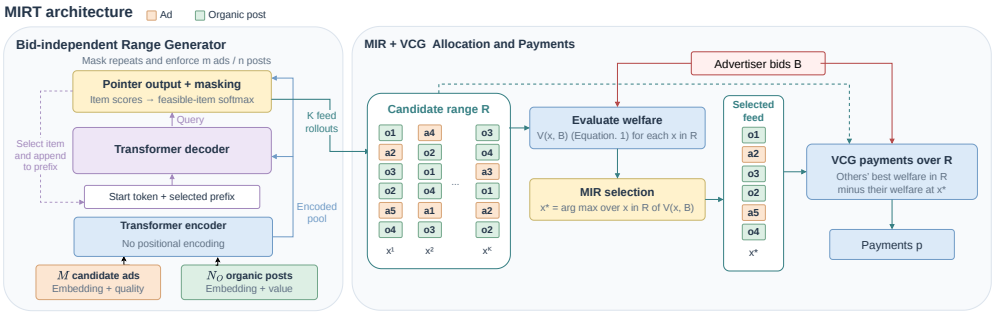
2.1 Modeling externalities

The multiplier $\gamma_j(x, p)$ models how the organic posts surrounding an ad change its click-through rate. Our modeling is based on the intuition that placing an ad among relevant organic posts increases its CTR. We experiment with the previous slot's similarity as the default externality: $\gamma_j(x, p) = 1 + \eta\, s_{j,o_{p-1}}$ $(\eta = 0.5)$ where o_{p-1} is the item in the slot above p, and $s_{j,o_{p-1}}$ is the similarity between that post and the ad. We explore various other models listed in Appendix B.

2.2 Bidder Models and Transfers

Bidder model and strategyproofness In addition to choosing a feed, we allow charging a transfer $t_i(B)$ to each advertiser based on the bids, and implicitly the chosen feed x. A bidder with welfare v_i may misreport some b_i. The utility of advertiser i is: $u_i(b_i, v_i, b_{-i}) = v_i q_i(x(B)) - t_i(B)$. We can now define strategyproofness or dominant strategy incentive compatibility: $u_i(v_i, v_i, b_{-i}) \geq u_i(b'_i, v_i, b_{-i})$, for all bidders i, true values v_i, opponent bid profiles b_{-i}, and possible misreports b'_i.

Figure 1: Schematic diagram of the MIRT architecture.



The VCG mechanism The platform's goal is to select the welfare-maximizing feed $x^\star \in \arg\max_{x\in\mathcal{X}} V(x, B)$. The Vickrey-Clarke-Groves mechanism is a strategyproof mechanism with x^* as its allocation rule [38; 10; 20], which charges bidders the externality they cause to everyone else by being present in the auction: $t_i(B) = \max_{x\in\mathcal{X}} V(x, B_{-i}) - (V(x^*, B) - b_i q_i(x^*))$. As discussed below, exactly computing x^* as required for VCG is intractable, so we use MIR mechanisms instead, which have the same allocation and transfer rule but with feed choices restricted to $x \in R$.

3 Learning Maximal-in-Range Mechanisms

The ideal solution to the feed generation problem would be to run a VCG advertisement auction. A VCG mechanism selects a welfare-maximizing allocation out of all possible allocations and calculates transfers for the corresponding allocation. In our advertisement auction setting, the allocation is an ordered selection of m ads and n organic posts out of a pool of M ads and N_O posts, which we call a feed. The number of possible feeds is $\binom{M}{m}\binom{N_O}{n}(m+n)!$, making exhaustive welfare evaluation intractable. Instead, we use a *maximal-in-range (MIR)* mechanism that selects the welfare-maximizing allocation from a smaller set of candidates (range), preserving incentive compatibility as long as the range is generated independently of advertiser bids [33].

An MIR mechanism is defined by a range R of candidate feeds. This range may depend on the context C, including the true ad qualities, organic welfare, item embeddings, and user features, but must be independent of advertiser bids to stay DSIC. Once the range is fixed and bids are observed, the mechanism performs well as long as one element has high performance. Therefore, a good MIR mechanism must predict high quality feeds from the context, and choose a combination of such feeds that covers as well as possible the various bid profiles which may arrive.

Our range generation model with parameters θ can produce a context dependent range $R_\theta(C)$. Given this range, the mechanism evaluates candidate welfare using the submitted bids, selects the allocation $a_\theta(B, C)$, and computes VCG transfers over the same range: $a_\theta(B, C) \in \arg\max_{x\in R_\theta(C)} V(x, B)$. Motivated by the sequential nature of feed construction and the prior works' models, we propose an RL-trained encoder-decoder architecture to generate a feed by selecting ads and organic posts autoregressively from a candidate pool of items. The model takes item embeddings, organic post welfare, and ad qualities as inputs and generates one feed per rollout. A collection of K distinct feeds creates the candidate range $R_\theta(C)$ for the MIR mechanism.

3.1 Neural Architecture

We use an encoder-decoder transformer [37] with a pointer output [39; 23] to construct feeds from bid-independent item features (1.38M parameters). The encoder has no positional encoding, therefore it is permutation-equivariant [44] and reordering the candidate pool does not reorder its representations. The pool is encoded once per range and shared across all K rollouts [25]. The decoder then builds a feed autoregressively, at each step applying causal self-attention over the items chosen so far, cross-attending to the encoded pool, and scoring every remaining candidate with a clipped scaled dot-product pointer [5; 23]. Decoder inputs carry a learned positional encoding [17].

Our design has the following considerations: first, a mask at every decoder step blocks items already placed or the items that would violate the feed's composition, so every rollout is a valid feed by construction rather than by a penalty term. Second, the VCG mechanism needs a diverse menu of feeds both to produce higher welfare and transfer assignments[21; 19]. In our design, each rollout is conditioned on a per-rollout embedding [7; 9] so that the K rollouts of one context produce K different feeds instead of K copies of similar feeds.

3.2 Training algorithm

We train the range generator using on-policy policy-gradient reinforcement learning [42]. The policy π_θ constructs each feed item by item per rollout, conditioned on the context C, repeated K times to generate the range. To score a range, the environment draws $D = 8$ bid profiles from the prior F and takes the welfare of the best feed in the range under each, so the reward is the expected best-of-K welfare, $r(R; C) = \mathbb{E}_{B\sim F}\left[\max_{x\in R} V(x, B)\right]$.

We follow POMO [25] which builds on REINFORCE policy gradient with an entropy bonus and a baseline shared across rollouts of the same instance rather than a learned critic. For each context we generate $G = 8$ ranges for the baseline, force their rollouts to begin with different items so they explore different feeds, and define each range's advantage as its reward minus the mean reward of the other $G - 1$ ranges [24], scoring all G on the same D bid profiles. Advantages are standardized and centered across batches. We additionally encourage exploration through forced initial selections and added rollout-diversity vectors. During training, we force distinct initial items for up to $\min(K, N)$ rollouts following Kwon et al. [25] and let the remaining rollouts sample their first item from the policy, including those sampled initial choices. For all training details, including network architecture and optimizer details, see Section E.

4 Sample Complexity of MIRT

We analyze the statistical generalization of MIRT under hard attention when the range is dynamically generated via beam search. Let $\mathcal{C}$ be the space of contexts and $\mathcal{B}$ the space of bid profiles. An auction instance is defined as a tuple $S = (C, B)$, where $C \in \mathcal{C}$ contains the features of N candidate content items, and $B \in \mathcal{B}$ contains the corresponding bids. All detailed proofs can be found in Section C.

As in Yang et al. [43], let s_θ denote the score computed by an autoregressive transformer without layer normalization, with piecewise-linear activation functions and hard attention, weights $\theta \in \mathbb{R}^W$ and *computational depth* L, defined as the total number of sequential attention and feed-forward layers. At each generation step $t \in \{1, \ldots, T\}$, the transformer scores a candidate piece of content a given the candidate pool C and the items placed so far. Writing $C(x_{<t})$ for the features of the items in the first $t - 1$ slots, we denote this score as $s_\theta(a \mid C, x_{<t})$.

MIRT generates a discrete range of candidate feeds $R_\theta(C)$ of cardinality at most K using deterministic beam search of width K over T content slots, with a fixed tie-breaking rule. The welfare function evaluated on instance S is $w(\theta; S) = \max_{x\in R_\theta(C)} V(x, B)$, where $V(x, B)$ is the welfare of feed x under bids B. We define the hypothesis class of welfare functions parameterized by the network as $\mathcal{W} = \{w(\theta; \cdot) \mid \theta \in \mathbb{R}^W\}$.

First, we state our main theoretical result, Theorem 1:

Theorem 1 (Pseudo-dimension of MIRT)**.** *Let $\mathcal{W}$ be the welfare function class induced by the transformer class defined above, with computational depth L and $W \geq L$ parameters, where beam search of width K generates ranges of feeds of length T from a set of N candidate pieces of content. Then,*

$$\mathrm{Pdim}(\mathcal{W}) = O\big(WLT \log(TW) \log(TKN)\big). \tag{2}$$

Theorem 1 enables us to bound the sample complexity of learning an almost-optimal member of the MIRT class using the standard pseudo-dimension uniform-convergence guarantee of Li et al. [28]:

Theorem 2 (Sample complexity of MIRT)**.** *Let $V_{\max}$ be the maximum possible welfare. Then for any distribution over auction instances $\mathcal{D}$, the sample complexity of learning with probability at least $1 - \delta$ a parametrization of $\mathcal{W}$ with expected welfare within ϵ of the optimal achievable by the class is:*

$$S = O\left(\frac{V_{\max}^2}{\epsilon^2}\left(WLT\log(TW)\log(TKN)\log\left(\frac{V_{\max}}{\epsilon}\right)+\log\left(\frac{1}{\delta}\right)\right)\right). \tag{3}$$

Proof Idea. To bound $\mathrm{Pdim}(\mathcal{W})$, we must bound the maximum number of auction instances m that $\mathcal{W}$ can shatter. The proof proceeds in five main steps:

1. **Reducing from bounding welfare cardinality to bounding range cardinality (Lemma 1):** The Transformer parameters θ affect welfare only through the generated range $R_\theta(C)$. In particular, for any fixed auction instance, two parameter vectors that generate the same range yield exactly the same welfare. Thus, the number of distinct welfare-threshold labelings that $\mathcal{W}$ can generate on any fixed collection of auction instances is upper bounded by the number of distinct range tuples it can generate.
2. **Characterizing range generation through pairwise comparisons (Lemma 2):** Beam search can be viewed as a sequence of branching decisions. At each autoregressive step, the next beam state is determined by pairwise comparisons between the Transformer scores of at most KN candidate continuations. Hence, conditional on the current beam state, at most $q = \binom{KN}{2} = O(K^2N^2)$ pairwise comparisons determine the next beam.
3. **Bounding the number of realizable range tuples by the VC dimension of the Transformer binary comparison class (Lemma 3):** Let $\mathcal{H}_{\mathrm{cmp}}$ denote the binary hypothesis class induced by these pairwise Transformer-score comparisons, with VC dimension $d_{\mathcal{H}_{\mathrm{cmp}}}$. Conditional on any realizable beam-state history, the comparison inputs at the next generation step are fixed. Thus, we can apply the growth function of $\mathcal{H}_{\mathrm{cmp}}$ sequentially across the T generation steps. Sauer's lemma [35] then bounds the number of distinct range tuples that can be generated on m fixed contexts as a function of the VC dimension of $\mathcal{H}_{\mathrm{cmp}}$.
4. **Bounding the VC dimension of the binary comparison class (Theorem 3):** We adapt the parameter-space partitioning argument of Yang et al. [43]. For any fixed collection of inputs, Yang et al. [43] showed that the parameter space can be partitioned into regions on which all Transformer branching decisions are fixed and the resulting Transformer scores are polynomial in the parameters of degree $O(L)$ (Lemma 4). Thus, within each such region, every pairwise score comparison is the sign of a difference of two degree-$O(L)$ polynomials, which is itself a polynomial of degree $O(L)$. Warren's theorem [41] bounds the number of realizable comparison patterns (Lemma 5), yielding $d_{\mathcal{H}_{\mathrm{cmp}}} = O(WL\log(TW))$.
5. **From range growth to pseudo-dimension (Theorem 1):** Pseudo-shattering m auction instances requires at least 2^m distinct range tuples. Comparing this requirement with the range-growth bound gives $\mathrm{Pdim}(\mathcal{W}) = O\big(d_{\mathcal{H}_{\mathrm{cmp}}} T\log(TKN)\big)$. Substituting the comparison-class VC-dimension bound yields $\mathrm{Pdim}(\mathcal{W}) = O(WLT\log(TW)\log(TKN))$.

5 Experiments

In this section we present numerical experiments that establish the effectiveness of our proposed MIRT framework compared to baseline mechanisms, including previous methods [47; 46].

5.1 Default Synthetic Environment

To sample our organic posts and ad embeddings, we consider 50 topic centroids and we sample the ads and posts around those. The default environment consists of 80 sampled ads and 300 sampled organic posts per centroid, therefore 15,000 posts and 4,000 ads. This environment contains 50,000 users, and a pool of $M = 10$ ads and $N_O = 18$ posts per user. Therefore the problem size is $(M, N_O, T) = (10, 18, 24)$ with fixed feed length of 24 with a fixed six ads per feed. In this setting, ad qualities follow a lognormal distribution with the cosine similarity between user-item embeddings as the log-mean and a log-standard deviation of 0.5. Organic values follow a Weibull distribution with mean of exponentiated user-item cosine similarity and shape of 0.6. (Appendix Figure 4) The default bid prior is exponential with rate 1 and no shift: (mean 1, standard deviation 1). Details on synthesizing the environment can be found in Appendix D. Additionally, the default environment reduces the impression weight by 0.02 with each position down the feed and also reduces the

conversion weight for ads by 0.01. Finally the externality function for this environment takes into account the similarity between the ad and its previous posts also introduced as Position-modulated externality in Appendix B with $\eta_{\text{above}} = 0.5$ and $\eta_{\text{below}} = 0$.

5.2 Experimental Setup

We designed our experiments to explore our method's robustness with respect to different problem sizes, bid distributions and externality functions.

Environment variants. Starting from the default environment ($(M, N_O, T) = (10, 18, 24)$), we vary one component at a time. We experiment with Exponential(1) (mean 1), Uniform$(0.001, 4)$ (mean ≈ 2), LogNormal$(\mu{=}0, \sigma{=}0.5)$ (mean $e^{0.125} \approx 1.13$), and Pareto$(\alpha{=}1.5, x_m{=}1)$ (mean 3, infinite variance) distributions. We train models for each externality model, and various problem sizes by considering different numbers of candidate ads, organic posts, and feed lengths. For the additive, activation, attention, multiplicative, and topic externalities, we use a neighborhood radius of $w = 4$, covering organic posts up to four positions before or after the ad, and an externality strength of $\eta = 0.5$. For more details on the externality functions, please see Appendix Table 3.

MIRT variants. We evaluate MIRT using ancestral sampling with range sizes $K \in \{32, 64, 128, 256, 512\}$, (denoted MIRT $K = .$). Every trained model, although trained on $K = 32$, is capable of generating any range size. MIRT range generation is independent of bids, so is exactly DSIC. However, varying the bid distributions, problem sizes and externality modelings will require separate model training.

Training details. Among the 50,000 dataset samples, we used 40,000 for training, 7,500 for testing and 2,500 for validation. The training hyperparameters follow the description in Section 3, with $K = 32$. The ancestral decoding temperature defaults to 1 and where mentioned, is tuned on a validation set (see Figure 5 for the effect of temperature).

5.3 Baselines and evaluation methods

MIR baselines. As our main baseline, we consider Random+MIR, which builds a bid-independent range of $K = 32$ feeds, each drawn uniformly at random from the feasible feeds. MIR then selects the feed with the highest welfare at the reported bids.

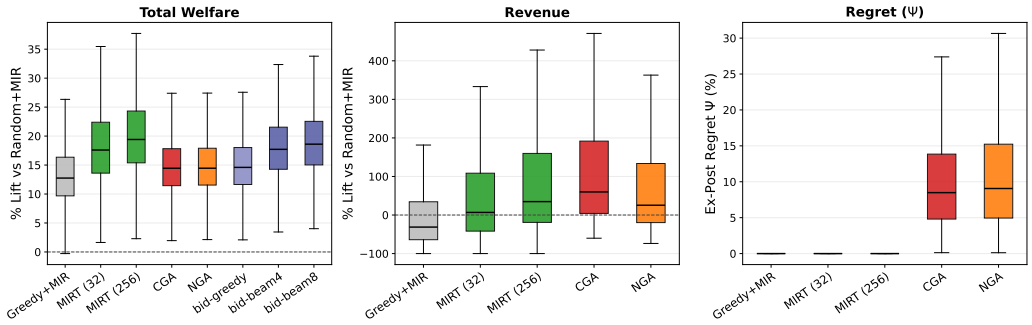
We also consider a bid-blind greedy method to generate the range (Greedy+MIR). Greedy+MIR scores each item by its bid-independent value, meaning welfare generated by an ad's quality and a post's user utility, and not the bids. It then builds a feed by keeping the highest-scoring feasible set of items and placing the best ones in the most prominent slots. The first feed follows this ranking exactly. The remaining $K - 1$ feeds perturb the scores with random noise, so the range holds K distinct high-quality feeds.

Comparisons to prior work. We re-implement modified versions of CGA and NGA [47; 46], adapted to our setting (see Appendix F for details). These methods have terms in the loss function to encourage incentive compatibility during training but do not guarantee exact DSIC. CGA and NGA regret hyperparameters (ρ, the strength of regret penalty) were tuned so they produce lower regret values to become closer to our zero regret setting. We found that additional regret minimization lowered their performance. It is possible that different hyperparameters could unlock improved performance with low regret; on the other hand, it is also possible that the empirical method for estimating regret fails to detect some misreports. We see these ambiguities as a fundamental limitation of approximately-strategyproof methods for neural mechanism design.

Bid-aware baselines. To construct a strong welfare benchmark for this setting, we used bid-aware beam search. This method does not generate a range for MIR, but by calculating the welfare produced by every item (including actual bids in the ad welfare) produces a single high welfare feed with beam-search algorithms with 4 and 8 width. This method is a strong welfare benchmark, but because it cannot be combined with an appropriate payment rule, it cannot be interpreted as a mechanism.

Performance evaluation. We compare MIRT variants with Random+MIR, Greedy+MIR, CGA, NGA, and bid-aware greedy and beam-search benchmarks. We evaluate MIRT using ancestral sampling. We report organic welfare, advertiser welfare, (total) social welfare, virtual welfare, and empirical regret which we denote Ψ. Regret is how much more utility an advertiser could have gained by misreporting their bid, measured after the bids are known [16]. When comparing all other methods, we report either absolute metrics or lift over Random+MIR.

Figure 2: **Performance in the reference environment.** Box-plot with medians/spread for test contexts for one trained model. The first two panels show per-context percentage lift over Random+MIR in total welfare, and virtual welfare. The final panel shows ex-post regret as a percentage. MIRT used sampling temperature 2.4 and CGA/NGA used $\rho = 1000$. MIRT achieves higher total welfare than CGA and NGA, and has zero regret by construction.



5.4 Main results for reference configuration

We first compare all methods under the reference configuration described above. We evaluate social welfare, virtual welfare, and incentive properties. Results are shown in Figure 2 and Appendix Table 6 includes more detail. Section G.1 also contains experiments training for the modified objective of virtual welfare. For our main objective of total welfare, MIRT outperforms all other mechanisms, including our Greedy+MIR baseline, and crucially, the prior SOTA approaches in the literature, CGA and NGA. For revenue, MIRT outperforms the Greedy+MIR baseline, while being marginally below CGA, and on par with NGA. Importantly, MIRT achieves this performance while being exactly strategyproof by construction. By comparison, CGA and NGA exhibit very high amount of regret.

5.5 Varying externality functions

Table 1: Mean ($\pm$ approximate 95% CI) absolute total welfare across held-out contexts under each externality function with default problem size and bid distribution. Highlighted cells are the highest-performing mechanisms. Corresponding virtual welfare is in Appendix Table 4. This table was generated with 40 epochs of training for each MIRT model, and sampling temperature 1 (except for the Prev. Organic externality, which used 2.4). CGA/NGA used $\rho = 1$.

Externality	Prev. Organic	Additive	Activation	Attention	Multiplicative	Topic	Position
Random+MIR	27.18 ± 0.214	27.34 ± 0.215	29.67 ± 0.227	28.50 ± 0.220	27.49 ± 0.216	27.18 ± 0.214	27.33 ± 0.214
Greedy+MIR	30.71 ± 0.239	30.83 ± 0.240	33.24 ± 0.255	31.93 ± 0.245	30.91 ± 0.242	30.63 ± 0.237	30.93 ± 0.241
MIRT ($K = 32$)	32.11 ± 0.245	31.80 ± 0.243	33.92 ± 0.257	32.41 ± 0.247	33.00 ± 0.254	31.87 ± 0.247	32.86 ± 0.250
MIRT ($K = 256$)	32.50 ± 0.246	32.20 ± 0.245	34.47 ± 0.258	32.86 ± 0.248	33.49 ± 0.256	32.34 ± 0.249	33.30 ± 0.252
CGA	31.08 ± 0.238	31.37 ± 0.241	33.55 ± 0.254	32.56 ± 0.247	31.35 ± 0.241	31.20 ± 0.239	31.15 ± 0.239
NGA	31.09 ± 0.238	31.33 ± 0.240	33.51 ± 0.254	32.59 ± 0.247	31.37 ± 0.241	31.23 ± 0.239	31.16 ± 0.239
bid-greedy	31.12 ± 0.238	31.36 ± 0.241	33.54 ± 0.254	32.61 ± 0.247	31.39 ± 0.241	31.23 ± 0.239	31.19 ± 0.239
bid-beam4	31.96 ± 0.243	32.29 ± 0.244	35.52 ± 0.262	33.47 ± 0.251	32.54 ± 0.249	32.20 ± 0.247	32.44 ± 0.246
bid-beam8	32.20 ± 0.244	32.42 ± 0.245	35.58 ± 0.262	33.53 ± 0.251	32.78 ± 0.250	32.35 ± 0.247	32.77 ± 0.248

We examine whether MIRT's welfare performance is robust to different forms of permutation externalities, as described in Appendix B. We vary the externality function while holding all other experimental parameters fixed. Results are in Table 1. For our main objective of welfare, MIRT outperforms all other mechanisms, for all externality functions. For virtual welfare (Appendix Table 4),

Table 2: Mean ($\pm$ CI across held-out contexts) absolute virtual welfare and total welfare across held-out contexts under each bid prior and default problem size and externality. Highlighted cells are the highest-performing mechanisms. Results in this table were generated with 40 epochs of training for each MIRT model, with temperature 2. CGA/NGA have $\rho = 1000$.

	Social Welfare				Virtual Welfare			
Distribution	Exponential	Uniform	Lognormal	Pareto	Exponential	Uniform	Lognormal	Pareto
Random+MIR	27.15 ± 0.214	34.27 ± 0.222	27.33 ± 0.205	44.86 ± 0.761	2.05 ± 0.039	4.83 ± 0.073	2.47 ± 0.040	5.60 ± 0.093
Greedy+MIR	30.71 ± 0.240	39.83 ± 0.257	31.38 ± 0.227	50.59 ± 0.781	1.44 ± 0.023	3.29 ± 0.048	1.93 ± 0.026	3.72 ± 0.058
MIRT ($K = 32$)	32.06 ± 0.245	42.67 ± 0.268	32.69 ± 0.230	53.49 ± 0.882	2.23 ± 0.035	5.55 ± 0.075	2.76 ± 0.032	5.05 ± 0.073
MIRT ($K = 256$)	32.52 ± 0.246	43.32 ± 0.269	32.95 ± 0.231	55.02 ± 0.916	2.76 ± 0.037	7.14 ± 0.081	3.75 ± 0.036	6.64 ± 0.076
CGA	31.08 ± 0.238	40.37 ± 0.252	31.19 ± 0.226	51.00 ± 0.767	3.26 ± 0.024	6.35 ± 0.024	3.29 ± 0.011	3.25 ± 0.015
NGA	31.09 ± 0.238	40.31 ± 0.252	31.20 ± 0.226	50.98 ± 0.766	2.50 ± 0.016	5.97 ± 0.022	2.90 ± 0.009	2.22 ± 0.014
bid-greedy	31.12 ± 0.238	40.39 ± 0.252	31.22 ± 0.226	51.02 ± 0.766	–	–	–	–
bid-beam4	31.96 ± 0.243	41.50 ± 0.260	32.06 ± 0.229	52.73 ± 0.813	–	–	–	–
bid-beam8	32.20 ± 0.244	41.87 ± 0.262	32.27 ± 0.230	53.55 ± 0.839	–	–	–	–

the best-performing mechanism depends on the externality function. However, again, MIRT has zero regret by construction, while the other mechanisms exhibit high amounts of regret.

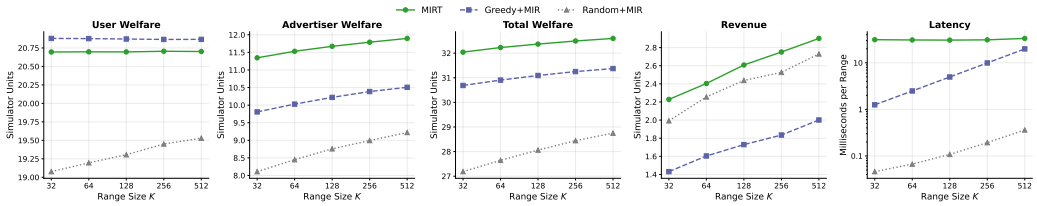
5.6 Varying bid distributions

In Table 2, we evaluate the robustness of MIRT's performance to different advertiser populations by varying the distribution of advertiser bids. Again, for our central objective of Social Welfare, MIRT outperforms all other mechanisms by a wide margin, even at $K = 32$. For uniform, lognormal, and Pareto distributions, it even outperforms all other mechanisms in terms of virtual welfare.

5.7 Varying range size

MIR mechanism welfare improves as more items are added to the range; when the range is large enough to include all feeds, exact VCG is recovered. In Figure 3, we show the user welfare, advertiser welfare, total welfare, and virtual welfare in the reference environment as we vary the range over $K \in \{32, 64, 128, 256, 512\}$, along with latency. MIRT ran with decoding temperature 2. Because MIRT runs in parallel on the GPU, its latency does not increase with the range size. Unsurprisingly, performance in both welfare and virtual welfare improves as the size of the range increases.

Figure 3: Scaling of performance with the size of the range.



6 Discussion and Conclusion

We present MIRT, an exactly strategyproof mechanism for whole-feed optimization. MIRT uses an autoregressive transformer to generate the feeds in the range of a maximal-in-range mechanism, optimized via a reinforcement learning signal that takes into account both individual feed and collective range quality. We show that with hard-attention transformers, near-optimal expected welfare is learnable with sample complexity polynomial in the transformer size and only logarithmic in the range width. In extensive experiments, we show that MIRT outperforms prior state-of-the-art approaches in welfare across all settings tested, and remains competitive in virtual welfare. At the same time, these prior approaches for joint ranking show high strategyproofness violations in our experiments, while MIRT ensures exact strategyproofness by construction. There are many promising directions for future work, including testing improved RL algorithms, extending our methodology to broader classes of mechanisms such as affine maximizers, and applying the same basic approach to many other mechanism design problems beyond feed optimization.

AI USE STATEMENT

We used generative AI in several ways in our paper. We had AI tools review our draft and provide suggestions for improved language as well as high-level suggestions about which parts of our argument were more or less convincing ("edit our paper to improve readability", "provide feedback on research methodology or experiments"). We used AI as a search tool in a human driven literature search to surface papers we might not have been aware of, in addition to related work found manually or already known to the authors ("source/search for information", "identify relevant literature", "summarize or analyze existing literature"). We used AI coding assistants extensively for our code, which includes matplotlib code to generate plots ("create or edit software code", "create or modify scientific figures"). The proof in Section 4 is primarily human work, but conversational AI assistance was used to suggest some counting lemmas to tighten the bound, as well as to edit the presentation of the proof for clarity and correctness ("formulate mathematical claims", "provide critical ingredients for proving mathematical claims," "assist in the writing of proofs").

To use the suggested ICLR language: In this work, we used generative AI tools to formulate mathematical claims, provide feedback on research methodology or experiments, provide critical ingredients for proving mathematical claims, and assist in the writing of proofs. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, design research methodology or experiments, interpret results, or propose or refine hypotheses, and cleaning and reformatting datasets, implementing methods, supporting qualitative and thematic data analysis, or assisting with translation are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures, create or edit software code, summarize or analyze existing literature, source/search for information, edit our paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

[1] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, USA, 1st edition, 2009. ISBN 052111862X.

[2] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 482–491. IEEE, 2001.

[3] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *NIPS 2016*, pp. 2091–2099, 2016.

[4] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *EC 2018*, pp. 173–174, 2018.

[5] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *International Conference on Learning Representations, Workshop Track*, 2017. URL https://arxiv.org/abs/1611.09940.

[6] Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David Parkes, and Duncan Rheingans-Yoo. Reinforcement learning of sequential price mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5219–5227, 2021.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, 2020. URL https://arxiv.org/abs/2005.12872.

[8] Carlos Carrion, Zenan Wang, Harikesh Nair, Xianghong Luo, Yulin Lei, Peiqin Gu, Xiliang Lin, Wenlong Chen, Junsheng Jin, Fanan Zhu, et al. Blending advertising with organic content in e-commerce via virtual bids. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 15476–15484, 2023.

[9] Felix Chalumeau, Shikha Surana, Clément Bonnet, Nathan Grinsztajn, Arnu Pretorius, Alexandre Laterre, and Tom Barrett. Combinatorial optimization with policy adaptation using latent space search. In *Advances in Neural Information Processing Systems*, volume 36,

2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/18d3a2f3068d6c669dcae19ceca1bc24-Abstract-Conference.html.

[10] Edward H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11:17–33, 1971. ISSN 0048-5829.

[11] Michael Curry, Tuomas Sandholm, and John Dickerson. Differentiable Economics for Randomized Affine Maximizer Auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

[12] Michael J. Curry, Uro Lyi, Tom Goldstein, and John P. Dickerson. Learning Revenue-Maximizing Auctions With Differentiable Matching. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 6062–6073. PMLR, May 2022.

[13] Shahar Dobzinski and Noam Nisan. Mechanisms for Multi-Unit Auctions. 2007.

[14] Zhijian Duan, Jingwu Tang, Yutong Yin, Zhe Feng, Xiang Yan, Manzil Zaheer, and Xiaotie Deng. A context-integrated transformer-based neural network for auction design. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 5609–5626. PMLR, 2022.

[15] Zhijian Duan, Haoran Sun, Yurong Chen, and Xiaotie Deng. A Scalable Neural Network for DSIC Affine Maximizer Auction Design. *Advances in Neural Information Processing Systems*, 36:56169–56185, December 2023.

[16] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning: Advances in differentiable economics. *Journal of the ACM*, 71(1):1–53, 2024.

[17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1243–1252, 2017. URL https://proceedings.mlr.press/v70/gehring17a.html.

[18] Gagan Goel, Chinmay Karande, and Lei Wang. Single Parameter Combinatorial Auctions with Partially Public Valuations. volume 6386, pp. 234–245. 2010. doi: 10.1007/978-3-642-16170-4_21.

[19] Nathan Grinsztajn, Daniel Furelos-Blanco, Shikha Surana, Clément Bonnet, and Thomas D. Barrett. Winner takes it all: Training performant RL populations for combinatorial optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2210.03475.

[20] Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973. ISSN 0012-9682. doi: 10.2307/1914085.

[21] Abner Guzmán-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://papers.nips.cc/paper_files/paper/2012/hash/cfbce4c1d7c425baf21d6b6f2babe6be-Abstract.html.

[22] Zhiyi Huang, Lei Wang, and Yuan Zhou. Black-Box Reductions in Mechanism Design. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim (eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 6845, pp. 254–265. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-22934-3 978-3-642-22935-0. doi: 10.1007/978-3-642-22935-0_22.

[23] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

[24] Wouter Kool, Herke Van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

[25] Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. POMO: Policy optimization with multiple optima for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2010.16011.

[26] R. Lavi, Ahuva Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 574–583, Cambridge, MA, USA, 2003. IEEE Computer. Soc. ISBN 978-0-7695-2040-7. doi: 10.1109/SFCS.2003.1238230.

[27] Xuejian Li, Ze Wang, Bingqi Zhu, Fei He, Yongkang Wang, and Xingxing Wang. Deep automated mechanism design for integrating ad auction and allocation in feed. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1211–1220, 2024.

[28] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.*, 62(3):516–527, 2001.

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[31] Yuchao Ma, Ruohan Qian, Bingzhe Wang, Qi Qi, Wenqiang Liu, Qian Tang, Zhao Shen, Wei Zhong, Bo Shen, Yixin Su, et al. Genauction: A generative auction for online advertising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12372–12380, 2025.

[32] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[33] N. Nisan and A. Ronen. Computationally Feasible VCG Mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47, May 2007. ISSN 1076-9757. doi: 10.1613/jair.2046.

[34] Sai Srivatsa Ravindranath, Zhe Feng, Di Wang, Manzil Zaheer, Aranyak Mehta, and David C. Parkes. Deep reinforcement learning for sequential combinatorial auctions. arXiv:2407.08022, 2024.

[35] N Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. ISSN 0097-3165. doi: https://doi.org/10.1016/0097-3165(72)90019-2. URL https://www.sciencedirect.com/science/article/pii/0097316572900192.

[36] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

[38] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961. ISSN 0022-1082. doi: 10.2307/2977633.

[39] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[40] Tonghan Wang, Yanchen Jiang, and David C. Parkes. GemNet: Menu-Based, Strategy-Proof Multi-Bidder Auctions Through Deep Learning. In *Proceedings of the 2024 ACM Conference on Economics and Computation*, 2024.

[41] Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133:167–178, 1968. URL https://api.semanticscholar.org/CorpusID:46487681.

[42] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[43] Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Tight sample complexity of transformers. In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6887–6923. PMLR, 29 Jun–03 Jul 2026. URL https://proceedings.mlr.press/v336/yang26a.html.

[44] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.06114.

[45] Zuowu Zheng, Ze Wang, Fan Yang, Jiangke Fan, Teng Zhang, Yongkang Wang, and Xingxing Wang. EGA-V2: An end-to-end generative framework for industrial advertising. *arXiv preprint arXiv:2505.17549*, 2025.

[46] Zuowu Zheng, Ze Wang, Fan Yang, Wenqing Ye, Weihua Huang, Wenqiang He, Teng Zhang, and Xingxing Wang. NGA: Non-autoregressive generative auction with global externalities for advertising systems. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, 2025.

[47] Ruitao Zhu, Yangsu Liu, Dagui Chen, Zhenjia Ma, Chufeng Shi, Zhenzhe Zheng, Jie Zhang, Jian Xu, Bo Zheng, and Fan Wu. Contextual generative auction with permutation-level externalities for online advertising. *arXiv preprint arXiv:2412.11544*, 2024.

A Related work: neural feed allocation mechanisms

There are several works using neural networks to learn feed-allocation auction mechanisms, including Ma et al. [31]; Zheng et al. [45]. We highlight two approaches of particular relevance to this work, CGA [47] and NGA [46], both of which consider feed generation in the presence of externalities, and both of which use RL to train their feed generators. Zhu et al. [47] employs an autoregressive decoder that takes context embeddings as input and directly generates ad allocations to maximize expected virtual welfare, taking into account externalities. Their objective is to maximize virtual welfare; the virtual-welfare-maximizing allocation rule has corresponding strategyproof Myerson payments. However, CGA does not learn the exactly optimal allocation rule, and moreover computing Myerson payments (via numerical integration) would be infeasibly slow. Therefore, they use a decoupled PaymentNet module to approximate correct payments, and empirically minimize regret in the fashion of Dütting et al. [16]. The result is an auction which performs well but for which exact strategyproofness cannot be ensured. Zheng et al. [46] introduces a non-autoregressive feed generator with dual encoders for candidate items and candidate positions, and an evaluator which outputs predicted click-through and conversion rates alongside payments, which are used to select a final feed. The networks are also trained using an approach similar to Dütting et al. [16] to encourage incentive compatibility. Our approach also uses transformers trained via RL to generate feeds, but we achieve exact strategyproofness and jointly generate our entire range of feeds at the same time.

B Externality functions

Externality models ($\gamma_j(x,p)$) are how we describe the interaction between ads and surrounding posts. In the following list, we present the different externality modeling functions.

Externalities are defined for a position p, in a feed x, filled with advertisement a_j. The ad a_j has embedding e_j, and a surrounding organic post o has embedding e_o. We write $s_{jo} = \cos(e_j, e_o)$ for their similarity, and $\kappa_x(o,p) = \lambda^{|\,p_x(o)-p\,|}$ for the decay factor that shrinks as o gets further from p. Most of these models have a strength hyperparameter η that defines the intensity of the interactions.

Additive similarity. This method sums up the nearby organic post embeddings with a decay factor based on distance from position p, over the window of slots $p-w$ to $p+w$:

$$\gamma_j(x,p) = 1 + \eta\,\cos\Big(e_j,\ \sum_{|\mathrm{pos}_x(o)-p|\,\le\,w} \kappa_x(o,p)\,e_o\Big), \qquad \eta \ge 0,$$

Neighborhood activation. The ad receives a fixed boost whenever at least one organic post in slots $p-w$ to $p+w$ is sufficiently similar to it, where $s_{jo} = \cos(e_j, e_o)$.

$$\gamma_j(x,p) = 1 + \eta\,\mathbf{1}\left\{\max_{|\mathrm{pos}_x(o)-p|\,\le\,w} s_{jo} > \tau_{\mathrm{act}}\right\}.$$

Softmax attention. The ad attends to all nearby organic posts, with more relevant posts receiving greater weight. Define

$$\alpha_{jo}(x,p) = \frac{\kappa_x(o,p)\exp(\langle e_j, e_o\rangle/\tau_{\mathrm{sm}})}{\sum_{|\mathrm{pos}_x(o')-p|\,\le\,w} \kappa_x(o',p)\exp(\langle e_j, e_{o'}\rangle/\tau_{\mathrm{sm}})}, \qquad \tau_{\mathrm{sm}} > 0.$$

where $\langle e_j, e_o\rangle$ is the unnormalized inner product of the two embeddings. The externality multiplier is:

$$\gamma_j(x,p) = 1 + \eta \sum_{|\mathrm{pos}_x(o)-p|\,\le\,w} \alpha_{jo}(x,p)\,s_{jo}.$$

Multiplicative complements and substitutes. Each organic post in slots $p-w$ to $p+w$ independently scales the ad's CTR, so the effects compound instead of averaging:

$$\gamma_j(x,p) = \prod_{|\mathrm{pos}_x(o)-p|\,\le\,w} (1 + \eta\,\kappa_x(o,p)\,s_{jo})\,.$$

Positive s_{jo} represents complementary content, whereas negative s_{jo} represents substitute or competing content.

Topic-category matching. Affinity is measured in a shared space of K latent topics rather than in raw embedding space. Let $\phi_{j,k}$ denote ad j's affinity for topic k and $\psi_{o,k}$ organic post o's loading on topic k, both distributions over the K topics. The surrounding context is the weighted mixture of its posts' loadings,

$$\bar{\psi}_k(x,p) = \frac{\sum_{|\mathrm{pos}_x(o)-p| \le w} \kappa_x(o,p)\, \psi_{o,k}}{\sum_{|\mathrm{pos}_x(o)-p| \le w} \kappa_x(o,p)},$$

and the ad benefits when its preferred topics are the ones that mixture is rich in:

$$\gamma_j(x,p) = 1 + \eta \left(K \sum_{k=1}^{K} \phi_{j,k}\, \bar{\psi}_k(x,p) \;-\; 1 \right).$$

Position-modulated. Writing o_{p-1} and o_{p+1} for the items in the slots directly above and below p,

$$\gamma_j(x,p) = 1 + \eta_{\text{above}}\, s_{jo_{p-1}} + \eta_{\text{below}}\, s_{jo_{p+1}}, \qquad \eta_{\text{above}} > \eta_{\text{below}},$$

where a term contributes 0 if that slot holds an ad or falls outside the feed.

Externality function	**Parameter values**
Additive Similarity	$\eta = 0.5$, $\lambda = 0.7$, $w = 4$
Neighborhood Activation	$\eta = 0.5$, $\tau_{\text{act}} = 0.3$, $w = 4$
Softmax Attention	$\eta = 0.5$, $\tau_{\text{sm}} = 0.5$, $\lambda = 0.7$, $w = 4$
Multiplicative interactions	$\eta = 0.5$, $\lambda = 0.7$, $w = 4$
Topic affinity	$\eta = 0.5$, $\lambda = 0.7$, $w = 4$, $K = 20$ topics
Position-modulated	$\eta_{\text{above}} = 0.6$, $\eta_{\text{below}} = 0.2$
One side Position-modulated	$\eta_{\text{above}} = 0.5$ (Default Externality)

Table 3: Default parameter values used in all experiments.

C Proofs From Section 4

In this section, we provide the remaining proof steps to bound the sample complexity of MIRT.

Lemma 1. *For any two parameter vectors $\theta_1, \theta_2 \in \mathbb{R}^W$ and any auction instance $S = (C, B)$, if $R_{\theta_1}(C) = R_{\theta_2}(C)$, then $w(\theta_1; S) = w(\theta_2; S)$.*

Proof. Fix $\theta_1, \theta_2 \in \mathbb{R}^W$ and an auction instance $S = (C, B)$ such that $R_{\theta_1}(C) = R_{\theta_2}(C)$.

Then

$$\begin{aligned} w(\theta_1; S) &= \max_{x \in R_{\theta_1}(C)} V(x, B) \\ &= \max_{x \in R_{\theta_2}(C)} V(x, B) \\ &= w(\theta_2; S). \end{aligned}$$

The first and third equalities follow from the definition of w, while the second follows from the assumption that the two parameter vectors generate the same range on context profile C. □

Lemma 2 (Algorithmic Branching of Beam Search)**.** *Fix a context profile C containing N candidate pieces of content. Generating a feed of length T using beam search with width K evaluates at most $E_{total} = O(TK^2N^2)$ pairwise comparisons of scores assigned by the transformer to candidate continuations. Given a fixed tie-breaking rule, the outcomes of these comparisons uniquely determine the output range $R_\theta(C)$.*

Proof. At each generation step $t \in \{1, \ldots, T\}$, beam search maintains at most K active prefixes. Each active prefix has at most N possible continuations, so there are at most KN candidate continuations at that step.

Let $s_\theta(C(x_t), p) \in \mathbb{R}$ denote the score assigned by beam search at step t to a candidate continuation p under context $C(x_t)$. Conditional on the beam state entering at step t (i.e., the set of current beams $\{x_{t-1}^1, \ldots, x_{t-1}^K\}$), the set of candidate continuations is fixed. The top K continuations are therefore uniquely determined by the outcomes of all pairwise comparisons $s_\theta(C(x_{t-1}^k), p) > s_\theta(C(x_{t-1}^{k'}), p')$, $k, k' \in [K], p, p' \in [N]$ among these candidate continuations, together with the fixed tie-breaking rule. The number of such comparisons is at most $q = \binom{KN}{2} = O(K^2N^2)$.

Thus, conditional on the beam state entering step t, the outcomes of at most q pairwise score comparisons uniquely determine the beam state entering step $t+1$, applying this argument sequentially over the T generation steps shows that a complete execution evaluates at most

$$E_{\text{total}} = Tq = T\binom{KN}{2} = O(TK^2N^2) \tag{4}$$

comparisons and that their sequential outcomes uniquely determine the final output range $R_\theta(C)$. □

Lemma 3 (Growth of Beam-Search Ranges)**.** *Let $\mathcal{H}_{\text{cmp}}$ be the binary hypothesis class induced by pairwise comparisons of scores assigned by the transformer class, $d_{\mathcal{H}_{\text{cmp}}} = \text{VCdim}(\mathcal{H}_{\text{cmp}})$ be its VC-dimension and let $q = \binom{KN}{2}$. For any fixed collection of m context profiles $C_1, \ldots, C_m$ arising from m auctions, the number of distinct range tuples $(R_\theta(C_1), \ldots, R_\theta(C_m))$ that can be generated as θ varies is at most*

$$\left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right)^{d_{\mathcal{H}_{\text{cmp}}}T}, \tag{5}$$

if $mq \geq d_{\mathcal{H}_{\text{cmp}}}$, and 2^{mqT} otherwise.

Proof. For each $t \in \{0, \ldots, T\}$, let $\Gamma_t(m)$ denote the maximum number of distinct tuples of beam states after t generation steps that can be produced on the fixed contexts $C_1, \ldots, C_m$ as θ varies. Before generation begins, the beam states are fixed, and hence $\Gamma_0(m) = 1$.

Fix an advertiser context profile C, and a realizable set of beam states after step $t-1$. Conditional on these, the candidate continuations considered at step t are fixed. By Lemma 2, determining the next beam requires at most $q = O(K^2N^2)$ pairwise comparisons for the fixed context C. Thus, across the m contexts, conditioned on the beam states after step $t-1$, the subsequent beam states at step t are determined by the labels assigned by $\mathcal{H}_{\text{cmp}}$ to a fixed collection of at most mq comparison inputs.

The number of possible next beam-state tuples conditional on the fixed history is at most $\Pi_{\mathcal{H}_{\text{cmp}}}(mq)$, where $\Pi_{\mathcal{H}_{\text{cmp}}}$ denotes the growth function of $\mathcal{H}_{\text{cmp}}$, i.e., the maximum number of binary labelings that the class $\mathcal{H}_{\text{cmp}}$ can realize on any fixed set of mq comparison queries. Conditioning on the fixed beam-state history restricts θ to a subset of $\mathbb{R}^W$, and restricting the parameter space cannot increase the number of labelings realizable by $\mathcal{H}_{\text{cmp}}$. Therefore, the growth function of the full class remains a valid upper bound. It follows that

$$\Gamma_t(m) \leq \Gamma_{t-1}(m)\Pi_{\mathcal{H}_{\text{cmp}}}(mq).$$

Applying this inequality recursively over the T generation steps yields

$$\Gamma_T(m) \leq \left(\Pi_{\mathcal{H}_{\text{cmp}}}(mq)\right)^T.$$

However, by Sauer's lemma [35], when $mq \geq d_{\mathcal{H}_{\text{cmp}}}$, we can bound the growth function of $\mathcal{H}_{\text{cmp}}$ as $\Pi_{\mathcal{H}_{\text{cmp}}}(mq) \leq \left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right)^{d_{\mathcal{H}_{\text{cmp}}}}$. Therefore,

$$\Gamma_T(m) \leq \left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right)^{d_{\mathcal{H}_{\text{cmp}}}T}. \tag{6}$$

If instead $mq < d_{\mathcal{H}_{\text{cmp}}}$, then trivially $\Pi_{\mathcal{H}_{\text{cmp}}}(mq) \leq 2^{mq}$, since there are at most 2^{mq} binary labelings of mq comparison inputs. Hence,

$$\Gamma_T(m) \leq 2^{mqT}. \tag{7}$$

The final output range is determined by the beam state after the T-th generation step. Thus, $\Gamma_T(m)$ upper-bounds the number of distinct range tuples $(R_\theta(C_1), \ldots, R_\theta(C_m))$, which proves the result. □

Now we have all the lemmata necessary for the proof of Theorem 1:

Proof of Theorem 1. Suppose that $\mathcal{W}$ pseudo-shatters m auction instances $S_1, \ldots, S_m$ with corresponding thresholds $y_1, \ldots, y_m \in \mathbb{R}$. Then, as θ varies, $\mathcal{W}$ realizes all 2^m binary labelings of the conditions $w(\theta; S_i) > y_i, i \in \{1, \ldots, m\}$.

By Lemma 1, two parameter vectors that generate the same tuple of ranges on $S_1, \ldots, S_m$ generate the same welfare vector, and thus the same binary labeling with respect to the thresholds $y_1, \ldots, y_m$. Consequently, pseudo-shattering the m instances requires the transformer to generate at least 2^m distinct range tuples as its parameters vary in $\mathbb{R}^W$.

Let $d_{\mathcal{H}_{\text{cmp}}} = \text{VCdim}(\mathcal{H}_{\text{cmp}})$ and $q = \binom{KN}{2}$. We distinguish the two cases of Lemma 3.

First, suppose that $mq < d_{\mathcal{H}_{\text{cmp}}}$. Then directly $m < \frac{d_{\mathcal{H}_{\text{cmp}}}}{q} \leq d_{\mathcal{H}_{\text{cmp}}}$.

Now suppose that $mq \geq d_{\mathcal{H}_{\text{cmp}}}$. By Lemma 3, the number of distinct range tuples is at most $\left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right)^{d_{\mathcal{H}_{\text{cmp}}} T}$. Since pseudo-shattering requires at least 2^m distinct range tuples, we must have

$$2^m \leq \left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right)^{d_{\mathcal{H}_{\text{cmp}}} T}.$$

Taking logarithms

$$m \leq d_{\mathcal{H}_{\text{cmp}}} T \log_2 \left(\frac{emq}{d_{\mathcal{H}_{\text{cmp}}}}\right).$$

Using the inversion $m \leq a \log m + b \implies m = O(a \log a + b)$, yields $m = O\left(d_{\mathcal{H}_{\text{cmp}}} T \log(Tq)\right)$.

Since $q = \binom{KN}{2} = O(K^2 N^2)$, we have $\log(Tq) = O\left(\log(TKN)\right)$. Hence,

$$m = O\left(d_{\mathcal{H}_{\text{cmp}}} T \log(TKN)\right). \tag{8}$$

The case $mq < d_{\mathcal{H}_{\text{cmp}}}$ yields the smaller bound $m < d_{\mathcal{H}_{\text{cmp}}}/q$, and is therefore covered by the same asymptotic upper bound. Thus,

$$\text{Pdim}(\mathcal{W}) = O\left(d_{\mathcal{H}_{\text{cmp}}} T \log(TKN)\right). \tag{9}$$

Finally, by Theorem 3, under $L \leq W$, $d_{\mathcal{H}_{\text{cmp}}} = O\left(WL \log(TW)\right)$. Therefore,

$$\text{Pdim}(\mathcal{W}) = O\left(WLT \log(TW) \log(TKN)\right). \tag{10}$$

□

Theorem 3 (VC dimension of pairwise Transformer comparisons)**.** *Fix a Transformer architecture A of depth L and with W trainable parameters, as in Yang et al. [43], and let $s_\theta(H) \in \mathbb{R}$ denote its real-valued pre-threshold output on an input sequence $H \in \mathbb{R}^{T \times d}$. Define the corresponding pairwise-comparison class*

$$\mathcal{H}_{\text{cmp}} := \left\{(H, H') \mapsto \mathbb{1}\left\{s_\theta(H) > s_\theta(H')\right\} : \theta \in \mathbb{R}^W\right\}. \tag{11}$$

Then

$$\text{VCdim}(\mathcal{H}_{\text{cmp}}) = O(WL \log(TWL)).$$

In particular, under the assumption $L \leq W$,

$$\text{VCdim}(\mathcal{H}_{\text{cmp}}) = O(WL \log(TW)). \tag{12}$$

We prove Theorem 3 following the parameter-space partitioning argument of Yang et al. [43]. The proof has two parts. First, for any fixed collection of input pairs, we partition the parameter space into regions on which all attention-routing and ReLU (or any piecewise-linear activation function's) branching decisions are fixed. Second, we show that within each such region, each pairwise comparison is the sign of a polynomial in the Transformer parameters of a bounded degree, allowing us to bound the number of comparison patterns using Warren's theorem [1].

Lemma 4 (Partition of the parameter space into polynomial regions)**.** *Fix m arbitrary input pairs $(H_1^+, H_1^-), \ldots, (H_m^+, H_m^-) \in \left(\mathbb{R}^{T\times d}\right)^2$. There exists a partition $\mathcal{P}_L$ of $\mathbb{R}^W$ such that, for every region $P \in \mathcal{P}_L$:*

1. *all attention-routing decisions and ReLU activation patterns of the Transformer are fixed on each of the $2m$ inputs $H_1^+, H_1^-, \ldots, H_m^+, H_m^-$;*
2. *for every $i \in [m]$, both $s_\theta(H_i^+)$ and $s_\theta(H_i^-)$ coincide on P with polynomials in θ of degree at most $D_{\text{out}} = O(L)$;*
3. *the number of regions satisfies $\log|\mathcal{P}_L| = O(WL(\log m + \log T + \log L))$.*

Proof. The proof follows the recursive parameter-space partitioning argument of Yang et al. [43, Appendix B.1]. For any collection of I fixed input sequences, their construction partitions the parameter space $\mathbb{R}^W$ into regions $\mathcal{P}_L$ such that, within every $P \in \mathcal{P}_L$, all attention-routing decisions and ReLU activation patterns on all I inputs are fixed. Consequently, the Transformer output on each input coincides, within P, with a polynomial in the parameters θ of degree $O(L)$. They upper bound the number of such regions as:

$$\log|\mathcal{P}_L| = O(WL(\log I + \log T + \log L))\,; \tag{13}$$

see Appendix B.1, in particular equations (28)–(30), of Yang et al. [43].

We apply this construction to the $2m$ fixed input sequences $H_1^+, H_1^-, \ldots, H_m^+, H_m^-$. The construction applies to an arbitrary collection of fixed inputs, so no relationship between H_i^+ and H_i^-, or between different pairs, is required. Thus, we simply set $I = 2m$ in the bound above. It follows that, within every region $P \in \mathcal{P}_L$, all branching decisions are fixed simultaneously on all $2m$ inputs, and, for every $i \in [m]$, both $s_\theta(H_i^+)$ and $s_\theta(H_i^-)$ are polynomials in θ of degree $D_{\text{out}} = O(L)$. Moreover,

$$\begin{aligned}\log|\mathcal{P}_L| &= O(WL(\log(2m) + \log T + \log L)) \\ &= O(WL(\log m + \log T + \log L))\,.\end{aligned} \tag{14}$$

□

Lemma 5 (Comparison patterns within a polynomial region)**.** *Fix a region $P \in \mathcal{P}_L$ as in Lemma 4. For each input pair $i \in [m]$, define $g_i^P(\theta) = s_\theta(H_i^+) - s_\theta(H_i^-), \theta \in P$. Then on P, each g_i^P coincides with a polynomial in θ of degree at most $D_{\text{out}} = O(L)$. Moreover, if $m \geq W$, the number Π_P^{cmp} of distinct pairwise-comparison vectors $\left(\mathbb{1}\{s_\theta(H_1^+) > s_\theta(H_1^-)\}, \ldots, \mathbb{1}\{s_\theta(H_m^+) > s_\theta(H_m^-)\}\right)$ realized as θ varies over P satisfies $\Pi_P^{\text{cmp}} \leq 2\left(\frac{2emD_{\text{out}}}{W}\right)^W$.*

Proof. By Lemma 4, for each $i \in [m]$ there exist polynomials $p_{i,+}^P$ and $p_{i,-}^P$ such that, on P, $s_\theta(H_i^+) = p_{i,+}^P(\theta), s_\theta(H_i^-) = p_{i,-}^P(\theta)$, with $\deg(p_{i,+}^P) \leq D_{\text{out}}$ and $\deg(p_{i,-}^P) \leq D_{\text{out}}$. Thus $g_i^P(\theta) = p_{i,+}^P(\theta) - p_{i,-}^P(\theta)$ is itself a polynomial in the same W parameters, of degree at most

$$\deg(g_i^P) \leq \max\{\deg(p_{i,+}^P), \deg(p_{i,-}^P)\} \leq D_{\text{out}}.$$

Although $2m$ transformer executions are required to define the partition, there are only m output-comparison polynomials $g_1^P, \ldots, g_m^P$. Furthermore, $\mathbb{1}\{s_\theta(H_i^+) > s_\theta(H_i^-)\} = \mathbb{1}\{g_i^P(\theta) > 0\}$. Thus the entire comparison vector is determined by the ternary sign pattern of the m polynomials $g_1^P, \ldots, g_m^P$. Applying Warren's theorem to these m polynomials in W variables, each of degree at most D_{out}, gives

$$\Pi_P^{\text{cmp}} \leq 2\left(\frac{2emD_{\text{out}}}{W}\right)^W.$$

□

Proof of Theorem 3. Suppose that $\mathcal{H}_{\text{cmp}}$ shatters m input pairs $(H_1^+, H_1^-), \ldots, (H_m^+, H_m^-)$. If $m < W$, then $m < W = O(WL\log(TWL))$, and the claimed bound follows immediately. Hence assume $m \geq W$.

Let $\mathcal{P}_L$ be the partition supplied by Lemma 4. Since the m pairs are shattered, all 2^m binary comparison vectors are realized. Decomposing according to the partition yields:

$$2^m \leq \sum_{P \in \mathcal{P}_L} \Pi_P^{\text{cmp}} \leq |\mathcal{P}_L| \max_{P \in \mathcal{P}_L} \Pi_P^{\text{cmp}}.$$

Applying Lemmata 4 and 5, we have $2^m \leq |\mathcal{P}_L| \cdot 2\left(\frac{2emD_{\text{out}}}{W}\right)^W$, where $\log|\mathcal{P}_L| = O(WL(\log m + \log T + \log L))$ and $D_{\text{out}} = O(L)$. Taking logarithms yields

$$\begin{aligned} m &= O(WL(\log m + \log T + \log L)) + O\left(W \log\left(\frac{2emD_{\text{out}}}{W}\right)\right) \\ &= O(WL\log m + WL\log T + WL\log L) + O\left(W\log\left(\frac{mL}{W}\right)\right) \\ &= O(WL\log m + WL\log T + WL\log L)\,. \end{aligned}$$

Thus $m \leq O(WL\log m) + O(WL(\log T + \log L))$. Using the inversion $m \leq a\log m + b \implies m = O(a\log a + b)$, [36, Lemma A.2] we obtain

$$\begin{aligned} m &= O(WL\log(WL) + WL\log T + WL\log L) \\ &= O(WL\log(TWL))\,. \end{aligned}$$

Therefore,

$$\text{VCdim}(\mathcal{H}_{\text{cmp}}) = O(WL\log(TWL))\,. \tag{15}$$

Finally, under the assumption that the transformer has fewer blocks L than trainable parameters W, we have $\log(TWL) \leq \log(TW^2) = O(\log(TW))$, thus

$$\text{VCdim}(\mathcal{H}_{\text{cmp}}) = O(WL\log(TW))\,. \tag{16}$$

□

Lemma 6 (Warren's Bound, Lemma 2.1 in Anthony & Bartlett [1])**.** *Let $p_1, p_2, \ldots p_m$ be polynomials in W variables, each of degree at most D. If $W \leq m$, the number of distinct sign vectors $(sgn(p_1(\theta)), sgn(p_2(\theta)), \ldots, sgn(p_m(\theta))) \in \{-1, 0, 1\}^m$ as θ varies in $\mathbb{R}^W$ is at most $2(2emD/W)^W$.*

D Environment Setup

We construct a synthetic environment with large pools of ads and organic posts, whose embeddings are sampled from Gaussian clusters around randomly generated topic centroids. Each user is represented by a weighted average of these centroids, with user-specific weights capturing different interests across topics.

Each instance of a dataset is constructed by uniformly sampling M ads and N_O organic posts without replacement from their respective pools for a given user embedding. We then sample the ad qualities, bids, and organic values for the selected user–item pairs. The resulting item embeddings, ad qualities, and organic values form the context C supplied to the range generator.

To model user interests, ad qualities and organic values are sampled based on user-item similarities, with their distributions determined by the cosine similarity between the user's representation and the item's embedding. Ad qualities and organic post values follow lognormal/Weibull distributions characterized by the user-item similarities. However these distributions and their level of noise are picked so the corresponding advertiser and user welfare distributions stay balanced (Appendix Figure 4). Thus, items that are more closely aligned with a user's interests tend to have higher qualities or values, while the two components of welfare stay comparable. Advertiser values-per-click are sampled i.i.d. from an exponential or other distribution, once per ad when the global pool is created,

Figure 4: **Welfare distributions for the default environment.** Generated over 2,000 contexts with 256 random feasible feeds each. From left to right: (1) ad bid–quality products and organic values; (2) welfare per feed; (3) within-context standard deviation across feeds; (4) within-context maximum minus minimum among sampled feeds. Blue denotes advertiser quantities and green denotes organic quantities.

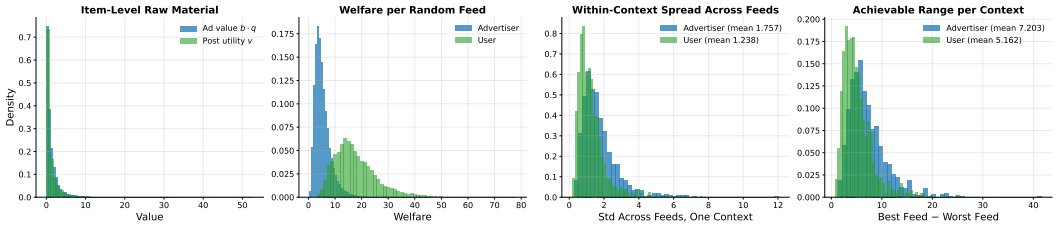
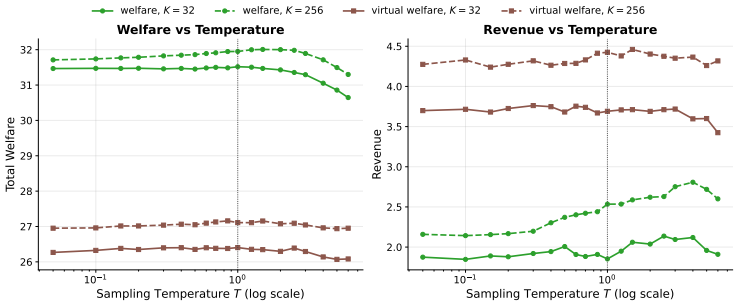


Figure 5: This figure shows the variation in performance of optimizing the softmax sampling temperature at test time on frozen models. For models trained on welfare, a higher temperature results in decreased welfare and increased revenue.



and are also sampled fresh during training. Figure 4 summarizes the resulting item values and feed-level welfare, including the variation across alternative feeds for the same user and candidate pool.

Engagement rates depend on an item's intrinsic quality, position, and surrounding content through the permutation-externality models defined in Section 2.

E Further MIRT Training Details

Our MIRT networks have 1.38M parameters with four encoder and two decoder layers with eight attention heads and 128 hidden size dimensions.

We optimize with AdamW [30], using an initial learning rate of 10^{-4}, weight decay of 10^{-4}, and gradient-norm clipping at 1.0. The learning rate follows a cosine decay schedule without restarts [29], over a maximum of 40 epochs with early stopping when there is no more than 0.3 percent improvement in the last 12 epochs. Each optimizer update uses 128 contexts, processed in two micro-batches with gradient accumulation.

F CGA and NGA Training Details

F.1 Contextual Generative Auctions

CGA [47] is an ad auction mechanism in which each context consists of a user's features, candidate ad features and their bids, and predicted click-through rates for that user-ad pair. Each context has 30 candidate ads which are competing for three slots. The architecture consists of three separately trained networks, an evaluator, a generator, and the PaymentNet.

The evaluator receives a historically shown ad sequence, user features, and existing individual-ad click predictions and is trained on historical click data to predict each ad's click-through rate. It processes the sequence using a bidirectional LSTM and combines its outputs with user features through an MLP to predict an externality multiplier for each displayed ad. This multiplier is applied to the existing ad-quality estimate to obtain the adjusted click-through rate. Unlike ours, their externality model captures only interactions among ads.

The generator uses a self-attention encoder to represent the candidate ads and a GRU decoder to select them one at a time. The frozen evaluator scores the generated sequences, providing rewards for policy-gradient training that aims to maximize the virtual welfare of a three-ad sequence. Finally, PaymentNet receives the selected ads' representations, bid information, and features derived from allocation probabilities and predicted click-through rates. It predicts a payment fraction between zero and one, which is multiplied by the advertiser's bid to determine its payment per click. PaymentNet is trained to increase revenue while penalizing ex-post regret, which measures how much advertisers could benefit by misreporting their bids. A tunable hyperparameter controls the strength of this penalty.

CGA ad-organic implementation. The original CGA paper considers only ads. To create a comparable baseline, we extend its candidate pool and architecture to include organic posts. In our implementation, the evaluator predicts each ad's click-through rate after accounting for position and surrounding content, and each organic post's welfare contribution after accounting for position. It is trained using our environment instead of historical data and learns to approximate the externality model used by the welfare function. In generator, we added separate input projections and learned item-type embeddings to map both ad and organic types into a shared representation space, so the CGA generator's self-attention is over the entire candidate pool. Finally, similar to the original paper, PaymentNet is trained on synthetic auction contexts with the generator and evaluator frozen. It predicts each displayed advertiser's payment per click as a fraction of its bid, to maximize revenue while penalizing ex-post regret. To estimate regret, we change an advertiser's bid, regenerate the feed, and recompute payments and click predictions to measure whether the advertiser benefits from misreporting.

Our CGA implementation has approximately 203,000 parameters and uses a two-layer Transformer encoder with hidden dimension 64, four attention heads, and feed-forward width 256, a 64-dimensional GRU decoder, a bidirectional LSTM evaluator with attention and separate ad and organic prediction heads, and a PaymentNet is a four layer MLP with 13,000 training parameters.

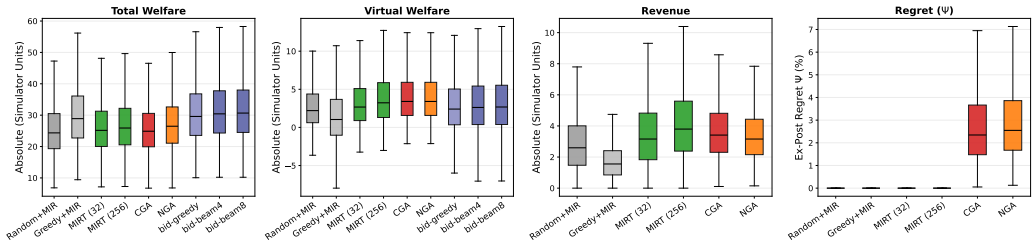
F.2 Non-autoregressive Generative Auctions

NGA [46] has a two network architecture, a generator for allocation and evaluator to score allocations and calculate payments. It extends the CGA architecture to also consider static surrounding organic content in its ad allocation. Unlike CGA's autoregressive GRU decoder, NGA uses item and position encoders to compute allocation scores for all item-position pairs, then create candidate feeds item by item through a decoder non-autoregressively. The evaluator scores generator's feed by producing click-through rates, conversion rates, and payment fractions per ad. These predictions are used to calculate a score based on the platform revenue for a feed.

NGA implementation. In our setting, organic posts are selected and reordered alongside ads, so we redesigned NGA's generator to encode ad-organic context and pass the shared item representations through an MLP to generate a feed. The evaluator is the same simulator-trained network we use for CGA, with separate heads for ad click-through rates and organic welfare contributions. In the welfare experiments we train the generator with policy gradients on the combined welfare. Payments come from the separately trained head of the evaluator that predicts payment fraction to maximize revenue and penalize regret.

Our NGA implementation has approximately 180,000 parameters and uses the same encoder, evaluator, and a four layer payment head, replacing the GRU decoder with learned 64-dimensional position embeddings and a three layer MLP that scores all item–position pairs in parallel in the generator.

Figure 6: Results of training models to optimize a virtual welfare objective (expected revenue) instead of welfare. MIR methods use Myerson payments instead of VCG payments. MIRT has a sampling temperature of 1, and CGA/NGA have $\rho = 1000$.



F.3 Regret

Regret is the largest increase in an advertiser's utility that misreporting its bid can produce. We estimate it for each advertiser selected under truthful bidding. We iterate over a grid of 20 bids for an advertiser holding the other bids fixed, and recompute the allocation and payments. We sum the positive gains across advertisers and normalize by the advertisers' truthful utility. During training, the regret penalty ρ is 1 unless otherwise noted.

G Additional Experimental Information

G.1 Virtual Welfare Experiments

Our mechanism design setting is quite challenging, as the combinatorial structure of feeds makes optimization difficult. However, because advertiser welfares are click-through rates multiplied by scalar values, it is a single-parameter setting, and therefore relatively well-behaved compared to general multi-parameter mechanism design [2; 32]. In particular, for our setting, assuming that CTR is nondecreasing in a bidder's bid, and that the lowest bidder type receives 0 utility, the corresponding Myerson payment rule is $t_i(B) = b_i q_i(x(B)) - \int_0^{b_i} q_i(x(z, B_{-i}))\, dz$.

A particularly useful property in single-parameter settings is that it is possible to define a quantity known as *virtual welfare*. The *virtual value* of a bidder with value v_i is $v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$. For any DSIC mechanism where the low type gets 0 utility, expected virtual welfare equals expected revenue. Assuming a property known as *regularity* of the distribution, the virtual-welfare-maximizing mechanism can be combined with an appropriate rule to maximize revenue, while also being DSIC. For our experiments, we also define a virtual-welfare-optimizing MIR mechanism. Equipped with the appropriate Myerson payment rule, this is also DSIC for regular distributions. Results on the default environment are in Figure 6. MIRT competes with other methods on welfare, outperforms baselines on revenue, and competes with CGA and NGA on revenue while being exactly strategyproof.

G.2 Additional Information from Experiments in Section 5

Table 4: Mean ($\pm$ CI across held-out contexts) absolute total revenue across held-out contexts under each externality function with default problem size and bid distribution. Highlighted cells are best-performing among mechanisms. Results in this table for all columns were generated with MIRT models trained for 40 epochs and temperature 1. CGA/NGA have $\rho = 1$, resulting in 10-16% regret.

Method	Prev. Organic	Additive	Activation	Attention	Multiplicative	Topic	Position
Random+MIR	2.01 ± 0.04	2.06 ± 0.04	2.74 ± 0.05	2.36 ± 0.05	2.13 ± 0.04	2.00 ± 0.04	2.08 ± 0.04
Greedy+MIR	1.43 ± 0.02	1.42 ± 0.02	1.74 ± 0.03	1.58 ± 0.03	1.43 ± 0.02	1.40 ± 0.02	1.47 ± 0.02
MIRT ($K = 32$)	2.13 ± 0.03	2.06 ± 0.03	2.25 ± 0.04	2.03 ± 0.03	2.28 ± 0.04	2.20 ± 0.04	2.22 ± 0.04
MIRT ($K = 256$)	2.64 ± 0.04	2.52 ± 0.04	2.78 ± 0.04	2.58 ± 0.04	2.81 ± 0.04	2.73 ± 0.04	2.72 ± 0.04
CGA	4.93 ± 0.04	3.89 ± 0.02	4.20 ± 0.03	4.51 ± 0.03	5.13 ± 0.04	4.35 ± 0.03	4.69 ± 0.03
NGA	4.32 ± 0.03	4.60 ± 0.03	5.24 ± 0.04	5.75 ± 0.04	4.85 ± 0.04	4.51 ± 0.03	5.12 ± 0.04

G.3 VARYING PROBLEM SIZES

We study how the methods scale with the numbers of candidate ads, organic posts, and feed positions. For each problem size, we retain the bid distribution, externality function, and organic-welfare weight from the reference environment. Results are in Table 5. Our method continues to perform competitively against zero regret baselines in both revenue and welfare across the problem sizes.

Table 5: Absolute total welfare and revenue (mean $\pm$ CI across held-out contexts) across different problem sizes, trained under the reference configuration. MIRT used a decoding temperature of 2.4, and CGA/NGA used $\rho = 1000$.

	(N_O=12, M=8, T=12) **3 ads, 9 posts**		**Default** (N_O=18, M=10, T=24) **6 ads, 18 posts**		(N_O=36, M=18, T=36) **9 ads, 27 posts**	
Method	**Welfare**	**Revenue**	**Welfare**	**Revenue**	**Welfare**	**Revenue**
Random+MIR	17.838 ± 0.105	2.155 ± 0.039	27.155 ± 0.214	2.047 ± 0.039	35.043 ± 0.125	2.785 ± 0.057
Greedy+MIR	20.048 ± 0.122	1.810 ± 0.029	30.708 ± 0.240	1.437 ± 0.023	44.619 ± 0.172	2.102 ± 0.035
MIRT (K=32)	20.885 ± 0.128	2.585 ± 0.037	32.020 ± 0.244	2.258 ± 0.036	46.882 ± 0.182	3.025 ± 0.044
MIRT (K=256)	21.346 ± 0.131	3.077 ± 0.040	32.500 ± 0.246	2.799 ± 0.037	47.904 ± 0.186	4.017 ± 0.048
CGA	20.222 ± 0.117	1.955 ± 0.013	31.084 ± 0.238	3.125 ± 0.021	47.076 ± 0.172	6.484 ± 0.050
NGA	20.238 ± 0.117	1.788 ± 0.011	31.095 ± 0.238	2.411 ± 0.013	47.302 ± 0.172	6.761 ± 0.046
bid-greedy	20.260 ± 0.117	n/a	31.121 ± 0.238	n/a	47.391 ± 0.172	n/a
bid-beam4	20.890 ± 0.123	n/a	31.956 ± 0.243	n/a	49.028 ± 0.180	n/a
bid-beam8	21.145 ± 0.127	n/a	32.205 ± 0.244	n/a	49.516 ± 0.187	n/a

Table 6: Absolute metrics across methods for the reference configuration as described in Section 5. Means with approximate 95% confidence intervals across held-out test contexts in the test set, alongside medians. Results in this table were generated with 40 epochs of training for each MIRT model with inference time temperature 2.4. For CGA/NGA, $\rho = 1000$.

	Organic Welfare		**Advertiser Welfare**		**Total Welfare**		**Revenue**		**Regret (Ψ)**	
Method	**Mean $\pm$ CI**	**Median**	**Mean $\pm$ CI**	**Median**	**Mean $\pm$ CI**	**Median**	**Mean $\pm$ CI**	**Median**	**Mean $\pm$ CI**	**Median**
Random+MIR	18.937 ± 0.202	17.307	8.217 ± 0.097	7.362	27.155 ± 0.214	25.739	2.047 ± 0.039	1.687	0.000 ± 0.000	0.000
Greedy+MIR	20.752 ± 0.217	19.089	9.956 ± 0.112	8.975	30.708 ± 0.240	29.177	1.437 ± 0.023	1.237	0.000 ± 0.000	0.000
MIRT (32)	20.554 ± 0.214	18.877	11.466 ± 0.124	10.412	32.020 ± 0.244	30.486	2.258 ± 0.036	1.949	0.000 ± 0.000	0.000
MIRT (256)	20.562 ± 0.214	18.910	11.938 ± 0.128	10.865	32.500 ± 0.246	30.904	2.799 ± 0.037	2.491	0.000 ± 0.000	0.000
CGA	20.915 ± 0.217	19.239	10.169 ± 0.112	9.215	31.084 ± 0.238	29.558	3.125 ± 0.021	3.006	0.105 ± 0.002	0.085
NGA	20.874 ± 0.217	19.199	10.221 ± 0.112	9.278	31.095 ± 0.238	29.545	2.411 ± 0.013	2.349	0.112 ± 0.002	0.091
bid-greedy	20.863 ± 0.217	19.196	10.257 ± 0.112	9.309	31.121 ± 0.238	29.592	n/a	n/a	n/a	n/a
bid-beam4	20.847 ± 0.216	19.165	11.109 ± 0.119	10.160	31.956 ± 0.243	30.421	n/a	n/a	n/a	n/a
bid-beam8	20.830 ± 0.216	19.151	11.374 ± 0.122	10.395	32.205 ± 0.244	30.668	n/a	n/a	n/a	n/a