A Probabilistic Framework to Learn Auction Mechanisms via Gradient Descent

Olivier Jeunen*, **Lampros Stavrogiannis, Amin Sayedi, Ben Allison**

Amazon

Abstract

The online advertising industry heavily relies on auction mechanisms to allocate impression opportunities to advertisers, and price them. In real-world auctions, we want to maximise *welfare* for auction participants while being *incentive-compatible* (allowing bidders to bid truthfully), but we may also wish to include other constraints such as improving revenue for the seller, or mitigating externalities on non-bidding third-parties. Whilst the first two goals can be attained by drawing on auction theory, more general constraints lead to objectives that cannot be optimised exactly.

In this work, we propose a general framework to learn incentive-compatible auction mechanisms via gradient descent. Following related work on "neural auctions", we resort to approximate continuous relaxations of the discrete sorting operation that render our objectives differentiable and thus amenable to optimisation in modern Deep Learning frameworks. We inject Gumbel-noise into the allocation rule, leading to a smoother loss surface and a stochastic auction mechanism under the Plackett-Luce model. Additionally, we derive a pricing rule that preserves incentive compatibility in this probabilistic setting. We present experiments for a known difficult case for exact optimisation: that of maximising revenue while maintaining optimal welfare and incentive compatible pricing. Results show we are able to improve on classical mechanisms, even under non-deterministic allocation.

1 Introduction & Motivation

Online advertising opportunities are sold through advertising auctions, where advertisers bid for a given impression opportunity, and the auctioneer decides (1) which ad to show, and (2) what price the advertiser needs to pay. Auctioneers can have multiple objectives, including: (1) for the auction to be efficient (i.e. welfare-maximising), (2) for the auction to be incentive-compatible (i.e. it is a dominant strategy for bidders to bid truthfully), and (3) for the seller's revenue to be optimised.

Objectives (1) and (2) are achieved by classical Vickrey (or *sealed-bid second-price*) auctions [38]. However, the auctioneer may have a range of other objectives: to optimise (or at least consider) revenue for the seller; to limit externalities or consider the welfare of non-bidding parties; and to ensure statistical properties of the auction if the mechanism is repeated many times. Even for simple revenue maximisation, the optimal mechanism is complicated. Myerson provides the revenue-optimal single-item auction [31]. However, real-world implementations of Myerson auctions require knowledge of the bidders' valuation distributions. For most multi-bidder and multi-item auction set-ups, exact revenue-optimising auction mechanisms are yet to be discovered.

In this paper we consider auctions with a broader set of objectives, where exact optimisation is infeasible and mechanisms are learned rather than specified. Computational efforts to design mechanisms are not new, but typically focus on a specific use-case [37]. Advances in machine learning have allowed auction mechanisms to be *learned* to optimise flexible objectives. Embracing the "deep learning" paradigm, Duetting et al. propose specific neural network architectures and objectives to learn general-purpose, approximately incentive-compatible auctions [11]. These advances have found their way into advertising applications, and several data-driven approaches to optimise auction mechanisms have been proposed in recent years [13, 25, 42]. We consider extensions in a similar vein, allowing approximate optimisation of arbitrary objectives for the auction. Our work additionally enables stochasticity of allocation to facilitate exploration, escape feedback loops, enable counterfactual reasoning and increase diversity.

We propose a general framework to learn auction mechanisms through gradient descent. Following earlier work [25], we adopt a continuous relaxation of the inherently non-differentiable discrete sorting operation. Beyond that, we propose to smooth the resulting objective by injecting Gumbel-distributed noise [4], which gives rise to a probabilistic lens under which stochastic auction mechanisms can be understood. We further provide an incentive-compatible pricing rule for the class of auctions we consider. We empirically validate our proposed methods using real-world data from an online advertising system, and show that: (1) auction mechanisms optimised through our framework outperform traditional mechanisms, (2) the injection of Gumbel-noise yields additional improvements, and (3) our framework naturally accommodates stochastic allocation without impact to welfare or revenue We illustrate our approach by

*Author is now at ShareChat, UK.

focusing on a known challenging case for manual auction design, that of optimising revenue for the seller. However, we note that our framework naturally accommodates any objective that can be formalised into a loss function, and we anticipate real world settings will use more complex and balanced objectives.

2 Related Work

We briefly review the related literature on learned auction mechanisms. For a more general overview of auction theory, see [22]. For accounts more specific to advertising applications, refer to [7].

Duetting et al. were the first to frame auction design as optimisation of a parameterised function [11]. They propose neural allocation and payment rules for various setups, depending on bidders' valuations for bundles of items being sold. In the style of Myerson [31], they assume access to the bidders' valuation distributions, using them to formulate a regularisation term that penalises non-incentive-compatible mechanisms. In contrast, our work focuses on single-item auctions that are incentive-compatible in an *exact* manner. We do *not* require access to (estimated) valuation distributions, and derive incentive-compatible prices directly from the structure of the allocation rule.

Zhang et al. propose the "Deep GSP" framework to learn ranking scores in GSP-style auctions, optimised for flexible objectives [42]. They do not directly consider gradient-based optimisation of inherently non-differentiable quantities such as revenue, but adopt a deterministic policy search approach instead. They regularise the learned allocation probability to be monotonic with respect to the bid—but as this is not an exact approach, the incentive compatibility constraint is not guaranteed. In contrast, we leverage the structure of the allocation policy to allow for exact incentive-compatibility, for either stochastic or deterministic auctions.

Driven by the observation that Myerson's optimal mechanisms are significantly simplified in cases where bidders are homogeneous [31], Golrezaei et al. propose the "boosted second price" framework to optimise revenue in the more realistic case where bidders are heterogeneous [13]. Even though their optimisation is data-driven, it focuses on a specific instance of deterministic boosted virtual bids, and is not amenable to gradient-based optimisation.

The work of Liu et al., introducing "neural auctions" [25], bears the closest resemblance to our work and is the main source of inspiration for this work. Notwithstanding this, there are several key differences. Indeed, whereas they focus on specific neural network architectures, we allow for any parameterisation within our framework. Because of the feature set they adopt, they deal with approximately incentive-compatible auctions, whereas we guarantee the constraint in an exact manner. Finally, the injection of Gumbel-noise improves the smoothness of our objective and helps gradient-based optimisation, whilst allowing for stochastic auctions, which were not considered by Liu et al. [25].

3 Problem Setting & Methodology

We deal with a single-slot auction setup in online advertising, similar in spirit to the one presented by Jeunen, Murphy, and Allison [20]. The main difference is that we do not expect bidders to model click-probabilities themselves. Instead, modelling click probabilities is the responsibility of the auctioneer. Nevertheless, our framework accommodates the former use-case, as we discuss further below.

Ad opportunities are denoted as $x \sim \mathsf{P}(X)$ with support in $\mathcal{X}$. A feature vector to represent x is denoted in bold as $\boldsymbol{x} \in \mathbb{R}^d$, with d the dimensionality of the feature space. The ad catalogue $\mathcal{A} \coloneqq \{(a_i, v_{a_i})\}_{i=1}^N$ consists of N ads a_i that are tied to a private valuation $v_{a_i} \in \mathbb{R}^+$ representing the advertiser's willingness-to-pay for a conversion-event after an impression (in USD). Note that this formulation is general: "conversion-events" subsume views, clicks, sales, incremental sales, or other outcomes resulting from rendering an ad. Competing advertisers can target heterogeneous outcomes—in an attempt to minimise notational clutter we do not explicitly model this here. Advertisers define their target outcome and valuation, and are subsequently only charged when their event occurs. We denote this with a binary random variable C. In this formulation, an advertiser's valuation of an impression opportunity is dependent upon the probability of C occurring. We can write the expected social welfare ω of showing ad a_i in context x as:

$$\mathbb{E}[\omega | A = a_i; X = x] \coloneqq v_{a_i} \cdot \mathsf{P}(C = 1 | A = a_i; X = x). \tag{1}$$

Note that this formulation assumes that the self-reported advertiser valuations v_{a_i} are *truthful*. Naturally, for general conversion-events the distribution $\mathsf{P}(C)$ is unknown, and it can be arbitrarily hard to estimate. Still, the auctioneer typically approximates this distribution by a parameterised model $\widehat{\mathsf{P}}_\theta(C|A;X) \approx \mathsf{P}(C|A;X)$ that gives rise to a welfare estimate that can inform an auction mechanism. For completeness:

$$\widehat{\omega_\theta}(a_i, x) \coloneqq v_{a_i} \cdot \widehat{\mathsf{P}}_\theta(C = 1 | A = a_i; X = x). \tag{2}$$

Under the assumption that $\widehat{\mathsf{P}}_\theta(C)$ is an unbiased estimator of $\mathsf{P}(C)$, we can say that bidding according to $\widehat{\omega_\theta}$ represents *truthful* bidding on the advertisers' behalf in expectation. Note that is a strong assumption. Nevertheless, because it is at the heart of many web-based platforms, it has given rise to a rich body of literature dedicated to improving these models [2, 16, 18, 19, 27, 29, 39].

In what follows, we assume $\widehat{\mathsf{P}}_\theta(C)$ to be given and fixed, and place our focus on the *auction mechanism* instead.[1]

3.1 Auction Mechanisms

An auction mechanism is a procedure that leads to two decisions: (1) which bidder wins the impression opportunity, and (2) at what price. Common examples in online advertising include first-price auctions (i.e. the highest bidder wins,

[1]Note that our formulation subsumes auction setups where the bidders themselves estimate $\mathsf{P}(C)$: we can assume that their self-reported valuation v_{a_i} represents their estimate, and their targeted "conversion-event" is then simply constant with $\mathsf{P}(C = 1) \equiv 1$.

paying their bid), and second-price auctions (i.e. the highest bidder wins, paying the runner-up bid). In what follows, we describe a general framework to reason about auctions. We will assume w.l.o.g. that we have a fixed ad catalogue $\mathcal{A}$, to reduce notational clutter.

Allocation is a stochastic decision rule mapping impression opportunities to a distribution over ads $\pi : \mathcal{X} \rightarrow \Delta^N$ (i.e. the N-dimensional probability simplex). The payment decision is a rule mapping impression opportunities to (a distribution over) prices per ad $\psi_i : \mathcal{X} \rightarrow \mathbb{R}^+$. The price ψ_i for an ad a_i is dependent on competing ads $\mathcal{A}_{\setminus i}$; we suppress this for notational convenience. An auction mechanism is the conjunction of allocation and pricing rules $\phi \coloneqq \{\pi, \psi\}$. Note that deterministic auction mechanisms are allowed in this setup when π is a degenerate distribution.

We can formalise metrics of interest that are used to evaluate auction mechanisms in practice. The expectation of *welfare* ω is:

$$\mathop{\mathbb{E}}_{i \sim \pi(x)}[\omega | \Phi = \phi] \coloneqq \int v_{a_i} \mathsf{P}(C = 1 | A = a_i; X = x) \mathrm{d}i \, \mathrm{d}x. \tag{3}$$

The expectation of *revenue* ρ is:

$$\mathop{\mathbb{E}}_{i \sim \pi(x)}[\rho | \Phi = \phi] \coloneqq \int \psi_i(x) \mathsf{P}(C = 1 | A = a_i; X = x) \mathrm{d}i \, \mathrm{d}x. \tag{4}$$

Unless we run online experiments where we actively sample $i \sim \pi(x)$, these quantities are not observable. Nevertheless, we can obtain Monte Carlo samples from ϕ, and generate value-based counterfactual estimates for these quantities via the Direct Method by plugging in $\widehat{\mathsf{P}}_\theta(C|A;X)$ for $\mathsf{P}(C|A;X)$, among other approaches [10]. If we have access to a dataset containing logged auctions $\mathcal{D}$, we can use this to obtain counterfactual estimates of expected revenue for any auction mechanism, denoted as $\widehat{\rho}(\phi)$.

If we can evaluate any auction mechanism ϕ with respect to a certain metric of interest, a natural next question to ask is whether we can *optimise* ϕ to maximise that metric, or even multiple objectives. In what follows, we will focus on the classical task of maximising the revenue an auction mechanism yields to the seller. As such, we aim to find:

$$\phi^\star = \mathop{\arg\max}_{\phi \in \Phi} \widehat{\rho}(\phi). \tag{5}$$

3.2 Incentive-Compatible Auction Mechanisms

If we do not place any restrictions on the space of auction mechanisms Φ to consider, the problem at hand might seem trivial at first glance. If we want to maximise the expected welfare of allocated ads, we simply show the ads with maximal predicted welfare $\widehat{\omega_\theta}$ (Eq. 2). If an auctioneer want to maximise revenue, we would simply increase the prices charged to advertisers $\psi_i(x)$. Rational agents would prefer *not* to participate in this auction as soon as prices exceed their personal valuations. Indeed, these valuations v_{a_i} are called advertisers' *willingness-to-pay* in the economics literature. This gives rise to the *Individual Rationality (IR)* constraint, stating that no auction participant will ever pay more than their (reported) valuation [38].

Adhering to the IR constraint, we cannot increase prices beyond advertisers' valuations. As such, when the goal is to maximise revenue under an IR constraint, we can simply charge advertisers their reported willingness-to-pay for every conversion event: $\psi_i(x) = v_{a_i}$. This is a first-price auction and is still problematic: advertisers do not have an *incentive* to truthfully report their valuation under these dynamics [12]. When advertisers strategically misreport their valuations, it obfuscates identifiability of the welfare that is generated through the auction mechanism (Eq. 1) — which clearly affects revenue as well. Furthermore, such strategic behaviour can lead to market instabilities we would generally prefer to avoid [15]. As such, following earlier work [11, 13, 25, 31, 42], we focus on auction mechanisms that are *dominant-strategy incentive-compatible (DSIC)* as well as IR [38]. That is, bidding truthfully is optimal for all advertisers, regardless of competitors' strategies.

Second-price auctions are arguably the most famous type of auction mechanism in the class we consider, and have seen broad adoption in the advertising industry. Here, the winning bidder pays the *runner-up* bid, which can be interpreted as the minimum bid the advertiser could have placed and still won the opportunity (sometimes referred to as the *critical price*). In the general case, IC auction mechanisms need to adhere to: (1) monotonicity of the allocation probability to the submitted bid, and (2) the payment needs to be the critical price. This result holds for *utility* and *value* maximisers alike in single-slot settings [13, 40].

The monotonicity of the allocation rules π (w.r.t. valuations v_{a_i}) can be enforced through its functional form. This can either be done through a min-max partially monotonic neural network [9] as adopted by Liu et al. [25], other neural approaches [6, 36], or by adopting any other monotonic functional form for π that does not rely on a neural network parameterisation.

When monotonicity is guaranteed, we additionally need an efficient way to compute the critical price. For neural network parameterisations, Liu et al. provide one possible approach that inverts the rank score function [25, §3.3, Eq. 7]. Other function forms exist that facilitate this inversion. As an example, we consider "*bid squashing*", prevalent in the sponsored search literature [23]. Bid squashing introduces a parameter $\alpha \in \mathbb{R}^+$ that *squashes* the estimated conversion probability. It transforms rankings scores as:

$$f_\alpha(a_i, x) \coloneqq v_{a_i} \widehat{\mathsf{P}}_\theta(C = 1 | A = a_i; X = x)^\alpha. \tag{6}$$

Now, we can denote the ad that wins the auction $a^\star$ as:

$$a^\star = \mathop{\arg\max}_{a_i \in \mathcal{A}} f_\alpha(a_i, x). \tag{7}$$

A deterministic allocation rule is:

$$\pi(x)|_{a_i} = \begin{cases} 1 & \text{iff } a_i = a^\star, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We define a_{sp} as the runner-up or *second-price* ad, that is:

$$a_{\mathrm{sp}} = \mathop{\arg\max}_{a_i \in \mathcal{A} \setminus a^\star} f_\alpha(a_i, x). \tag{9}$$

This gives rise to the following critical price:

$$\psi_i(x) = \frac{v_{a_{\mathrm{sp}}} \widehat{\mathsf{P}}_\theta(C=1|A=a_{\mathrm{sp}}; X=x)^\alpha}{\widehat{\mathsf{P}}_\theta(C=1|A=a_i; X=x)^\alpha}. \tag{10}$$

Here, the monotonicity constraint holds by design, and the critical price is available analytically. Furthermore, our definitions of $a^\star$ and a_{sp} ensure that the impression cost can never exceed an advertisers' willingness to pay: $\psi(x) \leq v_{a^\star}$. Indeed, bid squashing is IC and IR. Although squashing typically uses a single global parameter α, we can parameterise α as a function of the opportunity x, and substitute all occurrences with $\alpha(x)$ in Eqs. 6–10. In our experiments, we will refer to this variant as "personalised squashing" (`P-Squash`).

All of the IC and IR auction mechanisms discussed above (`(P-)Squash` as well as more general instantiations of our framework) can be extended to allow for stochastic allocations; we discuss this in detail in Section 3.4. Because the single-parameter squashing parameterisation allows us to illustrate behaviour visually, we will focus on it as a running example in the following subsections.

3.3 Towards a Differentiable Objective

We have now formulated our general problem setting, and can focus on the task at hand: *learning* the auction mechanism $\phi^\star$ that maximises expected revenue. Furthermore, we restrict $\phi^\star$ to be both IR and IC. For illustratory purposes, we adopt the squashing parameterisation laid out in Section 3.2.

For simplicity, assume we have no contextual covariates and a fixed catalogue of $|\mathcal{A}| = 3$. Furthermore, we have valuations $\boldsymbol{v} = \{1.25, 1.00, 1.50\}$, and conversion probabilities $\boldsymbol{c} = \{0.75, 0.90, 0.50\}$. Assume these probabilities are known and exact. In this simplified example, we can compute our metrics of interest: expected welfare ω, and expected revenue ρ. Varying the squashing parameter α, we plot these quantities in Figures 1(a) and 1(b), respectively. As expected, the parameter α affects both welfare and revenue. It is clear from Figure 1(b) that the traditional second-price auction mechanism of $\alpha = 1.0$ is *not* revenue-maximising in this toy example setting, and that setting either $\alpha \approx 0.45$ or $\alpha \approx 1.2$ can yield significant improvements. However, the grid-search optimisation does not generalise or scale to real-world settings where we wish to *learn* the parameters that maximise our objective. Instead, we wish to rely on widely established gradient-based optimisation techniques to find $\phi^\star$. Gradient-based techniques are not directly applicable to optimising arbitrary objectives because of the non-continuous and non-differentiable nature of ρ. Discontinuities are a direct result of the $\arg\max$ operations in Eqs. 7 & 9. To allow for gradient-based optimisation, we must resort to alternatives. Liu et al. [25] adopt the `NeuralSort` approximation, a continuous relaxation of the sorting operation [14]. Recently, Prillo and Eisenschlos proposed `SoftSort` [35], an extension that improves computational complexity. We will adopt `SoftSort` for the remainder of this work. `SoftSort` includes a hyper-parameter τ that controls the exactness of the approximation. As τ approaches 0, the relaxation becomes more exact. Nevertheless, an exact approximation can give rise to a highly non-smooth loss surface that can still hinder gradient-based optimisation. In practice, τ is a hyper-parameter that needs to be tuned for the downstream task. Figure 1(c) shows the true discontinuous values of ρ given α, along with the approximation $\tilde{\rho}$ for varying values of τ.

3.4 Towards a Smooth Objective

We have obtained a differentiable approximation to the expected revenue of an auction. Leveraging automatic differentiation, gradient descent and well-known optimisation libraries such as Tensorflow [1] and PyTorch [33], this enables large-scale learning of incentive-compatible auction mechanisms to maximise inherently non-differentiable metrics. Nevertheless, gradient-based optimisation has known failures for non-convex objectives. This is clear, even for the toy example approximation $\tilde{\rho}$ presented in Figure 1(c): because of its non-smoothness and local minima, the initialisation for any optimisation algorithm will influence its convergence and the final learned mechanism. When local optima are only approximations to the true objective, converging on them can be especially problematic. Given that we depend on approximations due to the model $\widehat{\mathsf{P}}_\theta(C)$ as well as the `SoftSort` relaxation, this is exactly the situation we find ourselves in.

Recent work in the related field of Learning-to-Rank (LTR) has dealt with similar situations, where a ranking model is learnt to maximise some non-differentiable listwise ranking metric through approximations (e.g. NDCG@k) [4]. They propose stochastic allocations via the Gumbel-max trick [28], which improve the generalisation capabilities of models learnt to optimise approximate ranking losses. By adding independently drawn noise variables from a $\mathrm{Gumbel}(\mu = 0, \beta)$-distribution and adding them to the ranking scores (Eq. 6), we can emulate sampling from a Plackett-Luce distribution that is defined by the ranking scores [26, 34, 41]. Indeed, it corresponds to repeated sampling without replacement from a categorical distribution where the probabilities are defined by a temperature-controlled softmax with parameter $\beta > 0$. This approach has seen widespread success in LTR scenarios since [17, 32]. For completeness, given our running example, this replaces the deterministic ranking scores from Eq. 6 with the following stochastic variant in every forward pass:

$$\begin{aligned} f_\alpha^G(a_i, x) &\coloneqq v_{a_i} \widehat{\mathsf{P}}_\theta(C=1|A=a_i; X=x)^\alpha + G_i, \\ &\text{where } G_i \sim \mathrm{Gumbel}(0, \beta). \end{aligned} \tag{11}$$

Aside from this (very efficient) sampling procedure, no further changes to the learning setup are necessary. Figure 2 illustrates how this transforms the objective function, for 12 randomly sampled instantiations of G_i, with $\beta = 0.01$. We expect this stochasticity to help gradient-based optimisers, and lead to more robust learnt auction mechanisms. In our experiments, we will refer to an auction mechanism optimised for such a stochastic objective as "Gumbel-Personalised squashing" (`GP-Squash`).

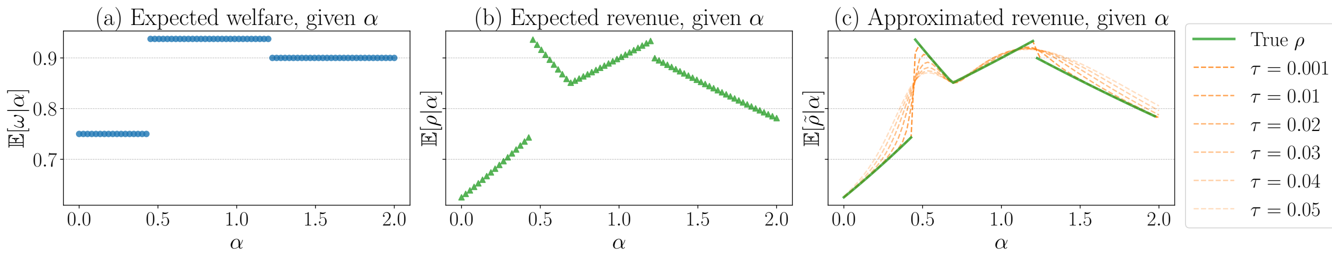

Figure 1: Expected welfare, revenue, and approximated revenue (using `SoftSort`) for a squashed second-price auction, varying the squashing parameter α and the approximation parameter τ. More details on this toy example can be found in §3.3.

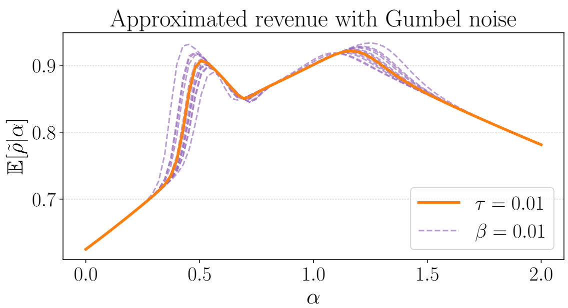

Figure 2: Approximated revenue for varying values of α, obtained through `SoftSort` with Gumbel-distributed noise.

Auction Mechanism	$\Delta\widehat{\rho}$(train)	$\Delta\widehat{\rho}$(test)
`Second-Price`	100%	100%
`Squash`	+0.01%	+0.01%
`P-Squash`	+1.34%	+1.10%
`GP-Squash`	+1.42%	+1.25%

Table 1: Relative improvements in estimated revenue for auction mechanisms learnt with and without Gumbel-noise injection, compared to traditional auction mechanisms.

Incentive-Compatible Pricing in Stochastic Auctions We have introduced the injection of Gumbel-noise to increase the stability of the optimisation procedure, even for deterministic auctions—but its interpretation as sampling rankings from a Plackett-Luce model enables us to model auction mechanisms with stochastic allocation rules. Stochastic auctions have a range of desirable properties: they allow us to escape from feedback loops, reason about counterfactual allocation decisions, and increase diversity compared to deterministic rules. However, with stochastic allocation we can no longer directly use the second-price payment rule introduced in Eq. 10. If we use the pricing rule above, the winning bidder is incentivised to *increase* their reported valuation to increase the allocation probability for their ad (as the softmax-transform now allocates additional probability mass to the inflated value), without increasing the price (when it is still solely defined by the runner-up).

In order to preserve incentive-compatibility, we need to adapt the pricing rules $\psi_i(x)$. An auction mechanism is IC when the bidder's expected utility with respect to their bid is maximised for truthful bidding. This implies that the derivative of their utility with respect to the bid needs to be zero when v_{a_i} is truthful. This gives rise to a differential equation (dependent on the allocation rule π), that can be solved to obtain the IC pricing mechanism (see e.g. [30, §2.1, Ex. 1]). High-performance open-source software packages like JAX can be used to efficiently obtain ψ_i for any parameterisation of π [3]. Although feasible, this is a costly approach.

Alternatively, we can adopt a stochastic pricing mechanism such as the one proposed by Meek, Chickering, and Wilson [30, §2.2]. The main advantage of this approach is its computational simplicity and efficiency compared to solving the necessary differential equations for deterministic IC pricing. For a set of sampled Gumbel-instantiations G_i, assign the winning and runner-up ads according to Eq. 11. We can then compute IC prices as follows:

$$\psi_i(x) = \frac{v_{a_{\mathrm{sp}}} \widehat{\mathsf{P}}_\theta(C=1|A=a_{\mathrm{sp}}; X=x)^\alpha + G_{\mathrm{sp}} - G_i}{\widehat{\mathsf{P}}_\theta(C=1|A=a_i; X=x)^\alpha}. \quad (12)$$

This approach extends to general forms for the allocation rule π, and is not solely applicable to the squashing parameterisation we have adopted for illustratory purposes.

4 Experiments & Discussion

We run experiments to validate our approach on real data. We use a dataset that contains logged information about real-world advertising auctions. Our data contain information about the ad catalogue $\mathcal{A}$, contextual covariates $\mathcal{X}$, and model estimates $\widehat{\mathsf{P}}_\theta(C)$. To the best of our knowledge, no such dataset is publicly available at the time of writing. Recent work introduces a simulation environment that currently encodes the $\widehat{\mathsf{P}}_\theta(C)$-model as bidder-side instead of auctioneer-side logic [20]. As such, even though it might be adaptable to our use-case, it is not directly applicable. Because of the lack of publicly available data and simulation environments that fit our needs, we use a proprietary dataset containing advertising impression and auction logs. Note that the approximation in reported evaluation metrics only stems from the use of $\widehat{\mathsf{P}}_\theta(C)$ instead of the true $\mathsf{P}(C)$, as we can use the non-differentiable but exact `argsort` operation here.

Methods and baselines. We evaluate the classical second-price mechanism [38] as well as the traditional squashing parameterisation with a single global parameter optimised through grid search [23]. We compare to two learned variants: `P-Squash` and `GP-Squash`. For both, we parameterise $\alpha_{\Xi}(x)$ as a 3-layer feed-forward neural network with parameters Ξ, ELU activations [8] and a Softplus output, optimised with Adam [21] in PyTorch [33]. We initialise the models to imitate the second-price mechanism ($\alpha_{\Xi}(x) \equiv 1, \forall x \in \mathcal{X}$). The input features $\mathcal{X}$ are a set of aggregation functions over the model estimates $\widehat{\mathsf{P}}_{\theta}(C)$ for all ads. Indeed, to preserve the IC property, the model *cannot* use advertiser valuations as its inputs.

When optimising `P-Squash`, we do not inject Gumbel-distributed noise into the objective function, as described in Section 3.4. In contrast, `GP-Squash` is optimised for this noisy, stochastic objective. Note that `GP-Squash` still remains a deterministic auction mechanism: the injected Gumbel-noise was only used to smooth the loss surface during learning, and no further noise was injected at evaluation time (unless explicitly noted otherwise).

Efficacy of Learnt Auction Mechanisms (RQ1–2)

Because we have access to the model estimates $\widehat{\mathsf{P}}_{\theta}(C)$, we can obtain value-based counterfactual estimates of welfare and revenue under any auction mechanism that can be expressed as an allocation policy π and pricing policy ψ (see Section 3.1 for more detail). As a result, these counterfactual revenue estimates allow us to evaluate the efficacy of auction mechanisms learnt via our framework.

We perform an 80-10-10% train-validation-test split, optimise hyper-parameters through grid search to maximise revenue estimates on the validation set, train models until convergence on the training set and report revenue estimates on the test set. Table 1 shows relative improvements of the learnt mechanisms over the classical Vickrey auction.

As expected, we can see positive improvements for both the `P-Squash` and `GP-Squash` variants. This result is encouraging, as we have so far only dealt with very simplistic parameterisations: the squashing approach, and feed-forward neural networks. We expect extensions to yield further improvements, e.g. a richer feature set, as well as more advanced modelling techniques for both the allocation and pricing policies [24]. We additionally observe that the injected Gumbel-noise has beneficial effects as a regularisation term—reducing the gap between the estimated revenue on the training and test sets. Finally, because of improved stability during the optimisation procedure, we observed faster convergence rates for `GP-Squash` as compared to `P-Squash`.

Revenue and Diversity in Stochastic Auctions (RQ3)

There are several motivations for stochastic allocation. It can help increase the diversity of ads that are shown, and adds a level of exploration to the collected data that is used to train future models, which can be beneficial as well [5]. Nevertheless, these advantages can quickly be overshadowed if they come at significant decreases in the short-term value generated by the auction (e.g. welfare, revenue). We now estimate the impact on welfare and revenue that comes with varying levels of stochasticity, by varying the scale-parameter β of the Gumbel-distribution. Note that to consider revenue under this stochastic allocation mechanism, we adopt the IC pricing rule defined in Eq. 12.

Figure 3 shows the relative impact on estimated welfare ($\widehat{\omega}$), estimated revenue ($\widehat{\rho}$), and the percentage of allocation decisions that would have been impacted by adopting the auction mechanism learnt under `GP-Squash` with the scale of the Gumbel-distribution set to β when deploying the policy. We can observe that for a wide range of β, the percentage of impacted allocation decisions can be significantly increased (nearly 20%) without significant impact on welfare or revenue. This result highlights the efficacy of the learnt auction mechanism and the market: when the ranking scores for the top and runner-up ads are very close to one another, this allows for the Gumbel-noise to change their ordering without significantly impacting the welfare or revenue that the auction mechanism yields. In these cases, stochastic allocation is highly preferable due to the aforementioned benefits. Naturally, for ever-increasing levels of noise, we do observe a negative impact on both ω and ρ.

These empirical results show that our proposed framework allows us to learn stochastic auction mechanisms that can significantly increase the revenue obtained by traditional auction mechanisms, with minimal impact on welfare.

5 Conclusions & Future Work

In this work, we have presented a general probabilistic framework that allows well-established gradient-based optimisation techniques to be used, to learn incentive-compatible auction mechanisms that maximise inherently non-differentiable objectives. Complementary to existing work, we propose to inject Gumbel-noise into the ranking scores as a regularisation technique. Its probabilistic interpretation as sampling rankings according to a Plackett-Luce model allows us to learn principled stochastic allocation rules, for which we derive an efficiently computable, incentive-compatible pricing mechanism.

For a simple class of auction mechanisms we call "Personalised Squashing" (`P-Squash`) and using logged data from a real-world advertising domain, we empirically validate that (1) auction mechanisms learnt through our framework are effective in improving the objectives they were optimised for, (2) the injection of Gumbel-noise is effective as a regularisation and smoothing technique, and (3) our framework enables stochasticity with minimal impact on welfare and revenue, but increased diversity and exploration.

Our work is only a step in the right direction, and there are many possible avenues to consider as future work. First, extending our framework to multiple objectives (beyond simple scalarisation techniques) will make it more applicable in real-world systems. Second, specific instantiations of our general framework leveraging advanced neural network architectures can lead to larger performance improvements. Third, we wish to validate our framework on open-source benchmarks that enable reproducibility for the wider research community [20]. Further development is necessary to achieve these goals.

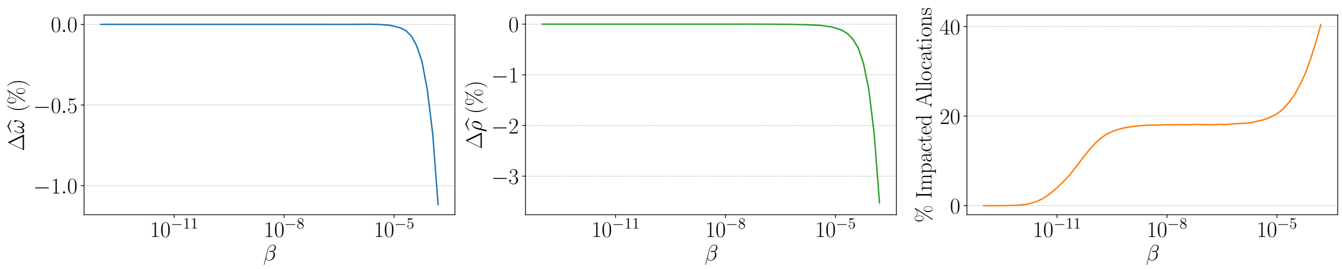

Figure 3: Relative impact on estimated welfare, estimated revenue and the percentage of impacted allocation decisions for varying values of β, the scale of the injected Gumbel-noise. We observe that, for a wide range of β, we can observe significantly diversified allocation decisions (nearly 20%), for no discernible impact on welfare or revenue.

References

[1] Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; and Zheng, X. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283. USENIX Association.

[2] Anil, R.; Gadanho, S.; Huang, D.; Jacob, N.; Li, Z.; Lin, D.; Phillips, T.; Pop, C.; Regan, K.; Shamir, G. I.; Shivanna, R.; and Yan, Q. 2022. On the Factory Floor: ML Engineering for Industrial-Scale Ads Recommendation Models. In *Proc. of the ORSUM Workshop at the 16th ACM Conference on Recommender Systems*, ORSUM '22. ACM.

[3] Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M. J.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; and Zhang, Q. 2018. JAX: composable transformations of Python+NumPy programs.

[4] Bruch, S.; Han, S.; Bendersky, M.; and Najork, M. 2020. A Stochastic Treatment of Learning to Rank Scoring Functions. In *Proc. of the 13th International Conference on Web Search and Data Mining*, WSDM '20, 61–69. ACM.

[5] Chapelle, O.; and Li, L. 2011. An Empirical Evaluation of Thompson Sampling. In *Proc. of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, 2249–2257. ISBN 9781618395993.

[6] Chilinski, P.; and Silva, R. 2020. Neural Likelihoods via Cumulative Distribution Functions. In *Proc. of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, UAI '20, 420–429. PMLR.

[7] Choi, H.; Mela, C. F.; Balseiro, S. R.; and Leary, A. 2020. Online display advertising markets: A literature review and future directions. *Information Systems Research*, 31(2): 556–575.

[8] Clevert, D.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). ICLR '16. arXiv.

[9] Daniels, H.; and Velikova, M. 2010. Monotone and Partially Monotone Neural Networks. *Trans. Neur. Netw.*, 21(6): 906–917.

[10] Dudík, M.; Langford, J.; and Li, L. 2011. Doubly Robust Policy Evaluation and Learning. In *Proc. of the 28th International Conference on International Conference on Machine Learning*, ICML'11, 1097–1104.

[11] Duetting, P.; Feng, Z.; Narasimhan, H.; Parkes, D.; and Ravindranath, S. S. 2019. Optimal Auctions through Deep Learning. In *Proc. of the 36th International Conference on Machine Learning*, volume 97 of *ICML '19*, 1706–1715. PMLR.

[12] Edelman, B.; and Ostrovsky, M. 2007. Strategic Bidder Behavior in Sponsored Search Auctions. *Decis. Support Syst.*, 43(1): 192–198.

[13] Golrezaei, N.; Lin, M.; Mirrokni, V.; and Nazerzadeh, H. 2021. Boosted Second Price Auctions: Revenue Optimization for Heterogeneous Bidders. In *Proc. of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, 447–457. ACM.

[14] Grover, A.; Wang, E.; Zweig, A.; and Ermon, S. 2019. Stochastic Optimization of Sorting Networks via Continuous Relaxations. In *Proc. of the 7th International Conference on Learning Representations*, ICLR '19. arXiv.

[15] Hammond, P. J. 1979. Straightforward Individual Incentive Compatibility in Large Economies. *The Review of Economic Studies*, 46(2): 263–282.

[16] He, X.; Pan, O., J.and Jin; Xu, T.; Liu, B.; Xu, T.; Shi, Y.; Atallah, A.; Herbrich, R.; Bowers, S.; and Candela, J. Q. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proc. of the 8th International Workshop on Data Mining for Online Advertising*, ADKDD'14, 1–9. ACM.

[17] Jagerman, R.; Wang, X.; Zhuang, H.; Qin, Z.; Bendersky, M.; and Najork, M. 2022. Rax: Composable Learning-to-Rank Using JAX. In *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 3051–3060. ACM.

[18] Jeunen, O.; and Goethals, B. 2021. Pessimistic Reward Models for Off-Policy Learning in Recommendation.

In *Proc. of the 15th ACM Conference on Recommender Systems*, RecSys '21, 63–74. ACM.

[19] Jeunen, O.; and Goethals, B. 2022. Pessimistic Decision-Making for Recommender Systems. *ACM Trans. Recomm. Syst.*

[20] Jeunen, O.; Murphy, S.; and Allison, B. 2022. Learning to Bid with AuctionGym. In *Proc. of the AdKDD Workshop at the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, AdKDD '22. ACM.

[21] Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. ICLR '15. arXiv.

[22] Krishna, V. 2003. *Auction Theory*. Elsevier Inc.

[23] Lahaie, S.; and Pennock, D. M. 2007. Revenue Analysis of a Family of Ranking Rules for Keyword Auctions. In *Proc. of the 8th ACM Conference on Electronic Commerce*, EC '07, 50–56. ACM.

[24] Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A.; Choi, S.; and Teh, Y. W. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proc. of the 36th International Conference on Machine Learning*, volume 97 of *ICML '19*, 3744–3753. PMLR.

[25] Liu, X.; Yu, C.; Zhang, Z.; Zheng, Z.; Rong, Y.; Lv, H.; Huo, D.; Wang, Y.; Chen, D.; Xu, J.; Wu, F.; Chen, G.; and Zhu, X. 2021. Neural Auction: End-to-End Learning of Auction Mechanisms for E-Commerce Advertising. In *Proc. of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, 3354–3364. ACM.

[26] Luce, R. D. 1959. *Individual choice behavior: A theoretical analysis*. Wiley.

[27] Ma, N.; Ispir, M.; Li, Y.; Yang, Y.; Chen, Z.; Cheng, D. Z.; Nie, L.; and Barman, K. 2022. An Online Multi-Task Learning Framework for Google Feed Ads Auction Models. In *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 3477–3485. ACM.

[28] Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* Sampling. In *Proc. of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, 3086–3094. MIT Press.

[29] McMahan, H. B.; Holt, G.; Sculley, D.; Young, M.; Ebner, D.; Grady, J.; Nie, L.; Phillips, T.; Davydov, E.; Golovin, D.; et al. 2013. Ad click prediction: a view from the trenches. In *Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1222–1230. ACM.

[30] Meek, C.; Chickering, D. M.; and Wilson, D. B. 2005. Stochastic and contingent-payment auctions. In *1st Workshop on Sponsored Search Auctions*. Citeseer.

[31] Myerson, R. B. 1981. Optimal Auction Design. *Mathematics of Operations Research*, 6(1): 58–73.

[32] Oosterhuis, H. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. In *Proc. of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, 1023–1032. ACM.

[33] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8026–8037.

[34] Plackett, R. L. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2): 193–202.

[35] Prillo, S.; and Eisenschlos, J. M. 2020. SoftSort: A Continuous Relaxation for the Argsort Operator. In *Proc. of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

[36] Rindt, D.; Hu, R.; Steinsaltz, D.; and Sejdinovic, D. 2022. Survival regression with proper scoring rules and monotonic neural networks. In *Proc. of The 25th International Conference on Artificial Intelligence and Statistics*, AISTATS '22, 1190–1205. PMLR.

[37] Sandholm, T. 2003. Automated Mechanism Design: A New Application Area for Search Algorithms. In Rossi, F., ed., *Principles and Practice of Constraint Programming*, CP '03, 19–36. Springer.

[38] Vickrey, W. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1): 8–37.

[39] Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; and Chi, E. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-Scale Learning to Rank Systems. In *Proc. of the Web Conference*, WWW '21, 1785–1797. ACM.

[40] Wilkens, C. A.; Cavallo, R.; and Niazadeh, R. 2017. GSP: The Cinderella of Mechanism Design. In *Proc. of the 26th International Conference on World Wide Web*, WWW '17, 25–32.

[41] Yellott, J. I. 1977. The relationship between Luce's Choice Axiom, Thurstone's Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2): 109–144.

[42] Zhang, Z.; Liu, X.; Zheng, Z.; Zhang, C.; Xu, M.; Pan, J.; Yu, C.; Wu, F.; Xu, J.; and Gai, K. 2021. Optimizing Multiple Performance Metrics with Deep GSP Auctions for E-Commerce Advertising. In *Proc. of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, 993–1001. ACM.