CONSTRAINED TRIPLE-HEAD PRICE OPTIMIZATION (C3PO) NETWORK FOR DISCRETE-CHOICE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a Constrained Triple-Head Price Optimization (C3PO) network for real-time bi-level decision making in discrete-choice environments. In these settings, a seller selects product or service offerings to achieve business objectives, while heterogeneous users make personalized acceptance or rejection decisions based on their preferences, ultimately determining seller outcomes. Existing approaches that fine-tune pre-trained tabular foundation models on synthetic data are limited by their imitation-learning paradigm and inability to enforce common pricing rules, including cross-product and global constraints. C3PO addresses these limitations through a novel transformer-based architecture that jointly integrates price imitation learning, multi-task revenue prediction, and in-context elasticity learning to generate robust pricing recommendations across domains without retraining. A lightweight differentiable constraint layer enables generated prices to satisfy business constraints while optimizing the revenue objective. During inference, frontier-model prompting elicits elasticity priors for previously unseen products informed by behavioral economics literature, improving out-of-distribution pricing effectiveness. We demonstrate strong out-of-distribution performance on simulated, synthetic, and real-world datasets. C3PO is trained on data generated from multiple classical discrete-choice models, comprising simulated customer segments and counterfactual action-outcome pairs, and is evaluated on randomly generated choice environments without observing the underlying preference structure. Across simulated benchmarks where the calibrated MNL model is misspecified relative to the ground-truth choice model, C3PO improves expected revenue by 1.4 to 7.9 percentage points over a calibrated MNL baseline. The constraint layer reduces average absolute constraint violations by more than 50%. We further deploy the model in real-world applications, including healthcare tender pricing and airline ancillary pricing, where C3PO improves directional pricing effectiveness, measured by price-increase and price-decrease recall, by 18–34 percentage points depending on the baselines considered, including calibrated MNL, classical optimization, and off-the-shelf tabular foundation-models.

1 INTRODUCTION

Discrete choice models (DCMs) are widely used in forecasting, assortment planning, pricing, marketing and other quantitative decision making processes across various industries such as transportation, restaurant and food service, retail, and entertainment when customers select among a finite set of products or services designed, provided and priced by the firms. DCMs naturally capture a bi-level decision process well suited to competitive settings between a firm's offering and customers' selection. From buyers' perspective, individual consumers evaluate the available options and select (or decline) items based on their own utility, driven by factors such as willingness to pay, perceived quality, and personal preferences. Meanwhile, the seller strategically sets the adjustable attributes of the offer set (for example, prices) to influence customer choices and optimize key performance metrics such as expected revenue or profit margin. The decision-making literature has traditionally emphasized domain-specific models tailored to particular applications, rather than pursuing a universal model capable of capturing the essential relationship and common structure underlying diverse tasks and general decision strategies. This reliance on specialized models arises in part from the fact that

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.

real-world decisions are shaped by complex human behavior and tacit knowledge that are difficult to capture with a single predictor or even an ensemble of traditional modeling approaches.

This paper aims to overcome these limitations by introducing a method for generating optimal actions that adapt to product types, customer heterogeneity and diverse constraints. Standard approaches optimize prices after fitting a parametric choice model (e.g., MNL) to each new domain. When the parametric family is misspecified, the choice model and resulting prices degrade, and the entire prediction-optimization pipeline must be rebuilt for every new product or market. An alternative is to fine-tune pre-trained tabular foundation models such as TabPFN (Hollmann et al., 2022) on pricing data, which mitigates the re-estimation cost but relies solely on imitation learning of unconstrained optimal actions and cannot effectively enforce the cross-product ordering, bound, and budget constraints that real pricing problems require. We design a new transformer-based architecture trained from scratch on simulated data spanning diverse discrete-choice families, that addresses both limitations. A **multitask component** captures the objective function (revenue–price curve) and its variation, shaping an **imitation learning module** that learns a representation of optimal prices, while an **in-context learning module** guides predictions by modeling price elasticity rather than directly tracking the gradients of the objective. A lightweight differentiable constraint layer minimizes business-rule violations without post-hoc repair.

Whereas the focus of this paper is on pricing as an exemplar of discrete-choice based decision-making, the proposed approach is applicable beyond pricing to many other problem areas such as optimal product design, assortment optimization, transportation planning, and marketing, among others.

We first validate our approach in controlled settings using synthetic datasets generated from well-established ground-truth discrete choice models. Next, we apply the fine-tuned model to approximately solve a complex stochastic optimization problem arising in multiple real-world settings such as competitive tender pricing for medical device supply, and airline ancillary pricing, illustrating how the model can enhance pricing strategies for major industry participants. Finally, we benchmark the fine-tuned model recommendations against those obtained from traditional approaches and highlight the relative strengths and limitations of each approach.

2 Related Work

Discrete Choice Models. McFadden (1974a;b) introduced the random utility framework and conditional logit model, thus laying the theoretical foundation of discrete choice modeling for which D. McFadden was awarded the Nobel Prize in Economics in 2000. This fundamental theory is enriched by (Anderson et al., 1992) where the discrete choice theory of product differentiation was developed, and it provides a unified framework for modeling horizontal and vertical product differentiation that addresses the independence of irrelevant alternatives (IIA) problem. (Gallego & Wang, 2014) derived the optimality conditions for maximizing expected margin for multinomial and nested logit models, which can be leveraged to calculate the optimal prices using a fixed point search. The work by (Zhang et al., 2018) extended this result to pricing under generalized extreme value models (GEV). Data driven and nonparametric approach for decision-making in discrete choice models are pursued in (Farias et al., 2009; 2013), and extended to Markov setting in (Blanchet et al., 2016).

Discrete choice models underpin a broad range of practical decision-making applications, including air travel demand forecasting and revenue management, (Garrow, 2016), personalized bundle pricing, (Xue et al., 2016), joint pricing and customer segmentation under business constraints via model-free column generation, (Subramanian et al., 2022), and for multi-product revenue optimization across product lines, (Gallego & Berbeglia, 2024). Related efforts in the vehicle routing domain include the work of (Mo et al., 2023) where the authors train a pair-wise attention pointer network to infer real-world driver routing behavior. Among other things, this network captures tacit driving hacks that are not readily modeled in classical vehicle routing formulations.

Code Generation Models for Policy Learning. Recent models such as CodeT5 (Wang et al., 2021) and StarCoder (Li et al., 2023) have been cited as having the ability to translate complex business constraints and objectives into executable policy functions, exploiting their inherent understanding of conditional logic and structured decision rules essential for discrete choice policy generation.

Tabular Foundation Models. TabPFN (Hollmann et al., 2022; 2025), building on the PFN framework (Müller et al., 2022), demonstrates that transformers pre-trained on synthetic data can achieve state-of-the-art (SOTA) performance on tabular classification tasks.

3 Problem Formulation

Consider a marketplace where a product/service provider (seller) offers multiple (K) options to buyers who can select one of these choices to maximize their own utilities. Let $\mathbf{x}, \mathbf{y}, \mathbf{z}_i$ denote the product feature vector, customers' choice vector, and i-th customer attribute vector, respectively. Let $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_I)$ be the attribute vector for all customers and $\mathbf{p}$ denote the pricing decision variable. Details on these variables and assumptions can be found in Appendix A. Given choices, a customer's utilities are denoted by $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,K})$, with corresponding choice probabilities $\mathbf{q} = (q_1, \ldots, q_K)$ based on the distribution of customer preferences. For example, the Multinomial Logit (MNL) model assumes that the random components of utility are independently and identically distributed (i.i.d.) according to a Gumbel distribution. Moreover, the default choice 0 can be added, which is often interpreted as the baseline option, for example, it could refer to the selecting economy seats on a commercial flight relative to business, premium, or first class; or the no-purchase option, typically not having observed data for price or product features.

Given these variables and their data instances, indexed by n, a multi-product pricing problem can be formulated as maximizing the expected revenue, subject to feasible prices in an action space Π:

$$\max_{\mathbf{p}_n \in \Pi} \ \mathbf{p}_n \cdot \mathbf{q}^\top(\mathbf{p}_n, \mathbf{x}_n, \mathbf{Z}_n). \tag{1}$$

Here, the probability of discrete choices $\mathbf{q}(\mathbf{u}_i)$ depends on the utility $\mathbf{u}_i(\mathbf{p}_n, \mathbf{x}_n, \mathbf{Z}_n)$, a function of the decision variables, non-decision features, and customer attributes. Consequently, the seller's optimal decisions depend on the non-decision features, for example through a pricing policy $\mathbf{p}_n^*(\mathbf{x}_n, \mathbf{Z}_n)$.

One viable approach to combining discrete choice models with machine learning techniques is to approximate $\mathbf{u}(\mathbf{x}, \mathbf{z})$ with a neural network, and then obtain choice probabilities through a softmax operation, as shown in Aouad & Désir (2026). Specifically, in Aouad & Désir (2026), $\mathbf{u}(\mathbf{x}, \mathbf{z})$ is treated as a random field and approximated via a system of neural networks accordingly. Their key technical result shows that, for any discrete choice model, there exists a neural network approximation whose induced choice probabilities are arbitrarily close (in total variation distance) to the original ones for any instance. The goal of our price optimization network is more ambitious: the model should not depend on any particular underlying choice model. Instead, a desirable theoretical guarantee is that, for any discrete choice model, the probabilities it generates can be approximated by those produced by the model, or equivalently, that the induced utility functions are themselves universally approximable; see (Yuan, 2023). Our model is then utilized to solve bi-level optimization equation 1 with the utilities, hence the probabilities, minimizes the loss function of the component for prediction. While we utilize the pricing domain for clarity of exposition, the proposed framework is domain-agnostic and readily extensible to diverse decision-making tasks and action spaces.

4 C3PO Model for Price optimization under Discrete Choice

4.1 Model Architecture

Figure 1 illustrates the C3PO architecture, which maps customer attributes and historical price–revenue contexts to constrained price predictions via a hierarchical transformer with in-context learning. The model receives two primary inputs: a vector of customer features and a set of context samples (p_i, r_i), where p_i is a historical price vector and r_i the corresponding revenue. At inference time, an elasticity signal—estimated from the context data and optionally anchored by a prior from economic literature—is incorporated as a control input.

C3PO combines three learning paradigms: (i) **Imitation learning**—the price head imitates an oracle pricing policy via regression on labeled optimal prices; (ii) **Multi-task revenue modeling**—an auxiliary revenue head learns the price-to-revenue mapping $f(p)$, is frozen at inference, and serves as a differentiable reward signal directing the price head toward higher-revenue regions during training; and (iii) **In-context learning with elasticity priors**—labeled samples prepended to the batch enable few-shot prediction, while a product-type-specific elasticity range anchors prices. The

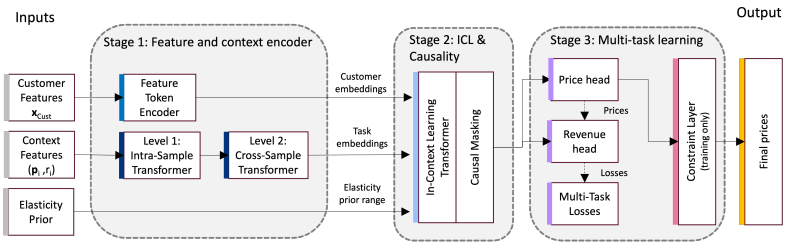


Figure 1: High-level C3PO model architecture.

model estimates per-customer elasticity from the context by comparing average revenue at below-median versus above-median price points; an external prior from the economics literature regularizes this estimate. Learning elasticity rather than the gradient $f'(p)$ improves cross-domain generalization and allows decision-makers to specify priors for new products via an LLM prompt.

Stage 1: Feature and Context Encoding. Customer features are mapped to embeddings by a feature token encoder. Each context sample (p_i, r_i) is processed by an *intra-sample* transformer that captures within-pair structure; the resulting embeddings are then aggregated by a *cross-sample* transformer attending across all context samples, yielding a task embedding of the pricing environment.

Stage 2: In-Context Learning with Causal Masking. Customer and task embeddings are concatenated and passed to an ICL transformer with a causal mask, enabling few-shot conditioning on example rows without retraining. At inference, the elasticity estimate introduces an analytical anchor price derived from the Lerner optimality condition (Anderson et al. (1992)) and a directional bias. The shared ICL representation feeds the price head.

Stage 3: Prediction Heads and Constraint Projection. The price head outputs candidate prices via imitation learning; the revenue head (active only during training) learns the price-to-revenue map, enabling gradient-based revenue maximization through the price head with the revenue head frozen. A differentiable constraint layer then projects raw prices onto a feasible set defined by box bounds, inter-product price ordering, and a global average-price constraint. The layer operates in three sequential passes: (1) clamping each price to its box bounds, (2) enforcing pairwise ordering by propagating adjustments along the product hierarchy, and (3) redistributing any residual average-price violation proportionally across products, keeping the per-sample cost linear in K (see Appendix B.4). Constraint satisfaction is additionally encouraged via soft penalties during training. More rigorous alternatives such as OptNet (Amos & Kolter, 2017) can substitute for this lightweight heuristic.

Training Objective. C3PO is trained with a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{\text{price}} + \lambda_{\text{rev}}\mathcal{L}_{\text{rev}} + \lambda_{\text{revmax}}(t)\,\mathcal{L}_{\text{revmax}} + \lambda_{\text{constr}}\mathcal{L}_{\text{constr}} + \mathcal{L}_{\text{reg}} \tag{2}$$

$\mathcal{L}_{\text{price}}$ is a Huber loss matching predicted prices to oracle optima. $\mathcal{L}_{\text{rev}}$ calibrates the revenue head at observed context prices and ranks optimal prices above random alternatives via a margin loss. $\mathcal{L}_{\text{revmax}}$ maximizes predicted revenue by backpropagating through the price head with the revenue head frozen; it is ramped in after a warmup once revenue predictions are sufficiently calibrated. $\mathcal{L}_{\text{constr}}$ penalizes constraint violations on pre-projection prices. $\mathcal{L}_{\text{reg}}$ comprises elasticity-prior alignment and analytical-anchor alignment losses (Appendix C.9).

4.2 Transfer Learning in C3PO

The in-context learning (ICL) capability of C3PO learns to make optimal decisions from sequences of the form $(\mathbf{x}', \mathbf{q}(\mathbf{x}'), \mathbf{p}^*(\mathbf{x}'))$, for any $\mathbf{x}'$ in the testing dataset $\mathbf{X}'$ of the size $(M + 2K) \times N'$. For this type of problem, it is common to normalize the decision variables using an appropriate baseline. For details, see Appendix B.1.1. There are two possible scenarios, depending on whether the choice probabilities $\mathbf{q}'$ are retrained during transfer learning.

4.2.1 The First Scenario

In the first scenario, no discrete-choice observations are available in the testing dataset. In this case, transfer learning relies on statistically contextual similarity between the pre-trained data and the

testing data. The transformer then leverages the pre-trained optimal policy $\mathbf{p}^\circ(\mathbf{x}_n)$ learned from the training dataset $\mathbf{X}_{M\times N}$, represented by learned parameters in the form of matrix weights, to approximate the optimal policy $\mathbf{p}^*(\mathbf{x}')$ for any out-of-distribution $\mathbf{x}'$ that might be best quantified by a set of different learned parameters. The quality of this approximation depends on the degree of similarity between the distributions of the testing and training datasets, and is an intensively studied subject in *domain adaptation*, see e.g. Redko et al. (2019), a topic of transfer learning with different source and target distributions but the same task. Here, we adopt some fundamental ideas and methodologies in domain adaptation, and derive some approximation error estimations specific to the problem of optimal decision making for discrete choices models.

Suppose that the pre-trained data $\mathbf{x}_n$ follows a distribution over the feature space, which is usually termed *source* distribution and denoted as π_S; the testing data $\mathbf{x}'$ follows a different distribution, is termed the *target* distribution and denoted as π_T. Each statistical hypothesis is represented by a high dimensional matrix weight θ, which means that the model $\mathcal{T}_\theta$ takes an input feature vector $\mathbf{x}$ to the output spaces (discrete for choice probabilities and continuous for optimal pricing). Specifically, here the model produces the optimal price as the solutions to the problem defined by equation 1. The set of all statistical hypotheses is denoted as Θ. The error of learning is described by $\mathbb{E}_\pi[\ell(\mathbf{x}, \mathbf{y}; \theta)]$ where ℓ denotes the loss function characterizing the accuracy of the hypothesis with parameter θ.

Assumption 4.1. *π_T is absolute continuous with respect to π_S, and there exists a positive number $\delta > 0$, such that $L_{S,T}$, the likelihood ratio of π_T with respect to π_S, is upper bounded by $1 + \delta$.*

Remark 4.2. *Depending on the topological structure of the feature space, the likelihood can be represented as either the ratio of probabilities(discrete case), or Radon-Nikodym derivative(continuous case), for details, see, e.g., Rohde et al. (2014).*

Definition 4.3. *For a subset $A \subseteq \Theta$, the A-divergence between two probability measures π, γ, denoted as $D_A(\pi, \gamma)$, is defined as $\sup_{a\in A} |\mathbb{E}_\pi[\ell(\mathbf{x}, \mathbf{y}; a)] - \mathbb{E}_\gamma[\ell(\mathbf{x}, \mathbf{y}; a)]|$.*

The following theorem characterizes the error relationship (proof in Appendix B.3).

Theorem 4.4. *Suppose that the source and target distributions satisfy Assumption 4.1, for U, a neighborhood of optimal matrix weight θ^* from pre-training, we have,*

$$e_T(\theta^*) \leq \min\{e_S(\theta^*) + \gamma_U + D_U(\pi_S, \pi_T), (1 + \delta)e_S(\theta^*)\} \tag{3}$$

with $e_T(\theta) = \mathbb{E}_{\pi_T}[\ell(\mathbf{x}, \mathbf{y}; \theta)]$, $e_S(\theta) := \mathbb{E}_{\pi_S}[\ell(\mathbf{x}, \mathbf{y}; \theta)]$, and $\gamma_U := \min_{\theta\in U}[e_T(\theta) + e_S(\theta)]$.

Remark 4.5. *While Theorem 4.4 falls into the family of typical results in domain adaptation, its novelty lies in its connection with normalized price range, and for the two-level LLM architecture, as well as the incorporation of the likelihood which is widely used in the pricing literature and introduces a multiplicative factor instead of the additive ones in the literature in domain adaptation.*

The definition of A-divergence is a variation of the $H\Delta H$ divergence that is introduced in Ben-David et al. (2010) and has been widely studied in domain adaption research. This definition allows us to specify a focus subset of hypotheses for the optimal pricing problem equation 1. As we examine the potential hypotheses which are reflected as learned matrix weights in the network model, the analysis can be restricted to a neighborhood U of the optimal weighted trained with the source data. Potentially, such restriction can lead to better estimations of the generalization error by utilizing well-documented "flat minimum" phenomenon for stochastic gradient descent, see Keskar et al. (2017); Jiang et al. (2020); Singh et al. (2025); Dziugaite & Roy (2018). More precisely, a proper selected neighborhood may produce a small D_U-divergence between the source and target distributions, thus smaller generalization error. Furthermore, the bound of the likelihood ratio characterizes relations between sensitivities of features (including demand) to price in the form of their joint probability, thus can be viewed as a more general quantification of the demand price elasticity as discussed in Appendix B.1.2 in the form of conditional probability. In this sense, Theorem 4.4 produces a very specific domain adaptation theory for optimal decision making for discrete choice models.

4.2.2 The Second Scenario

In the second scenario, the testing dataset contains either (i) choice probabilities already estimated by the training pipeline or (ii) a small number of discrete-choice observations.

Choice Probabilities: The choice probability vector $\mathbf{q}'$ is estimated using $(\mathbf{x}', \mathbf{p}')$ (or normalized prices $\tilde{\mathbf{p}}'$) where $\mathbf{p}' = \mathbf{p}(\mathbf{x}')$ or $\tilde{\mathbf{p}}' = \tilde{\mathbf{p}}(\mathbf{x}')$ is generated using either a predicted price $\mathbf{p}^\dagger$ or a

near-optimal price $\mathbf{p}^\circ$ obtained from the first scenario. Therefore, the error of estimation can be quantified by Theorem 4.4 with the transformer-based prediction function maps the feature inputs to probability vector defined over the possible choices.

Optimal Pricing: In our model, both the contextual information and the likelihood of discrete choices are incorporated. Thus, the optimal policy $\mathbf{p}^*(\mathbf{x}', \mathbf{q}'(\mathbf{x}'))$ is inferred via transfer learning from the pretrained policy $\mathbf{p}^\bullet(\mathbf{x}_n,\, \mathbf{q}(\mathbf{x}_n))$. The incorporation of either choice probabilities in the training or a small number of observations enriches probabilistic characterizations of the target domain, and thus naturally narrows the difference between the source and target domains, hence,

Lemma 4.6. *Suppose $\tilde{\pi}_T$ is obtained from π_T by conditioning on additional observed labels on a subset of the feature space, then $D_A(\pi_S, \tilde{\pi}_T) \leq D_A(\pi_S, \pi_T)$.*

This thus will lead to the following corollary concluding an improved bound on error estimation.

Corollary 4.7. *The error bound in the target domain can be improved to $e_T(\theta^*) \leq \min\{e_S(\theta^*) + \gamma_U + D_U(\pi_S, \tilde{\pi}_T), (1+\delta)e_S(\theta^*)\}$*

Hence, our model enables transfer learning through the features and choice probabilities obtained above where $\mathbf{q}'(\mathbf{x}')$ is computed using predicted or near-optimal prices. Consequently, a network model obviates the need for computationally expensive optimization of an NP-hard problem for every transaction, significantly reducing the decision-processing time. After observing the actual prices $\mathbf{p}'$ and the corresponding outcomes $\mathbf{y}'$, a more accurate probability estimation can be obtained to improve optimization when necessary. Another advantage of our network model is that it can be trained or fine-tuned using one of many well-known approaches for generating synthetic tabular data (see Appendix C.5 for more details). We demonstrate that it is sufficient to train the C3PO network using simulated data generated from classical discrete choice models in order to generalize well across diverse domains, heterogeneous customer preferences, and product types.

5 Experiments

5.1 Simulated Data Generation For Model Training

Training data are simulated using several classical DCMs: MNL (16%), Nested Logit (16%), Mixed MNL (27%), Iso-elastic (27%), and Linear (14%). Optimal price labels for MNL and NL datasets are computed via fixed-point search using the optimality condition in (Gallego & Wang, 2014), and nonlinear optimization for other models. The simulations yield 13,000 datasets with over 1.5 million labeled examples: 12,000 for training, the remainder for ablation studies, reserving 1,600 examples (13 datasets) for in-context learning (ICL). Segment-specific choice parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are randomly generated for each dataset. Complete details on generating price labels and what-if outcome data are in Appendix C.6 and C.10. The resultant C3PO model has about 337,000 trainable parameters.

5.2 Experimental Setup

C3PO is evaluated on real and simulated data from six domains: B2B tender-pricing, airline ancillary pricing, Amazon DVD sales (Farias et al., 2013), Hotel room selection (Newman et al., 2014), Swiss Metro ridership (Bierlaire et al., 2001), and Yogurt (Jain et al., 1994). Dataset details are in Table 10 (Appendix C.10); elasticity priors are drawn from the literature (Babić, 2005; Mankiw, 1998) and domain knowledge (Appendix C.7). Noting that the available ground truth differs across datasets, we organize evaluation into three groups (Table 1). Details on setup and baselines are in Appendix C.1.

Remarks: C3PO was trained on datasets with $K \in [2, 6]$ simulated choices and is evaluated without retraining on the DVD dataset with $K{=}15$ choices (Table 10 has additional dataset details). The Swiss Metro dataset is generated using the TasteNet-MNL model (Han et al., 2022) that embeds a neural network within an MNL structure to capture taste heterogeneity.

5.3 Revenue Analysis

Experimental Setup. For Groups A and B the ground-truth choice model enables exact expected-revenue computation for any price vector. We evaluate four price sources per instance: (i) **Optimal/Reference**—revenue-maximizing prices (Group A) or sampled offered prices (Group B); (ii) **C3PO**—our model's recommendations; (iii) **MNL**—a calibrated single-model baseline (below);

and (iv) **C3PO-Dir**—a deployment proxy adjusting each reference price by $\pm 5\%$ in C3PO's indicated direction, modeling the intended operating regime of small, frequent, tactical adjustments. All prices are capped at ± 1 historical standard deviation around the mean.

Baseline. Given that the ground truth models contain an MNL kernel, we generate a strong baseline by pricing using an MNL model (Gallego & Wang (2014)) calibrated on the 1,600 ICL examples of each domain by first minimizing revenue MSE and then refining via maximum likelihood estimation. This standard predict-and-optimize pipeline is in-distribution when the ground truth is itself MNL.

Results. Table 1 reports the ground-truth normalized revenue, regret or lift, and win rates. All datasets in groups A and B are evaluated on $10{,}000$ instances, except Yogurt which has 800 held-out examples derived from the original dataset (Jain et al., 1994). We summarize the key findings:

			GT Norm. rev. [95% CI]			
Grp	**Dataset**	**Method**	**Mean**	**Med.**	**Regret / Lift**	**GT model**
A — regret from optimal (lower is better)						
	DVD	C3PO	0.993^{***} [.991, .996]	0.964	-0.7%	Mixed MNL
		C3PO-Dir	1.000^{***} [.999, 1.00]	0.999	-0.0%	
		MNL	0.914 [.912, .916]	0.980	-8.6%	
	SWISS	C3PO	0.826^{***} [.822, .829]	0.833	-17.4%	TasteNet-MNL
		C3PO-Dir	0.962^{***} [.961, .962]	0.954	-3.8%	
		MNL	0.812 [.810, .815]	0.814	-18.8%	
	YOGURT	C3PO	0.824^{***} [.817, .831]	0.848	-17.6%	Mixed MNL
		C3PO-Dir	0.978^{***} [.978, .978]	0.978	-2.2%	
		MNL	0.786 [.779, .794]	0.773	-21.4%	
B — lift over offered prices (higher is better)						
	HOTEL	C3PO	1.070^{***} [1.069, 1.071]	1.067	$+7.0\%$	MNL
		C3PO-Dir	1.031^{***} [1.030, 1.031]	1.031	$+3.1\%$	
		MNL	1.122 [1.121, 1.124]	1.154	$+12.2\%$	
C — live data (directional only; see Tables 2, 3)						
	B2B, AIRLINE	—	*ground-truth revenue unavailable*			unknown

Table 1: Evaluation summary with 95% bootstrap CIs (B=10,000). Group A: regret from oracle optimum; Group B: lift over offered prices; Group C: directional metrics only. $^{***}p < 0.001$ vs. calibrated MNL (Wilcoxon signed-rank).

- **C3PO outperforms MNL on all misspecified domains.** On DVD (mixed-MNL ground truth), C3PO's -0.7% regret dominates MNL's -8.6% for an improvement of 7.9 pp. On Yogurt (mixed-MNL), C3PO (-17.6%) outperforms MNL (-21.4%) by 3.8 pp, and on Swiss Metro (TasteNet-MNL), C3PO (-17.4%) outperforms MNL (-18.8%) by 1.4 pp. All differences are significant at $p < 0.001$ (Wilcoxon signed-rank). C3PO's gains are largest when the degree of MNL misspecification is greatest.
- **C3PO-Dir confirms deployment potential.** The C3PO-Dir directional proxy adjusts reference prices by a small fixed step in C3PO's recommended direction. C3PO-Dir is ground-truth aware when the reference is optimized (Group A) and is the dominant Group A performer. This result suggests that when C3PO's direction is correct, it can apply small repeated adjustments to a good-quality reference price and converge to near-optimal revenue.
- **MNL wins in-distribution.** On Hotel (MNL ground truth, offered-price reference), MNL achieves the largest lift ($+12.2\%$) while C3PO still delivers a significant $+7.0\%$ lift over offered prices, remaining competitive across choice-model families.

Statistical testing. All pairwise revenue comparisons use the Wilcoxon signed-rank test on per-instance differences; 95% bootstrap CIs (B=10,000) are reported in Table 1. All C3PO-vs-MNL and C3PO-Dir-vs-MNL comparisons are significant at $p < 0.001$ across all four datasets.

Remark 5.1. *The domain adaptation bound in Theorem 4.4 provides qualitative insight into C3PO's empirical behavior: the largest revenue gains over MNL occur on DVD and Yogurt, where the*

mixed-MNL ground truth is most distant from the MNL family, corresponding to larger γ_U for the MNL baseline but not for C3PO's nonparametric hypothesis class.

5.4 Directional Performance (PDR/PIR)

Price Increase Recall (PIR) and Price Decrease Recall (PDR), as defined in Ye et al. (2018), measure how effectively a model identifies opportunities to raise prices on winning bids (PIR) and to lower prices on losing bids (PDR). A *balanced* pair—high $\min(\text{PDR}, \text{PIR})$—signals a well-calibrated recommender rather than one systematically biased toward raising or lowering prices.

Live datasets in group C: The B2B ICL dataset consists of 1600 competitive tender bids sampled from four years of historical data having an average win rate of about 34%, and is used to price bids for the next year. The reference prices are generated by the bidder's rule-based system. The airline ICL dataset consists of 1600 examples of upgrade purchases for a global carrier, and is used to price 10,000 upgrade choices across diverse test markets. The historical win rate is less than 14% and the reference prices are optimized values generated by a sophisticated revenue management system.

5.4.1 B2B: in-depth comparison.

The B2B setting is the canonical buy vs no-buy problem and admits the richest set of baselines. We compare C3PO against (off-the-shelf) TabPFN, a fine-tuned TabPFN (TabPFN-FT), and the constrained stochastic-optimization policy of Subramanian et al. (2022), which uses full model knowledge (Table 2). The TabPFN-FT fine-tuning details are in Appendix C.4. C3PO attains the highest $\min(\text{PDR}, \text{PIR}) = 0.611$, exceeding the stochastic optimizer (0.430), TabPFN-FT (0.589), and TabPFN (0.393). TabPFN and the stochastic policy both exhibit a high PDR but a weak PIR while C3PO instead produces the most *balanced* PDR-PIR, which is our directional KPI.

5.4.2 Directional performance across all domains.

Table 3 compares the PDR, PIR, and $\min(\text{PDR}, \text{PIR})$ for C3PO versus MNL for the live datasets. The PDR-PIR comparison for all datasets is available in Table 6 of the appendix. A consolidated table of PDR-PIR and revenue gain for groups A and B is presented in appendix C.3.

C3PO outperforms MNL on directional accuracy across all domains, maintaining balanced PDR/PIR ($\min > 0.41$) on every dataset. The live datasets represent real choice processes involving unobserved competitive heterogeneity (B2B) and nested substitution with extreme class imbalance (airline), neither of which MNL can represent regardless of sample size.

B2B Pricing Model	PDR	PIR
C3PO	**0.611**	**0.673**
Baseline TabPFN	0.767	0.393
TabPFN-FT	0.723	0.589
Stochastic Opt. Policy	0.646	0.430
MNL	0.443	0.299

Table 2: B2B directional pricing metrics: $\min(\text{PDR}, \text{PIR})$ is maximized by C3PO. Stochastic-optimization and TabPFN both under-price.

	PDR ↑		PIR ↑		$\min(\text{PDR}, \text{PIR})$ ↑	
Dataset	C3PO	MNL	C3PO	MNL	C3PO	MNL
B2B	**0.611**	0.443	**0.673**	0.299	**0.611**	0.299
AIRLINE	**0.507**	0.135	**0.476**	0.318	**0.476**	0.135
Δ (avg)	+0.270		+0.266		+0.327	

Table 3: Directional pricing metrics on live datasets. C3PO improves $\min(\text{PDR}, \text{PIR})$ over MNL by $+31$ pp on B2B and $+34$ pp on Airline.

The calibrated MNL baseline systematically overprices by underestimating price elasticity from limited data, yielding low PDR across all domains (see Appendix C.2 for analysis). On the other hand, C3PO's non-parametric approach maintains balanced directional accuracy by learning flexible, context-conditioned price mappings pre-trained across thousands of diverse choice environments.

5.5 Ablation Studies

The C3PO architecture is designed to overcome the limitation of TabPFN in three areas, which we want to analyze through ablation studies: 1. Move beyond imitation-learning of prices to incorporate multi-task learning of revenue prediction. 2. The impact of eliminating or reducing ICL examples. 3. Ability to learn the elasticity for a new domain via ICL examples and the use of an elasticity prior that is available in the literature. The ablation studies analyze the impact of each of these components of C3PO. Simulated test data from the various discrete choice models used to train C3PO are included as an additional dataset in these studies. Table 4 reports the results averaged across all the datasets.

Scenario	MAE	PDR	PIR
C3PO	.38±.11	.49±.12	.54±.13
ICL-OFF	.66±.13	.20±.16	.81±.17
CONSTR-ON	.31±.15	.43±.30	.57±.30
IMIT-ONLY	.35±.06	.59±.20	.41±.16
PRIOR-OFF	.41±.08	.44±.15	.58±.19
SIMPLE-ICL	.40±.08	.43±.14	.59±.18

Table 4: Avg. pricing metrics (mean±std).

	Abs. Mean		% Mean	
Constraint	Unconstr.	Constr.	Unconstr.	Constr.
Box	.19±.39	.07±.16	9.7±19.7	3.3±8.2
Ordering	.15±.17	.06±.09	16.9±20.2	9.8±15.5
Avg. Price	.07±.10	.03±.04	6.0±8.1	3.3±5.5

Table 5: Avg. constraint violations (mean±std).

In the first study we turn off the attention network that models the revenue-price curve and replace it with a statistical summary. This reduces C3PO to an 'imitation-only' learning model akin to TabPFN. The performance deteriorates when it attempts to generalize to some of the new domains as can be seen in the skewed PDR-PIR numbers, and the min(PDR, PIR) drops from C3PO's score of 0.49 to 0.41. The detailed results of this simplified network are shown in Table 11 in Appendix C.8.

The second study evaluates the impact of ICL on pricing. In one setting, ICL is disabled ("ICL-OFF"). In another, the ICL examples are reduced to 100 ("SIMPLE-ICL"). With no ICL, the model fails to generalize effectively, with recommended prices clustering near the upper bound and the KPI dropping to 0.2, and just 100 ICL examples suffice to improve the KPI to 0.43. In contrast, the MNL calibration quality is expected to degrade when the calibration data reduces from 1600 to 100 samples as it relies on point estimates of β. Tables 12 and 13 in Appendix C.8) present detailed results.

The third study introduces an explicit elasticity prior to account for the consumers' product-specific willingness to pay based on necessity and the availability of substitutes. The results are shown in Table 14 in Appendix C.8. The prior is useful when a relatively narrow elasticity range can be derived using domain knowledge. In the absence of a product-specific prior, model accuracy declines and the KPI drops to 0.44 as the system falls back to the default elasticity range of (-3.0, -1.0).

5.6 Constraint Satisfaction

We report on the performance of the heuristic constraint satisfaction layer described in Section 4.1 Table 5 tracks the constraint satisfaction percentage and average percentage violation across the datasets, for three common constraint types that show up in practical discrete choice settings.

Table 15 and 16 in Appendix C.10 depict the feasibility performance before, and after, activating the constraint layer, respectively, for each of the datasets, We observe a substantial reduction in infeasibility for most domains. Table 4 summarizes the directional KPI dropping from 0.49 to 0.43 after activating the constraint layer, while reducing the absolute constraint violations by more than 50%. A post-processing step can be applied to project the final predictions into the feasible space.

6 Conclusions

We introduce C3PO, a constrained triple-head transformer for bi-level price optimization in discrete-choice settings. By unifying imitation learning, multi-task revenue modeling, and in-context elasticity learning with a differentiable constraint layer, C3PO produces domain-adaptive pricing policies from a single model trained entirely on simulated data spanning diverse choice-model families.

On four benchmarks with known ground-truth models, C3PO improves expected revenue over calibrated MNL by 1.4–7.9 pp when MNL is misspecified, with all gains significant at $p < 0.001$. The directional proxy C3PO-Dir, which applies small repeated adjustments in C3PO's recommended direction, achieves near-optimal revenue on all Group A datasets, confirming deployment viability. C3PO improves balanced directional accuracy by 31–34 pp over calibrated MNL.

The architecture scales naturally: training on additional choice-model families and customer-segment scenarios requires only generating more simulated data, with no structural changes. Inference-time elasticity priors, obtainable via a single LLM prompt, further extend the model to unseen product categories without retraining. Limitations include degraded performance in extreme elasticity regimes and approximate constraint satisfaction. Future work will address these via richer training distributions and differentiable optimization layers.

AI USE STATEMENT

In this work, we used generative AI tools for editing and polishing the text. We have not used generative AI tools to generate mathematical proofs. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All missing proofs are deferred to the appendix. Simulated data for training and all the ICL and evaluation datasets in Group A and B (excluding private client data in Group C) are available at `https://anonymous.4open.science/r/C3PO-024F/`. Details on generating the price labels are deferred to the appendix.

ETHICS STATEMENT

This paper aims to advance machine learning for optimal decision-making under uncertainty in discrete-choice environments by leveraging foundation models for decision making using a novel constrained triple-head price optimizer architecture (C3PO). When applied appropriately, C3PO can help sellers and service providers implement high-quality, data-driven recommendations that respond more rapidly to uncertainty and improve key performance indicators. Customers in price-sensitive segments are also expected to benefit, as service providers seeking to increase win rates often offer more competitive discounts. The C3PO framework is particularly advantageous for smaller firms that may lack the resources or specialized expertise required to build, maintain, and deploy sophisticated stochastic optimization models.

However, C3PO can execute actions rapidly and this comes with risks if not employed judiciously, e.g., in low-elasticity settings without practical safeguards. While C3PO is trained on counterfactual actions that can satisfy constraints, including regulatory and fairness goals, it is possible that such constraints may be relaxed to boost a company's financial performance. For example, algorithmic decisions that include demographic attributes within contextual data can turn into hyper-gouging and surveillance-based pricing practices that can be disadvantageous to customer groups Federal Trade Commission (2025).

Real-world implementations of C3PO should ensure transparency and uphold all regulatory and fairness requirements embedded in the optimization frameworks used for training. Preserving these requirements helps maintain the feasibility and integrity of the actions the model is trained to replicate. We also recommend using an independent post-processing layer to audit model outputs and verify compliance. Ultimately, practitioners must apply careful due diligence and consider the broader economic and societal implications before deploying these models at scale.

REFERENCES

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017.

Simon P Anderson, Andre De Palma, and Jacques-Francois Thisse. *Discrete choice theory of product differentiation*. MIT press, 1992.

Ali Aouad and Antoine Désir. Representing random utility choice models with neural networks. *Management Science*, 2026. doi: 10.1287/mnsc.2023.02189. URL `https://doi.org/10.1287/mnsc.2023.02189`.

Ante Babić. Microeconomics by Robert S. Pindyck and Daniel L. Rubinfeld. *Financial theory and practice*, 29(4):385–386, 2005.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Michel Bierlaire, Kay W Axhausen, and Georg Abay. The acceptance of modal innovation: The case of swissmetro. *Swiss Transport Research Conference*, 2001. URL https://transp-or.epfl.ch/pythonbiogeme/examples_swissmetro.html.

Jose Blanchet, Guillermo Gallego, and Vineet Goyal. A markov chain approximation to choice modeling. *Operations Research*, 64(4):886–905, 2016.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

Brllrb. Uber and lyft dataset boston, ma. https://www.kaggle.com/datasets/brllrb/uber-and-lyft-dataset-boston-ma, 2019.

Gintare Karolina Dziugaite and Daniel Roy. Entropy-sgd optimizes the prior of a pac-bayes bound: Generalization properties of entropy-sgd and data-dependent priors. In *International Conference on Machine Learning*, pp. 1377–1386. PMLR, 2018.

Vivek Farias, Srikanth Jagabathula, and Devavrat Shah. A data-driven approach to modeling choice. *Advances in Neural Information Processing Systems*, 22, 2009.

Vivek F Farias, Srikanth Jagabathula, and Devavrat Shah. A nonparametric approach to modeling choice with limited data. *Management science*, 59(2):305–322, 2013.

Federal Trade Commission. Ftc surveillance pricing study indicates wide range of personal data used to set individualized consumer prices. https://www.ftc.gov/news-events/news/press-releases/2025/01/ftc-surveillance-pricing-study-indicates-wide-range-personal-data-used-set-individ 2025.

Guillermo Gallego and Gerardo Berbeglia. Bounds and heuristics for multiproduct pricing. *Management Science*, 70(6):4132–4144, 2024.

Guillermo Gallego and Ruxian Wang. Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research*, 62(2): 450–461, 2014.

Laurie A Garrow. *Discrete choice modelling and air travel demand: theory and applications*. Routledge, 2016.

Yafei Han, Francisco Camara Pereira, Moshe Ben-Akiva, and Christopher Zegras. A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability. *Transportation Research Part B: Methodological*, 163:166–186, 2022.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *NeurIPS, First Table Representation Workshop*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Instacart. The instacart online grocery shopping dataset 2017. https://www.instacart.com/datasets/grocery-shopping-2017, 2017.

Dipak C Jain, Naufel J Vilcassim, and Pradeep K Chintagunta. A random-coefficients logit brand-choice model applied to panel data. *Journal of Business & Economic Statistics*, 12(3):317–328, 1994.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

P. Langley. Crafting papers on machine learning. In Pat Langley (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *Transactions on Machine Learning Research*, 2023.

Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen–tabular data generation with tabpfn. In *NeurIPS, Second Table Representation Learning Workshop*, 2023.

N Gregory Mankiw. *Principles of microeconomics*, volume 1. Elsevier, 1998.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, 1974a.

Daniel McFadden. The measurement of urban travel demand. *Journal of public economics*, 3(4): 303–328, 1974b.

Daniel McFadden and Kenneth Train. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470, 2000.

Baichuan Mo, Qingyi Wang, Xiaotong Guo, Matthias Winkenbach, and Jinhua Zhao. Predicting drivers' route trajectories in last-mile delivery using a pair-wise attention-based pointer neural network. *Transportation Research Part E: Logistics and Transportation Review*, 175:103168, 2023.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Jeffrey P Newman, Mark E Ferguson, Laurie A Garrow, and Timothy L Jacobs. Estimation of choice-based models using sales data from a single firm. *Manufacturing & Service Operations Management*, 16(2):184–197, 2014.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018/.

M. Parkin, M. Powell, and K. Matthews. *Economics*. Addison-Wesley, 2008. ISBN 9780132041225. URL https://books.google.com/books?id=Kp6Ls7j3tVIC.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani. *Advances in domain adaptation theory*. Elsevier, 2019.

Charles A Rohde et al. *Introductory statistical inference with the likelihood function*. Springer, 2014.

Paul A. Samuelson and William D. Nordhaus. *Economics*. McGraw-Hill, 19th edition, 2009.

Sidak Pal Singh, Hossein Mobahi, Atish Agarwala, and Yann Dauphin. Avoiding spurious sharpness minimization broadens applicability of sam. In *International Conference on Machine Learning*, pp. 55702–55719. PMLR, 2025.

Shivaram Subramanian, Wei Sun, Youssef Drissi, and Markus Ettl. Constrained prescriptive trees via column generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36(4), pp. 4602–4610, 2022.

Kenneth Train et al. *Discrete choice methods with simulation*, volume 2. Cambridge university press Cambridge, 2003.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 8696–8708, 2021.

Zhengliang Xue, Zizhuo Wang, and Markus Ettl. Pricing personalized bundles: A new approach and an empirical study. *Manufacturing & Service Operations Management*, 18(1):51–68, 2016.

Peng Ye, Julian Qian, Jieying Chen, Chen-hung Wu, Yitong Zhou, Spencer De Mars, Frank Yang, and Li Zhang. Customized regression model for airbnb dynamic pricing. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 932–940, 2018.

Yang Yuan. On the power of foundation models. In *International conference on machine learning*, pp. 40519–40530. PMLR, 2023.

Heng Zhang, Paat Rusmevichientong, and Huseyin Topaloglu. Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research*, 66(6):1559–1570, 2018.

Yihua Zhang, Prashant Khanduri, Ioannis C Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Process. Mag.*, 2024.

A Notations and Assumptions for the Optimization Formulation

- There are K products offered by the seller, indexed by $k = 1, \ldots, K$. Accordingly, a customer's choice is characterized by $\mathbf{y} = (y_1, \ldots, y_K) \in \mathbb{R}^K$, identifying the purchased product y, satisfying $\sum_{k=1}^{K} y_k = 1$. In the case that default choice 0 is added, we will have $\sum_{k=0}^{K} y_k = 1$.
- Seller's decisions are also characterized by a vector $\mathbf{p} = (p_1, p_2, \ldots, p_K) \in \mathbb{R}^K$, a representative example is a price vector, associated with the revenue vector $\mathbf{r} = (r_1, r_2, \ldots, r_K)$, which will be the one considered in this paper.
- There are M observable, non-decision product features, measured either as real-valued variables or elements of finite lattices (e.g., binary indicators). These numeric and categorical attributes are denoted by $\mathbf{x} = (x_1, \ldots, x_M) \in \mathbb{R}^M$ for any purchased record.
- There are L observable customer attributes, denoted by $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,L}) \in \mathbb{R}^L$ for each customer (or customer segment) $i = 1, 2, \ldots, I$, and $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_I)$.
- There are N historical transaction data indexed by $n = 1, \ldots, N$. Each transaction is represented by $(\mathbf{p}_n, \mathbf{x}_n, \mathbf{y}_n, \mathbf{Z}_n)$.

B Theoretical Development

B.1 Methodologies for Decision Model under Discrete Choice

B.1.1 Normalization of the Decision Variables

Normalization is important both for the optimization procedure and for obtaining meaningful insights into the resulting optimal decisions, and is conceptually distinct from the feature standardization used in prediction tasks. For example, pricing models often require negative demand elasticity, which places structure on how normalized prices should behave. The choice of baseline varies across pricing applications and may include cost, list price, competitive price, etc. For example, in an airline dataset, we normalize the upgraded-service prices using the base economy fare $p_{0,n}$ as the benchmark, yielding normalized prices of the form $\tilde{p}_{k,n} = \frac{p_{k,n}}{p_{0,n}} - 1$, for any k in the upper classes.

B.1.2 Price Elasticity

Some basic facts and results that can enhance our understanding of the basic mechanism between the optimal pricing decisions and the key quantity of price elasticity, thus improve the design and execution of C3PO, are collected below.

Some basic facts in optimization can help us to understand the basic mechanism between the optimal pricing decisions and the key quantity of price elasticity.

Definition B.1. *The demand elasticity matrix $E \in \mathbb{R}^{K \times K}$ is defined as $E_{jk} = \frac{p_k}{q_j} \frac{\partial q_j}{\partial p_k}$ for all $j, k = 1, 2, \ldots K$.*

Remark B.2. *The definition of demand elasticity matrix, which can also be written as $E_{jk} = \frac{\partial q_j}{q_j} / \frac{\partial p_k}{p_k} = \frac{\partial \ln q_j}{\partial \ln p_k}$ and should be viewed as scaled gradient of the demand with respect to price, and it follows the basic logic of being the ratio between changes in demand and changes in price that is well known in the economics literature, see e.g. Parkin et al. (2008). The diagonal entries E_{kk} are* own-price elasticities *and they are nonpositive naturally. The off-diagonal entries E_{jk} $(j \neq k)$ are* cross-price elasticities*, and they are typically positive for substitutes. Comparing to the slope of demand curve, price elasticity is scale-invariant and dimensionless, thus is preferred for analysis of diverse product and with counterfactual actions such as ours.*

In the one product case, i.e. $K = 1$, the price elasticity of demand takes the form of $E(p) := pq'(p)/q(p)$. Without any constraints, then it is easy to see from the optimal condition, we have $\epsilon(p^*) = -\frac{p^*}{p^* - c}$, in particular, $E(p^*) = -1$ when $c = 0$. In the multi-product case, several basic relationships between the optimal price and the cost structure can be derived easily from the optimality condition.

Proposition B.3 (multi-product optimality condition)**.** *At an interior optimum* $\mathbf{p}^*$*, the first-order conditions in vector form are*

$$\mathsf{Diag}(\mathbf{p})\mathbf{q}/R + E(\mathbf{p}^*)^\top \mathbf{s}(\mathbf{p}^*) = \mathbf{0}\,, \tag{4}$$

where $\mathbf{s} = [s_1, s_2, \ldots, s_K]^\top$*,* $s_k = (p_k - c_k)\, q_k/R, k = 1, 2, \ldots, K$ *is the* profit-weighted revenue share *of product* k*.*

Proof. Setting $\partial R/\partial p_k = 0$ for each $k = 1, 2, \ldots, K$:

$$q_k + \sum_{j=1}^{K} (p_j - c_j)\frac{\partial q_j}{\partial p_k} = 0\,. \tag{5}$$

Thus,

$$q_k + \sum_{j=1}^{K} \frac{(p_j - c_j)\, q_j}{p_k} \cdot \underbrace{\frac{p_k}{q_j}\frac{\partial q_j}{\partial p_k}}_{E_{jk}} = 0$$

Equivalently,

$$q_k + \sum_{j} \frac{(p_j - c_j)\, q_j}{R} \cdot \frac{R}{p_k} \cdot E_{jk} = 0\,.$$

This implies

$$\frac{p_k q_k}{R} + \sum_{j} \frac{(p_j - c_j)\, q_j}{R} \cdot E_{jk} = 0\,.$$

Equation (4) thus follows. □

B.1.3 MNL Closed Form and the IIA Structure

In MNL with utilities $V_k = \alpha_k - \beta p_k$, i.e., price sensitivity is common across all products and outside option $V_0 = 0$, we have,

$$q_k(\mathbf{p}) = \frac{e^{\alpha_k - \beta p_k}}{1 + \sum_j e^{\alpha_j - \beta p_j}}\,.$$

A direct consequence is that,

Proposition B.4 (MNL elasticity structure)**.**

$$E_{kk} = -\beta\, p_k\, (1 - q_k)\,, \tag{6}$$

$$E_{jk} = +\beta\, p_k\, q_k \quad (j \neq k)\,. \tag{7}$$

Proof. $\partial q_k/\partial p_k = -\beta\, q_k(1 - q_k)$ and $\partial q_j/\partial p_k = \beta\, q_j\, q_k$ for $j \neq k$ (standard MNL derivatives). Multiplying by p_k/q_k and p_k/q_j respectively. □

Furthermore,

Theorem B.5 (MNL multi-product optimal price)**.** *The revenue-maximizing price for product* k *satisfies*

$$p_k^* = \frac{1}{\beta(1 - q_k^*)}\,. \tag{8}$$

Proof. The FOC equation 5 with $c_k = 0$ gives $q_k + \sum_j p_j\, \partial q_j/\partial p_k = 0$. Using MNL derivatives:

$$q_k + p_k(-\beta\, q_k(1 - q_k)) + \sum_{j \neq k} p_j\, \beta\, q_j\, q_k = 0\,.$$

Dividing by $q_k > 0$: $1 - \beta\, p_k(1 - q_k) + \beta \sum_{j \neq k} p_j\, q_j = 0$. $1 - \beta\, p_k(1 - q_k) + \beta \sum_{j \neq k} p_j\, q_j = 0$ immediately implies $1 - \beta p_k + \beta \sum_{j=1}^{K} p_j\, q_j = 0$, equivalently, $p_k^* = (1 + \beta R)/\beta$. The fixed-point $p_k^* = (1 + \beta R^*)/\beta$ combined with $R^* = \sum_k p_k^* q_k^*$ yields $p_k^* = 1/[\beta(1 - q_k^*)]$ after algebraic rearrangement. □

B.2 Bi-level Optimization

As we elaborated in the introduction, the optimization problem equation 1 belongs to the extensively studied domain of bi-level optimization. A general formulation of bi-level optimization takes the following form,

$$\min_{\mathbf{x}\in\mathbb{R}^{d_x}} \ell(\mathbf{x}) := f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})), \quad \text{s.t. } \mathbf{y}^*(\mathbf{x}) \in \arg\min_{\mathbf{y}\in\mathbb{R}^{d_y}} g(\mathbf{x}, \mathbf{y}), \tag{9}$$

with given functions $f(\mathbf{x}, \mathbf{y})$ and $g(\mathbf{x}, \mathbf{y})$. Under strong (usually convexity) conditions on the functions, algorithms have been developed and shown to be efficient. For problems with complex inner level structures such as those in equation 1, algorithmic development is still quite limited. For theory, algorithms and other detailed discussion on bi-level optimization, see e.g. Borkar (1997); Zhang et al. (2024).

B.3 Proof of Theorem 4.4

Proof of Theorem 4.4. In a parameterized learning regime, hypotheses are represented by the optimized parameter. In the case of transformers, they are the weights after the training, which are represented as θ^*. Consider U, a neighborhood of θ^*, for any $\theta' \in U$, define $e_T(\theta, \theta') := \mathbb{E}_{\pi_T}[|\ell(\mathbf{x}, \mathbf{y}; \theta) - \ell(\mathbf{x}, \mathbf{y}; \theta')|]$ and $e_S(\theta, \theta') := \mathbb{E}_{\pi_S}[|\ell(\mathbf{x}, \mathbf{y}; \theta) - \ell(\mathbf{x}, \mathbf{y}; \theta')|]$. We have

$$\begin{aligned} e_T(\theta^*) \leq & e_T(\theta') + e_T(\theta^*, \theta') \\ \leq & e_T(\theta') + e_S(\theta^*, \theta') + |e_T(\theta^*, \theta') - e_S(\theta^*, \theta')| \\ \leq & e_S(\theta^*) + e_T(\theta') + e_S(\theta') + D_U(\pi_S, \pi_T), \end{aligned}$$

where the inequalities follow from the definition of $e_T(\theta, \theta')$ and the triangle inequality. Therefore, we can conclude that

$$e_T(\theta^*) \leq e_S(\theta^*) + \gamma_U + D_U(\pi_S, \pi_T).$$

On the other hand, note that, $e_T(\theta^*) = \mathbb{E}_{(x,y)\sim D_T}[\ell(\mathbf{x}, \mathbf{y}; \theta^*)]$, and

$$\begin{aligned} \mathbb{E}_{(x,y)\sim D_T}[\ell(\mathbf{x}, \mathbf{y}; \theta^*)] &= \mathbb{E}_{(x,y)\sim D_S}[\ell(\mathbf{x}, \mathbf{y}; \theta^*) L_{S,T}] \leq (1+\delta)\mathbb{E}_{(x,y)\sim D_S}[\ell(\mathbf{x}, \mathbf{y}; \theta^*)] \\ &= (1+\delta) e_S(\theta^*), \end{aligned}$$

with the inequality follows from Assumption 4.1. Therefore, we have the desired result. □

B.4 Effects of Inter-Product Ordering Constraints

In discrete choice-based price optimization, a firm offers K substitutable products to heterogeneous customers who select the utility-maximizing option (including a no-purchase alternative). In many settings—airline fare classes, hotel room tiers, subscription plans—business logic requires a price hierarchy among products:

$$p_{\pi(k)} \geq p_{\pi(k-1)} + \Delta_k, \quad k = 2, \ldots, K, \tag{10}$$

where π is a known ordering (e.g., economy $\leq$ business $\leq$ first class) and $\Delta_k \geq 0$ are minimum inter-tier price gaps. When prices are output directly by a neural network, these constraints must be enforced structurally.

Proposition B.6 (Ordering constraints by construction)**.** *Let $z_1, \ldots, z_K \in \mathbb{R}$ be unconstrained network outputs. Define prices recursively:*

$$p_{\pi(1)} = z_1, \tag{11}$$

$$p_{\pi(k)} = p_{\pi(k-1)} + \Delta_k + \mathrm{softplus}(z_k), \quad k = 2, \ldots, K, \tag{12}$$

where $\mathrm{softplus}(x) = \log(1 + e^x)$*. Then $p_{\pi(k)} > p_{\pi(k-1)} + \Delta_k$ for all k and all $z \in \mathbb{R}^K$.*

Remark B.7 (Reasoning and implications)**.** *From positivity of softplus we have, $p_{\pi(k)} - p_{\pi(k-1)} = \Delta_k + \mathrm{softplus}(z_k) > \Delta_k$. This construction is a standard technique for enforcing positivity in neural network outputs applied here to the multi-product pricing setting.*

Remark B.8 (Limitations)**.** *While the softplus reparameterization cleanly enforces ordering, it might not be sufficient in the presence of additional constraints. For example, price range constraints in the forms of $\ell_k \leq p_k \leq u_k, k = 1, \dots K$ will become difficult to enforce given the recursive expansion equation 12. Furthermore, average price constraints $\frac{1}{K}\sum_k p_k \geq \bar{p}$ become non-separable and non-linear in z. Moreover, when target price gaps are near-tight ($p_{\pi(k)} - p_{\pi(k-1)} \approx \Delta_k$), the required inverse $z_k = \log(e^{p_{\pi(k)} - p_{\pi(k-1)} - \Delta_k} - 1)$ has the tendency to go to $-\infty$, thus creating numerical instability during training. These limitations motivate a dedicated constraint satisfaction layer that operates directly in price space, jointly enforcing bounds, ordering, and aggregate constraints without distorting the network's output representation.*

C Omitted Details for Experiments

C.1 Datasets for Discrete Choice Modeling

Several publicly available datasets have been adapted for discrete choice research. In retail and e-commerce, the Amazon product data (Ni et al., 2019) and Instacart online grocery dataset (Instacart, 2017) provide rich choice signals from millions of user interactions that can be restructured as discrete choice problems. For transportation choices, the Boston Uber/Lyft dataset (Brllrb, 2019) offers fare quotes with contextual features, while the Swissmetro dataset (Bierlaire et al., 2001) remains a benchmark for revealed and stated preference modeling in transportation research. In addition to the simulated datasets, our proposed approach aims to leverage these additional resources to fine-tune the network model in order to generalize better across different choice contexts.

Details for the three groups of data listed in Table 1:

- *Group A (simulated, oracle-optimal reference)* comprises Amazon DVD (Farias et al., 2013), Swiss Metro (Bierlaire et al., 2001), and Yogurt (Jain et al., 1994). For each evaluation instance the reference price is the revenue-maximizing price computed under the known ground-truth model, so recommended prices can never exceed it and *regret from optimal* is well defined.
- *Group B (simulated, offered-price reference)* comprises the Bodea Hotel dataset (Newman et al., 2014), which follows the Newman two-step estimation protocol: the reference prices are simulated offered prices sampled from the historical Bodea price ranges rather than optimal prices. Consequently methods can exceed the reference, and the meaningful quantity is *revenue lift over the offered prices*.
- *Group C (live, no ground truth)* comprises the two datasets drawn from actual client engagements—B2B tender pricing and airline ancillary pricing (AIRLINE). Here the true choice model is unknown, so no ground-truth revenue can be computed; we can only compare recommended price movements against the offered legacy prices and their observed win/loss outcomes. For these we report the model-free directional metrics PDR and PIR.

C.2 Detailed PDR-PIR Statistics

		PDR ↑		**PIR** ↑		$\min(\text{PDR}, \text{PIR})$ ↑	
Group	**Dataset**	**C3PO**	**MNL**	**C3PO**	**MNL**	**C3PO**	**MNL**
Live data	B2B	**0.611**	0.443	**0.673**	0.299	**0.611**	0.299
	AIRLINE	**0.507**	0.135	**0.476**	0.318	**0.476**	0.135
Simulated data	DVD	**0.430**	0.022	0.530	**0.955**	**0.430**	0.022
	SWISS	**0.640**	0.331	0.410	**0.691**	**0.410**	0.331
	YOGURT	**0.500**	0.060	0.543	**0.956**	**0.500**	0.060
	HOTEL	0.410	—	0.530	**1.000**	0.410	—
Avg (excl. Hotel)		**0.543**	0.198	0.522	**0.644**	**0.490**	0.169

Table 6: Directional pricing metrics across all domains. C3PO consistently achieves higher PDR (price-decrease recall among losses) while MNL tends toward high PIR by uniformly raising prices. The balanced metric $\min(\text{PDR}, \text{PIR})$ favors C3PO in every dataset where MNL is available, with an average improvement of $+32.1$ pp.

Table 6 reports PDR (price-decrease recall among losses) and PIR (price-increase recall among wins) for C3PO and calibrated MNL across all six domains.

MNL scores high on PIR but very low on PDR, while C3PO maintains more balanced directional accuracy. This asymmetry reveals that MNL systematically recommends price *increases* across nearly all products and instances. The balanced metric $\min(\text{PDR}, \text{PIR})$ consequently favors C3PO in every domain where MNL is available, by an average of $+32.1$ pp.

MNL overpricing: The MNL baseline is calibrated from the same in-context learning (ICL) data made available to C3PO. With limited data and no access to the true heterogeneous preference

distribution, a calibrated MNL model *underestimates* price sensitivity. When $|\beta|$ is biased downward, the resultant markup $1/|\beta|$ is inflated, shifting the entire optimal price vector upward. MNL's underestimation of elasticity from limited or aggregate data is well documented in the discrete choice literature. Train et al. (2003) notes that the standard MNL model imposes homogeneous preferences across individuals. When consumers differ in their price sensitivity, the estimated coefficient reflects an average response across heterogeneous segments, potentially muting predicted demand responses and elasticities. Similarly, McFadden & Train (2000) show that accounting for random taste heterogeneity through mixed logit models can materially affect inference on preferences and willingness-to-pay relative to homogeneous specifications.

C.3 Consolidated Revenue and PDR-PIR Table

Revenue gain over reference. Table 7 reports for each dataset in groups A and B, C3PO's directional balance metric $\min(\text{PDR}, \text{PIR})$ alongside the percentage regret (Group A) or lift (Group B) relative to the reference price.

				Regret / Lift	
Dataset	**PDR**	**PIR**	**min(PDR,PIR)**	**C3PO**	**MNL**
Group A — regret from optimal					
DVD	0.43	0.53	**0.43**	-0.7%	-8.6%
SWISS	0.64	0.41	0.41	-17.4%	-18.8%
YOGURT	0.50	0.54	0.50	-17.6%	-21.4%
Group B — lift over offered					
HOTEL	0.41	0.53	0.41	$+7.0\%$	$+12.2\%$

Table 7: C3PO directional balance and revenue performance. C3PO outperforms MNL on regret in all three Group A datasets, with the largest margin on DVD (7.9 pp) and Yogurt (3.8 pp). Directional balance $\min(\text{PDR}, \text{PIR})$ is above 0.41 on all domains.

C.4 Fine-Tuned Tabular Foundation Model Baseline for B2B Pricing

The TabPFN-FT baseline in Table 2 is obtained by sequentially fine-tuning the pre-trained TabPFN 2.1 model (Hollmann et al., 2022; 2025) on the simulated discrete-choice datasets described in Section 5.1. The model backbone is frozen and only the output head is updated, using MSE loss on the optimal price labels with a learning rate of 10^{-4}. Fine-tuning uses 3,000 of the 12,000 training datasets; loss reduction exhibits diminishing returns beyond approximately 2,000 datasets. Training completes in under one hour on a single NVIDIA A100 GPU.

TabPFN-FT treats pricing as a regression (imitation-learning) task: given contextual features and 1,600 in-context examples, it predicts the optimal price directly. Unlike C3PO, TabPFN-FT does not model the revenue–price curve, ignores pricing constraints, and does not incorporate elasticity priors—the three extensions that distinguish C3PO.

On the B2B tender dataset, TabPFN-FT achieves PDR $= 0.723$ and PIR $= 0.589$, improving substantially over the off-the-shelf TabPFN ($0.767/0.393$) and the calibrated MNL ($0.443/0.299$). However, its directional balance $\min(\text{PDR}, \text{PIR}) = 0.589$ remains below C3PO's 0.611, reflecting the limitations of imitation-only learning: without an explicit revenue objective, TabPFN-FT tends to under-price (high PDR, lower PIR), a bias that C3PO's revenue head and elasticity prior correct. This comparison motivates the triple-head architecture and confirms that fine-tuning alone, while convenient and beneficial, is insufficient for balanced pricing recommendations.

C.5 Synthetic Data Generation (SDG)

In addition to simulated data, C3PO can be trained, fine-tuned, and evaluated using several approaches for generating synthetic tabular data. The TabPFN-based generation follows the approach of (Ma et al., 2023) of using a pretrained transformer to produce class-logit functions $\boldsymbol{\mu}(\mathbf{x})$ whose softmax defines $\Pr(y \mid \mathbf{x})$.

Beyond TabPFN, we also tested Synthetic Data Vault (SDV) methods that include Gaussian Copulas, CTGAN, TAVE, and bootstrap-based generators. While these methods are vital for finetuning C3PO with synthesized choice structures from different domains, they do not come with analytically derivable decision optimality conditions. Therefore, we restricted our focus to simulated data from discrete choice models (e.g., MNL), which offer unlimited samples and known optimality conditions for benchmarking, albeit under idealized modeling assumptions.

C.6 Sample Code to Generate Recommended Price Labels

This is a plain-vanilla python implementation of a known method based on Gallego & Wang (2014) to enable reproducibility of the controlled experiment with NL and MNL models. Similar algorithms can be used to generate good quality labels for other models such as MMNL, Iso-elastic, and Linear formulations.

Listing 1: Python implementation of Multinomial and Nested Logit pricing

```
import numpy as np
import math
from typing import Dict, Any

def calculate_prices_nested_logit(alpha: np.ndarray,
                                  beta: np.ndarray,
                                  nest_assignments: np.ndarray,
                                  lam: float,
                                  tau_nest: Dict[int, float] = None,
                                  tol: float = 1e-10,
                                  maxit: int = 10000) -> Dict[str, Any]:

    # Generating price recommendations under Multinomial and Nested Logit Models (Ba
    # Optimal for MNL and a fast heuristic for NL

    alpha = np.asarray(alpha, dtype=float)
    beta = np.asarray(beta, dtype=float)
    nest_assignments = np.asarray(nest_assignments, dtype=int)
    if np.any(beta >= 0):
        raise ValueError("All beta_i must be negative.")

    N = len(alpha)
    K = len(np.unique(nest_assignments))

    alpha_eff = alpha.copy()
    if tau_nest is not None:
        for m, tau in tau_nest.items():
            idx = np.where(nest_assignments == m)[0]
            alpha_eff[idx] = alpha_eff[idx] * (1.0 + tau)

    def nested_logit_probs_given_prices(p: np.ndarray):
        u = alpha_eff + beta * p
        S = {}
        P_cond = np.zeros(N, dtype=float)
        for m in range(K):
            idx = np.where(nest_assignments == m)[0]
            exp_scaled = np.exp(u[idx] / lam)
            S[m] = exp_scaled.sum()
            P_cond[idx] = exp_scaled / S[m]
        denom = sum(S[m] ** lam for m in range(K)) + 1.0
        P_nest = {m: (S[m] ** lam) / denom for m in range(K)}
        q_outside = 1.0 / denom
        q_inside = np.zeros(N, dtype=float)
        for m in range(K):
```

```
            idx = np.where(nest_assignments == m)[0]
            q_inside[idx] = P_nest[m] * P_cond[idx]
        return q_inside, q_outside

    def expected_rev_given_R(R: float):
        p = R - 1.0 / beta
        q_inside, q_outside = nested_logit_probs_given_prices(p)
        ER = float(np.sum(p * q_inside))
        return ER, p, q_inside, q_outside

    R = float(max(1e-6, np.mean(1.0 / (-beta))))
    damping = 0.5
    for _ in range(maxit):
        ER, p, q, q0 = expected_rev_given_R(R)
        R_next = (1.0 - damping) * R + damping * ER
        if abs(R_next - R) <= tol * max(1.0, abs(R)):
            R = R_next
            break
        R = R_next
    else:
        def F(x):
            v, *_ = expected_rev_given_R(x)
            return v - x
        a, bnd = 0.0, max(10.0, R*4.0 + 10.0)
        fa, fb = F(a), F(bnd)
        tries = 0
        while fa*fb > 0 and tries < 60:
            bnd = bnd*2.0 + 10.0
            fb = F(bnd)
            tries += 1
        if fa*fb > 0:
            raise RuntimeError("Failed to bracket R* in NL case.")
        for _ in range(200):
            mid = 0.5*(a+bnd); fm = F(mid)
            if abs(fm) < 1e-12 or abs(bnd - a) < 1e-12:
                R = mid; break
            if fa*fm < 0:
                bnd, fb = mid, fm
            else:
                a, fa = mid, fm

    ER, p, q, q0 = expected_rev_given_R(R)
    return {"price": p, "revenue": ER, "q_in": q, "q_out": q0, "method": "fixed_poin
```

Table 8: Sample of representative price elasticity of demand values. See Babić (2005) and Mankiw (1998) for details.

Product / Category	**Elasticity**
Water (basic consumption)	-0.1
Electricity (short-run)	-0.1 to -0.3
Gasoline (short-run)	-0.2 to -0.3
Milk	-0.5
Gasoline (long-run)	-0.6 to -0.8
Electricity (long-run)	-0.7
Restaurant meals	-2.3
Air travel (long-run)	-2.0
Luxury goods	-1.5 to -3.0
Automobiles (long-run)	-1.2 to -1.5

C.7 Elasticity Prior from the Literature

Listing 2: Python example of a prompt to obtain an elasticity prior range

```
prompt = (
    f"<|system|>You are an economics professor who teaches from Mankiw's "
    f"Principles of Economics and Pindyck & Rubinfeld's Microeconomics.</s>"
    f"<|user|>Price elasticity of demand (e) is negative for normal goods. "
    f"INELASTIC means |e| < 1 (between -1.0 and 0): gasoline $e\sim -0.3$, "
    f"bread $e\sim -0.2$, milk $e\sim -0.5$. "
    f"ELASTIC means |e| > 1 (more negative than -1.0): restaurant meals "
    f"$e\sim -1.6$, vacations $e\sim -2.5$, new cars $e\sim -1.5$.\n\n"
    f"Is '{product_string}' typically elastic or inelastic based on "
    f"economics textbooks? "
    f"Answer with exactly one word: 'elastic' or 'inelastic'.</s>"
    f"<|assistant|>")
```

C.8 Ablation Studies Tables

This section reports detailed dataset-level experimental results that are summarized in the main section. Across these datasets, we test the model on more than 600,000 simulated price points, and a cumulative total of over 300,000 sample price points drawn from real-world and public datasets. The underlying discrete choice models are diverse and include logit, neural network, and tree-ensemble based choice structures, some of which are not seen during training.

Table 9: C3PO model performance. PDR/PIR shown as Predicted/Optimal.

Data set	MAE	PDR	PIR
B2B	0.2723	0.61/0.51	0.67/0.41
SIMULATED	0.4150	0.49/0.48	0.43/0.61
DVD	0.4742	0.43/0.52	0.53/0.63
HOTEL	0.2603	0.41/0.43	0.53/0.61
AIRLINE	0.3321	0.51/0.51	0.48/0.50
SWISS	0.4632	0.64/0.42	0.41/0.62
YOGURT	0.3491	0.50/0.53	0.54/0.52

C.9 C3PO Training Loss

For each mini-batch of B customers, a random 30% are designated as in-context examples whose optimal prices are revealed to the ICL transformer; the loss is computed on the remaining queries $\mathcal{Q}$. The total loss equation 2 comprises five terms.

Data set	K	Win Rate	Prior Elasticity	Choice Model
SIMULATED	6	0.2351	(-3.00, -0.50)	MULTIPLE
B2B	1	0.3419	(-2.00, -1.00)	UNKNOWN
DVD	15	0.4442	(-1.50, -0.50)	MMNL
HOTEL	6	0.9086	(-2.50, -1.50)	MNL
AIRLINE	2	0.1378	(-2.50, -0.50)	UNKNOWN
SWISS	3	0.7508	(-0.60, -0.20)	TASTENET-MNL
YOGURT	4	0.9639	(-1.50, -0.50)	MMNL

Table 10: Dataset characteristics and metadata.

Table 11: IMITATION-ONLY ablation results. PDR/PIR shown as Scenario/Baseline.

Eval	MAE	PDR	PIR
SIMULATED	0.4211	0.54/0.49	0.35/0.43
B2B	0.3279	0.90/0.61	0.17/0.67
DVD	0.4181	0.45/0.43	0.54/0.53
HOTEL	0.2977	0.25/0.41	0.70/0.53
AIRLINE	0.2383	0.49/0.51	0.49/0.48
SWISS	0.3633	0.68/0.64	0.37/0.41
YOGURT	0.3435	0.50/0.50	0.54/0.54

Price imitation ($\mathcal{L}_{\text{price}}$). A blend ($\alpha$=0.9) of symmetric Huber loss and an asymmetric variant that estimates per-customer elasticity from context by comparing average revenue at below-median vs. above-median price points. The resulting ratio $\rho_i \in [0.5, 2.0]$ up-weights over-pricing errors for elastic customers and under-pricing errors for inelastic ones.

Revenue calibration ($\mathcal{L}_{\text{rev}}$). Huber regression of predicted vs. actual revenue at all P context price points.

Revenue maximization ($\mathcal{L}_{\text{revmax}}$). Predicted query prices are passed through the *frozen* revenue head and the negative 'ground truth' revenue is minimized: $\mathcal{L}_{\text{revmax}} = -\mathbb{E}_{i\in\mathcal{Q}}[\hat{r}(\hat{\mathbf{p}}_i)/\bar{r}_{\text{ctx}}]$. The weight $\lambda_{\text{revmax}}(t)$ is zero during a warmup ($t \leq 200$), ramps linearly to λ_{target}=0.5 over the next 200 iterations.

Constraint penalty ($\mathcal{L}_{\text{constr}}$). Penalizes violations of per-product bounds, inter-product ordering, and a global weighted-average price floor on pre-projection prices.

Auxiliary regularizers ($\mathcal{L}_{\text{reg}}$). Two terms provide economic inductive bias: (i) an *elasticity prior alignment* loss that penalizes the model's inferred elasticity $\hat{\varepsilon}$ for falling outside a user-specified range $[\varepsilon_{\text{lo}}, \varepsilon_{\text{hi}}]$ and discourages price adjustments opposing the elasticity mismatch direction, and (ii) an anchor alignment loss motivated by the Lerner condition (Anderson et al. (1992)) that nudges predictions toward analytically derived optimal-price anchors (from the first-order condition ε^*=-1). Appendix section B.1.2 presents a detailed elasticity analysis.

Default loss weights are: w_{price}=1.0, λ_{rev}=0.25, λ_{revmax}=0$\rightarrow$0.5 (ramped), λ_{constr}=2.0, λ_{prior}=0.75, λ_{anchor}=0.25.

C.10 Training Data Generation and Loss Plots

The B2B experiment (B2B) represents a fundamental discrete choice model. The airline experiment (AIRLINE) relies on anonymized booking data derived from one of the largest global legacy airline carriers. The DVD and HOTEL datasets are derived from the discrete-choice literature. The DVD dataset in particular, tests the model's scalability in terms of generating prices for a much larger set of choices than what it was trained on. The last two datasets test C3PO performance in a low-elasticity setting. For each domain, we provide 1,600 ICL examples with elasticity priors drawn from the literature.

Table 12: ICL-OFF ablation results. PDR/PIR shown as Scenario/Baseline.

Eval	MAE	PDR	PIR
SIMULATED	0.7430	0.15/0.49	0.87/0.43
B2B	0.8009	0.07/0.61	1.00/0.67
DVD	0.8208	0.46/0.43	0.53/0.53
HOTEL	0.4262	0.31/0.41	0.68/0.53
AIRLINE	0.7096	0.02/0.51	0.96/0.48
SWISS	0.5361	0.42/0.64	0.62/0.41
YOGURT	0.6423	0.13/0.50	0.87/0.54

Table 13: SIMPLE-ICL ablation results. PDR/PIR shown as Scenario/Baseline.

Eval	MAE	PDR	PIR
SIMULATED	0.4135	0.51/0.49	0.41/0.43
B2B	0.3363	0.31/0.61	0.88/0.67
DVD	0.4115	0.50/0.43	0.45/0.53
HOTEL	0.3316	0.36/0.41	0.60/0.53
AIRLINE	0.3552	0.49/0.51	0.53/0.48
SWISS	0.4690	0.63/0.64	0.42/0.41
YOGURT	0.3609	0.40/0.50	0.59/0.54

The model is trained sequentially on each dataset using the aforementioned loss function with a learning rate of 10^{-4}, AdamW optimizer, smooth-L1 and ReLU loss functions, $d_{\text{model}} = 32$, a batch size of 128, and 30% of the rows in each batch are used as examples to enable in-context learning. The resultant number of trainable parameters is about 337,000.

The overall training time is less than an hour on a Windows Intel 20-core laptop with a NVIDIA RTX4000 GPU and 64GB RAM.

Customer heterogeneity is simulated by incorporating seven binary customer attributes for each discrete-choice dataset, which yields 2^7 different customer segments per dataset. The choice cardinality varied between $[2, 6]$. The average price elasticity of the simulated product choices range between (-3.0, -1.0), reflecting mostly price-elastic settings. Note the choices with elasticity less than -1.0 are classified as price elastic, whereas elasticity values in the interval $(-1, 0)$ indicate weakly elastic products Samuelson & Nordhaus (2009). Given this training data choice, C3PO is suited for price-elastic purchase settings where the optimal prices typically lie in the interior of the bounds and do not simply drift to the upper bound, which is often an optimal choice in a weakly elastic setting.

For training, we additionally incorporate information about the feasible space of the underlying stochastic optimization problem through a flattened matrix of counterfactual "what-if" outcomes generated from multiple feasible price vectors. These outcomes are produced using the underlying choice models in our controlled experiments. Specifically, for each data row, we generate 50 random price vectors per segment and record the corresponding expected revenues. We generate one such row per segment, yielding a total of 128 rows per dataset. Each row contains the recommended price vectors corresponding to the given discrete-choice parameter settings. As a result, the contextual features and recommendation for each row take the form $(\boldsymbol{z},\ 50 \times (\boldsymbol{p}, r^*),\ \boldsymbol{y})$, with $L + 50 \times (K + 1) + K$ columns.

We generate prices $\mathbf{p}$ for the 'what-if' data by sampling from a normal distribution with mean and standard deviation equal to 1.0, and clip the sampled values to lie within the interval $[0, 2]$. This design restricts prices to a narrow range around the mean, reflecting real-world pricing distributions that typically arise from limited experimentation. Prior to predicting prices in a new domain, we scale the contextual data, the prices and revenues by the maximum value seen in the ICL dataset.

Table 14: PRIOR-OFF ablation results. PDR/PIR shown as Scenario/Baseline.

Eval	MAE	PDR	PIR
SIMULATED	0.4186	0.48/0.49	0.47/0.43
B2B	0.3036	0.35/0.61	0.87/0.67
DVD	0.4553	0.62/0.43	0.32/0.53
HOTEL	0.3465	0.36/0.41	0.61/0.53
AIRLINE	0.3796	0.44/0.51	0.58/0.48
SWISS	0.4685	0.64/0.64	0.41/0.41
YOGURT	0.3516	0.55/0.50	0.51/0.54

Table 15: Constraint violations: Baseline (unconstrained predictions).

Eval	Constraint	Abs Mean $\pm$ Std	Abs Max	% Mean $\pm$ Std
AIRLINE	Bound	0.07 ± 0.02	0.22	$3.3 \pm 1.0\%$
	Order	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
	AvgPrc	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
B2B	Bound	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
	Order	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
	AvgPrc	0.25 ± 0.21	0.82	$17.9 \pm 15.5\%$
DVD	Bound	1.21 ± 0.11	1.70	$60.7 \pm 5.7\%$
	Order	0.35 ± 0.01	1.64	$45.6 \pm 1.1\%$
	AvgPrc	0.08 ± 0.01	0.13	$10.7 \pm 1.5\%$
HOTEL	Bound	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
	Order	0.19 ± 0.01	1.10	$15.4 \pm 0.6\%$
	AvgPrc	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$
SIMULATED	Bound	0.00 ± 0.00	0.16	$0.0 \pm 0.0\%$
	Order	0.23 ± 0.01	1.10	$21.9 \pm 1.2\%$
	AvgPrc	0.21 ± 0.04	0.77	$19.5 \pm 3.7\%$
SWISS	Bound	0.02 ± 0.08	0.45	$1.0 \pm 3.8\%$
	Order	0.44 ± 0.03	1.19	$52.4 \pm 14.5\%$
	AvgPrc	0.00 ± 0.00	0.02	$0.0 \pm 0.1\%$
YOGURT	Bound	0.00 ± 0.03	0.374	$0.19 \pm 1.26\%$
	Order	0.01 ± 0.02	0.43	$1.89 \pm 2.53\%$
	AvgPrc	0.00 ± 0.00	0.00	$0.0 \pm 0.0\%$

We now describe how to generate recommended prices for in-context learning and training. Note that the prediction targets are the optimal actions of the product/service provider that maximize a seller's expected revenue or profit margin, under the assumption that customers respond to the seller's actions through their discrete choice behavior over the offered products. For known choice models, such as the nested logit (NL) and multinomial logit (MNL) models, we compute the recommended prices $\mathbf{p}_i = \mathbf{p}^\circ(\mathbf{x}_n, \mathbf{z}_i)$ for each customer segment i using the procedure based on the optimality conditions described in Gallego & Wang (2014). For robustness, we incorporate additional fallback logic and numerical safeguards. The Python implementation is provided in Appendix C.6.

Table 16: Constraint violations: Constrained predictions.

Eval	Constraint	Abs Mean ± Std	Abs Max	% Mean ± Std
AIRLINE	Bound	0.00 ± 0.00	0.00	0.0 ± 0.0%
	Order	0.08 ± 0.01	0.18	18.6 ± 1.3%
	AvgPrc	0.00 ± 0.00	0.00	0.0 ± 0.0%
B2B	Bound	0.00 ± 0.00	0.00	0.0 ± 0.0%
	Order	0.00 ± 0.00	0.00	0.0 ± 0.0%
	AvgPrc	0.00 ± 0.00	0.00	0.0 ± 0.0%
DVD	Bound	0.50 ± 0.13	1.03	25.0 ± 6.5%
	Order	0.29 ± 0.01	1.56	47.9 ± 1.1%
	AvgPrc	0.13 ± 0.01	0.16	16.8 ± 1.2%
HOTEL	Bound	0.00 ± 0.00	0.00	0.0 ± 0.0%
	Order	0.02 ± 0.00	0.10	2.3 ± 0.4%
	AvgPrc	0.00 ± 0.00	0.01	0.0 ± 0.0%
SIMULATED	Bound	0.03 ± 0.02	0.73	1.4 ± 0.9%
	Order	0.01 ± 0.00	0.16	0.4 ± 0.2%
	AvgPrc	0.06 ± 0.01	0.65	5.4 ± 0.9%
SWISS	Bound	0.00 ± 0.00	0.00	0.0 ± 0.0%
	Order	0.01 ± 0.01	0.12	1.1 ± 2.1%
	AvgPrc	0.02 ± 0.02	0.11	4.3 ± 4.7%
YOGURT	Bound	0.00 ± 0.02	0.36	0.16 ± 1.1%
	Order	0.01 ± 0.02	0.43	1.87 ± 2.6%
	AvgPrc	0.00 ± 0.00	0.00	0.0 ± 0.0%



Figure 2: Training loss as a function of dataset count, including a log-scaled y-axis and a 50-point moving average.

When customer preferences are represented through complex nonlinear interactions or learned via opaque, data-driven classification models for discrete choice, the optimization algorithm described above can only be used as a heuristic procedure. In practical applications, we can adopt the column-generation-based policy optimization approach proposed in Subramanian et al. (2022) to construct near-optimal pricing policies that satisfy a broad range of operational, regulatory, fairness, and business constraints.

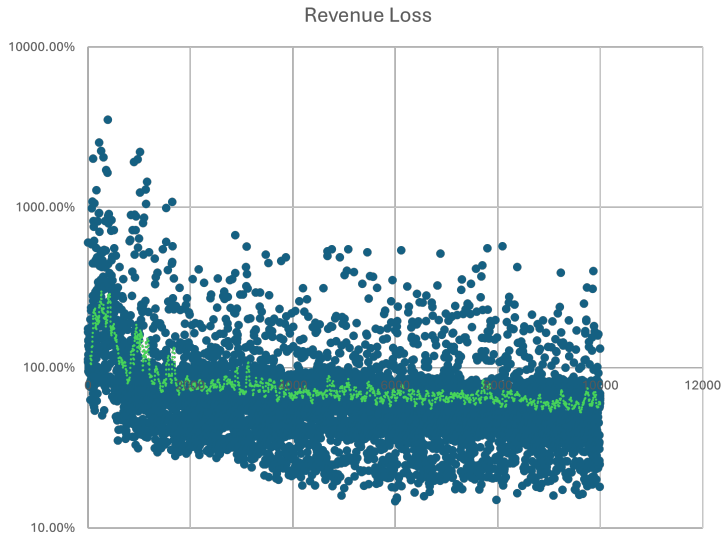


Figure 3: Revenue-loss term as a function of dataset count, including a log-scaled y-axis and a 50-point moving average.

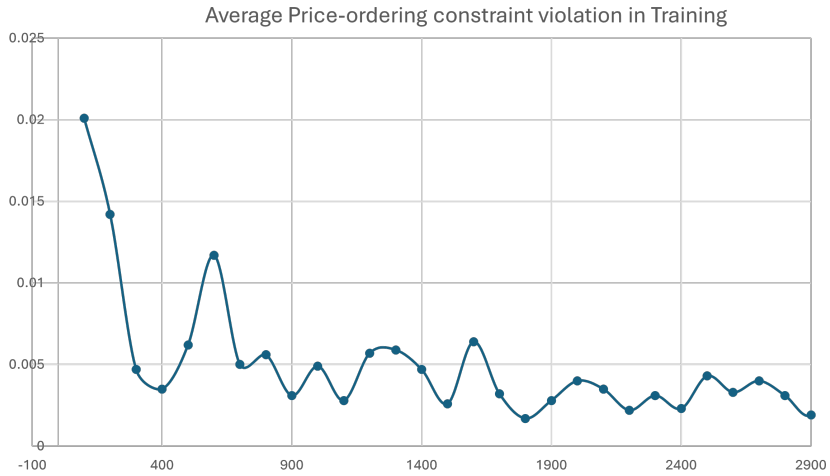


Figure 4: Average price ordering constraint violation as a function of dataset count, including a linear y-axis and a 50-point moving average.

The resulting training dynamics are illustrated in Figure 2, which reports the total training loss values along with a 50-iteration moving average to highlight the mean convergence trajectory. The reduction in the revenue loss component and the average price-ordering constraint violation (when the constraint layer is active) during training is shown in figures 3 and 4.