DIFFUSION-BASED DATA-DRIVEN ASSORTMENT OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Assortment optimization is a fundamental problem in revenue management, typically addressed using parametric choice models such as the multinomial logit (MNL) and its variants. While these models enable tractable formulations, their performance is sensitive to model misspecification and often struggles to capture complex customer behavior. In this paper, we propose a model-agnostic framework for assortment optimization based on guided discrete diffusion. We represent assortments as binary vectors and perform stochastic search via a learned reverse diffusion process, avoiding explicit combinatorial enumeration. To incorporate decision objectives, we introduce a reward-guided mechanism that biases local transitions using pessimistic estimates of expected revenue. This allows the method to effectively balance exploration and exploitation during generation. Empirically, we show that the proposed approach consistently identifies high-quality assortments and remains robust under model misspecification, often recovering near-optimal solutions in high-dimensional settings. Moreover, the generative nature of diffusion enables the production of diverse high-performing assortments, offering flexibility beyond a single deterministic solution. These results highlight the potential of generative modeling as a scalable and robust paradigm for combinatorial optimization in data-driven decision-making.

1 INTRODUCTION

Assortment Optimization (AO) is a core problem in operations research, revenue management, and online platforms (McFadden, 1974; Rooderkerk et al., 2013; Talluri & Van Ryzin, 2004; Rusmevichientong et al., 2010; Strauss et al., 2018). In this setting, the decision-maker selects a subset of products, known as *assortments*, to offer to customers in order to maximize expected revenue or utility under customer choice behavior. AO arises in a wide range of applications, including retail product display, online recommendation systems, advertisement placement, and service bundling (Kök et al., 2008; Fisher & Vaidyanathan, 2014; Qi et al., 2020; Liu et al., 2025). As digital platforms scale and product catalogs grow, the ability to optimize assortments efficiently becomes increasingly important for both economic performance and user experience.

Despite its practical importance, general AO remains challenging, especially under flexible or data-driven choice models (Jagabathula, 2014; Strauss et al., 2018). The difficulty stems from both the combinatorial nature of the decision space, which grows exponentially with the number of products, and the nonlinear substitution effects induced by customer choice behavior, where adding or removing one product can alter the demand of many others (Train, 2009; Farias et al., 2013; Chen & Mišić, 2022). Existing approaches therefore often rely on parametric choice models such as the *multinomial logit (MNL)* and its variants, which admit tractable formulations and specialized optimization procedures (McFadden, 1974; Manski et al., 1981; McFadden & Train, 2000; Kök et al., 2008; Feng et al., 2022). However, these methods can be sensitive to model misspecification in the presence of heterogeneous preferences or complex substitution patterns (Train, 2009; Farias et al., 2013; Bertsimas & Mišic, 2015; Sturt, 2025), and scalable optimization remains difficult for richer models and large-scale or constrained settings (Bront et al., 2009; Jagabathula et al., 2020; Désir et al., 2020). These challenges are further amplified in the offline setting, where optimization must rely solely on historical interaction data without active exploration (Dong et al., 2025; Han et al., 2025). Developing methods that are both computationally efficient and robust to model misspecification in this setting remains an open challenge.

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.

In parallel, *diffusion-based generative models* have emerged as a powerful framework for modeling complex high-dimensional distributions (Ho et al., 2020; Song et al., 2020a;b; Chen et al., 2024). Beyond standard generation tasks, recent work suggests that diffusion models can also serve as effective stochastic optimizers for structured and combinatorial spaces (Sun & Yang, 2023; Zhao et al., 2024). This perspective is particularly appealing for AO: rather than solving a difficult combinatorial optimization problem directly, one may learn to generate high-quality assortments through iterative refinement. Such a generative approach naturally supports exploration and can produce multiple competitive candidate assortments under limited computational budgets.

In this paper, we propose **Diffusion-based Data-Driven Assortment Optimization (D3AO)**, a generative framework for offline AO. Our method learns a diffusion prior over binary assortment vectors and uses reward-guided refinement to steer generation toward high-value regions of the combinatorial space. This enables efficient search over large discrete action spaces without explicit reliance on a restrictive parametric choice model. Empirically, D3AO consistently identifies high-quality assortments and remains robust under model misspecification. Our main contributions are summarized as follows:

- *A new generative formulation of AO.* We cast AO as a sampling problem over combinatorial structures and introduce a diffusion-based framework tailored to discrete decision spaces.
- *Offline optimization with implicit behavior modeling.* Our method avoids explicit parametric choice model estimation, improving robustness under model misspecification.
- *Efficient generation of high-quality assortments.* The proposed approach produces multiple near-optimal solutions via stochastic generation, avoiding exhaustive combinatorial search.
- *Empirical validation in controlled settings.* Through experiments on synthetic benchmarks, we show that our method consistently identifies high-quality assortments and remains competitive under correct specification while substantially outperforming misspecified parametric baselines.

1.1 Related Works

Assortment Optimization. Assortment optimization has been extensively studied under a variety of customer choice models (McFadden, 1974; Talluri & Van Ryzin, 2004; Kök et al., 2008; Strauss et al., 2018). A major line of work focuses on parametric models, such as MNL and its variants, which admit tractable formulations and specialized algorithms (Manski et al., 1981; Rusmevichientong et al., 2010; Feng et al., 2022). To relax restrictive structural assumptions, subsequent work considers nonparametric and data-driven choice models (Farias et al., 2013; Bertsimas & Mišic, 2015), often at the cost of greater computational complexity. A large related literature studies online AO, where the decision-maker sequentially learns customer preferences while optimizing assortments (Rusmevichientong et al., 2010; Sauré & Zeevi, 2013; Gong et al., 2022; Li et al., 2025). More recently, offline AO has received increasing attention, where decisions must be learned from historical data without active exploration (Wang et al., 2023a; Dong et al., 2025; Han et al., 2025).

Diffusion-based Generative Models. Diffusion models provide a flexible framework for learning complex distributions through iterative denoising (Ho et al., 2020; Song et al., 2020a;b). While originally developed for continuous domains, they have been extended to discrete state spaces (Austin et al., 2021; Sun et al., 2022), enabling applications to categorical and combinatorial objects. Recent work further develops guidance mechanisms for steering generation toward desired properties (Bansal et al., 2023; Schiff et al., 2025). Building on these advances, diffusion-based methods have recently been applied to combinatorial optimization (Sun & Yang, 2023; Sanokowski et al., 2024; Zhao et al., 2024; Wang et al., 2025; Aloui et al., 2026). Our work brings this perspective to offline assortment optimization.

Offline Learning. Offline learning studies how to make decisions using historical interaction data without active exploration (Swaminathan & Joachims, 2015; Prudencio et al., 2023). A central challenge is distribution shift induced by the behavioral policy, which leads to selection bias, limited coverage, and unreliable extrapolation (Dudík et al., 2011; Joachims et al., 2018; Levine et al., 2020). Prior work addresses these issues through off-policy evaluation, counterfactual learning, and pessimistic or conservative decision making (Fujimoto et al., 2019; Kumar et al., 2020). Our setting shares these challenges, but is further complicated by combinatorial action spaces and customer choice–dependent rewards (Shimizu et al., 2024). We address this setting through a guided generative optimization framework tailored to assortment decisions.

2 OFFLINE ASSORTMENT OPTIMIZATION

We first formalize the offline assortment optimization problem studied in this paper.

Problem formulation. Let $[N] = \{1, \dots, N\}$ denote the set of products, and let $\mathscr{S} \subseteq 2^{[N]} \setminus \{\emptyset\}$ denote the feasible set of assortments, which may encode cardinality or business constraints. We augment the product set with an outside option 0, representing no purchase. For each assortment $s \in \mathscr{S}$, a customer chooses an item $A \in s \cup \{0\}$ according to a choice policy $p(\cdot \mid s)$, and the seller receives reward $r(s, A)$, where typically $r(s, 0) = 0$ and $r(s, a) = 0$ for $a \notin s$.

The expected revenue of an assortment s is

$$R(s) = \mathbb{E}_{A \sim p(\cdot|s)}[r(s, A)] = \sum_{a \in [N] \cup \{0\}} r(s, a)\, p(a \mid s) = \sum_{a \in s} r(s, a)\, p(a \mid s),$$

and the assortment optimization problem is $s^\star \in \arg\max_{s \in \mathscr{S}} R(s)$.

This problem is difficult for two main reasons. First, the decision space is combinatorial, with $|\mathscr{S}|$ typically growing exponentially in N. Second, customer choice induces nonlinear substitution effects, so the value of one product depends on the others offered in the assortment. As a result, exact optimization quickly becomes intractable beyond small-scale settings. In general, when the choice behavior is flexible or data-driven, the optimization problem becomes computationally intractable (often NP-hard), and admits no efficient exact solution beyond small-scale instances (Bront et al., 2009; Jagabathula, 2014).

Offline learning setting. We consider the offline setting, where the decision-maker only has access to logged assortment-choice interactions (Wang et al., 2023a; Dong et al., 2025; Han et al., 2025). The dataset is $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$, where $S_i \in \mathscr{S}$ is a historical assortment and $A_i \in S_i \cup \{0\}$ is the observed customer choice. We assume the reward function $r(s, a)$ is known, as in standard revenue-management settings where rewards correspond to prices or margins. Unlike the online setting, the learner cannot query new assortments or actively improve coverage. The observed assortments are generated by a behavior policy μ over $\mathscr{S}$, which may concentrate on only a small and biased portion of the combinatorial space. Consequently, optimal or near-optimal assortments may be rarely observed or entirely absent from the data, making evaluation and optimization of unseen assortments difficult.

2.1 A MAXIMUM ENTROPY PERSPECTIVE ON ASSORTMENT OPTIMIZATION

In the offline setting, the observed assortments $\{S_i\}_{i=1}^n$ are generated by a behavior policy μ over the feasible set $\mathscr{S}$. In practice, these assortments are typically produced by heuristics, learned policies, or human operators, and are therefore neither uniformly random nor globally optimal. Instead, they are often biased toward higher-reward assortments while still retaining some stochasticity or exploration (Ziebart et al., 2008; Levine et al., 2020). To capture this, we model the behavior policy as a Boltzmann distribution

$$\mu(s) \propto \exp\big(\beta_0 R(s)\big), \qquad s \in \mathscr{S}, \tag{1}$$

where $\beta_0 > 0$ controls the degree of concentration on high-reward assortments. Small β_0 yields a nearly uniform policy, while large β_0 concentrates on near-optimal assortments.

Importantly, equation 1 is not merely a modeling assumption: it arises as the solution to an entropy-regularized revenue maximization problem.

Theorem 1 (Entropy-regularized characterization of Boltzmann policies)**.** *Let $\mathscr{S} \subset \{0, 1\}^N$ be a finite set of feasible assortments and let $R : \mathscr{S} \to \mathbb{R}$ be a reward function. Consider*

$$\max_{\mu \in \Delta(\mathscr{S})} \ \mathbb{E}_{s \sim \mu}[R(s)] + \frac{1}{\beta} \mathcal{H}(\mu),$$

where $\mathcal{H}(\mu) = -\sum_{s \in \mathscr{S}} \mu(s) \log \mu(s)$ is the Shannon entropy (Shannon, 1948), and $\beta > 0$. Then the unique maximizer is

$$\mu^\star(s) = \frac{\exp(\beta R(s))}{\sum_{s' \in \mathscr{S}} \exp(\beta R(s'))}, \qquad s \in \mathscr{S}.$$

Proof. See Appendix §C.1. □

Theorem 1 shows that the Boltzmann policy is a soft-optimal decision rule that balances reward maximization and exploration through entropy regularization. This perspective motivates our approach: rather than searching directly for a single optimal assortment, we construct a distribution that concentrates on high-reward regions of the combinatorial space while preserving diversity. Diffusion models provide a natural mechanism for learning and refining such distributions.

Connection to maximum-entropy IRL. This viewpoint is closely related to maximum-entropy inverse reinforcement learning (Ziebart et al., 2008; Wulfmeier et al., 2015; Snoswell et al., 2020), where trajectories satisfy $p(\tau) \propto q(\tau)\exp\big(R(\tau)\big)$, with $q(\tau)$ induced by the dynamics and $R(\tau)$ the cumulative reward. Our setting can be viewed as a degenerate one-step version of this framework, where assortments play the role of actions and the base measure is implicit. Under this view, the historical data is generated by a soft-optimal policy, and our goal is to shift mass toward higher-reward assortments while retaining sufficient exploration.

3 D3AO: Diffusion-based Data-Driven Assortment Optimization

We now present *Diffusion-based Data-Driven Assortment Optimization* (D3AO), a three-stage framework for offline assortment optimization. Given historical assortment-choice data, D3AO first learns a neural choice model, then constructs a plug-in reward estimator, and finally performs guided stochastic optimization using a discrete diffusion model. An overview is shown in Figure 1.

Step 1: Choice model estimation. Given offline data $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$, we learn a neural choice model $p_\theta(a \mid s)$ that maps an assortment $s \in \mathscr{S}$ to a probability distribution over $[N] \cup \{0\}$, with support restricted to $s \cup \{0\}$. The model is trained by maximum likelihood:

$$\hat{\theta} \in \arg\min_\theta -\frac{1}{n}\sum_{i=1}^n \log p_\theta(A_i \mid S_i).$$

In practice, we adopt the neural choice models of Wang et al. (2023a), which output masked probabilities over feasible items. This flexible parameterization allows us to capture nonlinear substitution patterns beyond standard parametric models.

Step 2: Reward estimation. Using the fitted choice model, we estimate the expected reward of any assortment s by

$$R_{\hat{\theta}}(s) = \sum_{a \in s \cup \{0\}} r(s,a)\, p_{\hat{\theta}}(a \mid s).$$

This plug-in estimator replaces the unknown choice policy with the learned model and provides a tractable surrogate objective for evaluating candidate assortments in the offline setting.

Step 3: Guided generative optimization. Since the offline data are collected by the Boltzmann-like behavioral policy $\mu(s) \propto \exp(\beta_0 R(s))$, we use a discrete diffusion model to learn and improve a generative approximation of the historical assortment distribution. First, the diffusion model is trained on offline assortments $\{S_i\}_{i=1}^n$ to learn a generative prior $\hat{\mu}$ over plausible assortments. Concretely, the forward process gradually corrupts assortments toward a simple reference distribution, $\mu = \mu_0 \to \cdots \to \mu_T \approx \mathrm{Uniform}(\{0,1\}^N)$, and the learned reverse process approximately inverts this trajectory, $\mathrm{Uniform}(\{0,1\}^N) = \hat{\mu}_T \to \cdots \to \hat{\mu}_0 = \hat{\mu}$. Sampling from $\hat{\mu}$ alone mainly reproduces the historical policy. We therefore guide the reverse denoising process using the estimated reward $R_{\hat{\theta}}$, biasing generation toward higher-value assortments while preserving the structural prior learned from data. Further details of the diffusion formulation and guidance mechanism are given in §4.

Overall, D3AO combines a learned behavior prior with reward-guided refinement to perform policy improvement in the combinatorial assortment space.

4 Guided Discrete Diffusion

We model assortments as binary vectors and use a discrete diffusion process to generate candidate solutions through iterative denoising. Throughout this section, we focus on the unconstrained setting

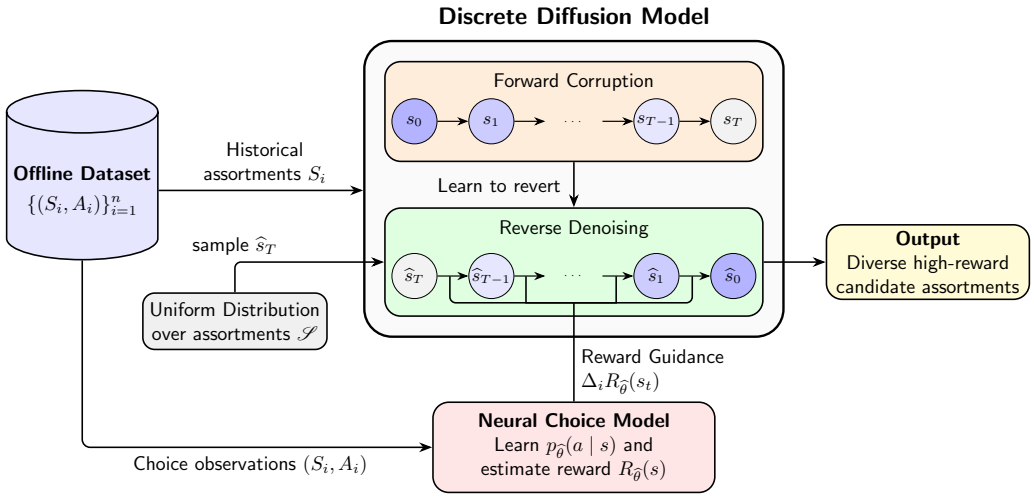


Figure 1: Overview of D3AO. From offline assortment-choice data, we learn a neural choice model and a discrete diffusion prior over assortments. During sampling, the reverse diffusion process is guided by the estimated reward to generate high-quality candidate assortments.

$\mathscr{S} = \{0,1\}^N$. Unlike standard discrete diffusion, our goal is not exact generative modeling, but reward-guided combinatorial optimization.

4.1 Denoising Discrete Diffusion

We first introduce the underlying denoising diffusion model over binary assortments, which serves as the generative backbone of our approach.

Forward process. We adopt a discrete-time Bernoulli corruption process that progressively randomizes an assortment vector. Starting from a clean assortment $s_0 \in \{0,1\}^N$, the forward process generates a sequence $\{s_t\}_{t=0}^T$ according to

$$q(s_{1:T} \mid s_0) = \prod_{t=1}^{T} q_t(s_t \mid s_{t-1}), \tag{2}$$

where each transition acts independently across coordinates:

$$q_t(s_{t,i} \mid s_{t-1,i}) = (1-\beta_t)\mathbf{1}\{s_{t,i} = s_{t-1,i}\} + \frac{\beta_t}{2}, \qquad s_{t,i} \in \{0,1\}.$$

Equivalently, with probability β_t, the i-th bit is resampled uniformly from $\{0,1\}$; otherwise it is unchanged, and $q_t(s_t \mid s_{t-1}) = \prod_{i=1}^N q_t(s_{t,i} \mid s_{t-1,i})$. As t increases, the distribution approaches a near-uniform law over $\{0,1\}^N$. Appendix §C.2 gives a finite-time total-variation bound in terms of $\bar{\alpha}_t = \prod_{\tau=1}^t (1-\beta_\tau)$.

Denoising parameterization. The reverse model is parameterized by a neural network that predicts the clean assortment s_0 from a noisy state (s_t, t). Specifically, the network outputs logits $g_\phi(s_t, t) \in \mathbb{R}^N$, which define Bernoulli probabilities $\alpha_{t,i} = \sigma\big(g_\phi^{(i)}(s_t, t)\big)$ for $i \in [N]$, and hence a factorized distribution over the clean sample:

$$\tilde{p}_\phi(\tilde{s}_0 \mid s_t, t) = \prod_{i=1}^{N} \alpha_{t,i}^{\tilde{s}_{0,i}} (1-\alpha_{t,i})^{1-\tilde{s}_{0,i}}.$$

Let $\mathcal{D}$ denote the empirical distribution of assortments in the offline dataset. We sample $s_0 \sim \mathcal{D}$, $t \sim \mathrm{Unif}\{1,\dots,T\}$, and $s_t \sim q_t(\cdot \mid s_0)$, and train the network by binary cross-entropy:

$$\mathcal{L}(\phi) = \mathbb{E}_{s_0 \sim \mathcal{D}} \mathbb{E}_{t \sim \mathrm{Unif}\{1,\dots,T\}} \mathbb{E}_{s_t \sim q_t(\cdot \mid s_0)} \left[-\sum_{i=1}^{N} \big(s_{0,i} \log \alpha_{t,i} + (1-s_{0,i}) \log(1-\alpha_{t,i}) \big) \right].$$

This objective trains g_ϕ to approximate the clean-assortment logits from corrupted inputs. See Appendix §C.3 for its population posterior-marginal characterization.

Optimization-oriented reverse update. In standard D3PM-style discrete diffusion (Austin et al., 2021), the clean-sample predictor is combined with the forward posterior to define the reverse kernel

$$p_\phi(s_{t-1} \mid s_t, t) \propto \sum_{\tilde{s}_0 \in \mathscr{S}} q(s_{t-1}, s_t \mid \tilde{s}_0)\, \tilde{p}_\phi(\tilde{s}_0 \mid s_t, t). \tag{3}$$

In our setting, however, the goal is efficient generation of high-reward assortments rather than accurate recovery of the reverse diffusion dynamics. We therefore use the predicted clean-sample distribution directly as the reverse proposal $p_\phi(s_{t-1} \mid s_t, t) := \tilde{p}_\phi(s_{t-1} \mid s_t, t)$, i.e., $s_{t-1,i} \sim \mathrm{Bernoulli}(\alpha_{t,i})$ independently for $i \in [N]$. Compared with equation 3, this update is computationally cheaper and acts as an accelerated denoising step. Empirically, it performs better in our reward-guided combinatorial optimization setting.

4.2 Reward Guidance

The denoising model above learns a structured prior over assortments from offline data, but by itself it mainly reproduces the historical behavior distribution. To improve solution quality, we guide the reverse process using the estimated reward $R_{\hat{\theta}}$. Intuitively, the learned reverse model provides a data-driven reference distribution over plausible assortments, while reward guidance performs a local improvement step that shifts probability mass toward higher-value candidates.

Guided reverse process. Directly incorporating the global reward $R_{\hat{\theta}}(s_{t-1})$ into the reverse transition is intractable in the high-dimensional combinatorial space. We therefore adopt a coordinate-wise approximation. For each coordinate $i \in [N]$, let $s_t^{(i=1)}$ and $s_t^{(i=0)}$ denote s_t with its i-th entry set to 1 and 0, respectively, and let

$$\Delta_i R_{\hat{\theta}}(s_t) := R_{\hat{\theta}}(s_t^{(i=1)}) - R_{\hat{\theta}}(s_t^{(i=0)}).$$

This quantity measures the local reward gain of setting the i-th coordinate to one while keeping the remaining coordinates fixed at $s_{t,-i}$. We then modify the reverse logits by

$$\tilde{g}_\phi^{(i)}(s_t, t) = g_\phi^{(i)}(s_t, t) + \lambda_t\, \Delta_i R_{\hat{\theta}}(s_t),$$

where $\lambda_t \geq 0$ controls the guidance strength. The guided transition is $p_{\mathrm{guide}}(s_{t-1,i} = 1 \mid s_t, t) = \sigma\big(\tilde{g}_\phi^{(i)}(s_t, t)\big)$. Thus, the diffusion model provides a prior over plausible assortments, while $\Delta_i R_{\hat{\theta}}(s_t)$ biases each coordinate toward higher estimated reward.

Although the logit-shift guidance rule may appear ad hoc, it admits a simple variational interpretation. The following result shows that it is the closed-form solution of a KL-regularized local improvement problem, with the unguided reverse proposal serving as the reference distribution.

Theorem 2 (Reward guidance as KL-regularized local policy improvement)**.** *Fix a reverse diffusion step t and a current state $s_t \in \{0,1\}^N$. Consider the unguided reverse proposal $p_\phi(x \mid s_t, t) = \prod_{i=1}^N \mathrm{Bernoulli}\big(x_i; \sigma(g_\phi^{(i)}(s_t, t))\big)$ for $x \in \{0,1\}^N$. Given a reward function $R : \{0,1\}^N \to \mathbb{R}$ with local reward differences $\Delta_i R(s_t) = R(s_t^{(i=1)}) - R(s_t^{(i=0)})$, $i \in [N]$, define the coordinate-wise local reward surrogate $\ell_{s_t}^R(x) := \sum_{i=1}^N x_i \Delta_i R(s_t)$. For any $\lambda_t > 0$, consider*

$$\max_{q \in \Delta(\{0,1\}^N)} \left\{ \mathbb{E}_{x \sim q}[\ell_{s_t}^R(x)] - \frac{1}{\lambda_t} \mathrm{KL}\left(q \,\|\, p_\phi(\cdot \mid s_t, t)\right) \right\}.$$

Then the unique optimizer is

$$q_t^\star(x \mid s_t) = \frac{p_\phi(x \mid s_t, t) \exp(\lambda_t \ell_{s_t}^R(x))}{Z_t(s_t)},$$

where $Z_t(s_t)$ is the normalizing constant. Moreover, $q_t^\star(\cdot \mid s_t)$ factorizes across coordinates and satisfies

$$q_t^\star(x_i = 1 \mid s_t) = \sigma\left(g_\phi^{(i)}(s_t, t) + \lambda_t \Delta_i R(s_t)\right).$$

Consequently, the logit-shift guidance rule is the closed-form solution of a KL-regularized local policy improvement step.

By Theorem 2, if the true reward R were available, the guided transition would be the exact KL-regularized improvement of the unguided reverse proposal under the local surrogate $\ell^R_{s_t}$. In the offline setting, D3AO implements a plug-in version of this update by replacing unknown R with the model-based estimate $R_{\hat{\theta}}$, so the practical logit shift $\tilde{g}^{(i)}_{\phi}$ above can be understood as an estimated local policy improvement step.

This interpretation is closely related to policy-based methods in reinforcement learning, especially KL-regularized policy improvement (Schulman et al., 2015; 2017). In these methods, policy updates favor actions with high reward or advantage while controlling deviation from a reference policy through a KL constraint or penalty. Analogously, in our setting, the unguided reverse proposal serves as the reference policy, the coordinate-wise surrogate $\ell^R_{s_t}$ serves as a local reward or advantage signal, and the KL term keeps the guided transition close to the learned behavior prior.

Guidance schedule. The reward estimate is less reliable when s_t is highly corrupted, so guidance should be weak in the early reverse steps and stronger near the end of denoising. This yields a coarse-to-fine search strategy: the reverse process first explores broad high-reward regions induced by the learned prior, and then progressively refines samples toward better assortments. Accordingly, we use the schedule

$$\lambda_t = \lambda_{\max} \left(1 - \frac{t-1}{T-1}\right)^{\gamma}, \qquad t = 1, \dots, T, \tag{4}$$

where $\lambda_{\max} > 0$ is the maximum guidance strength and $\gamma \geq 1$ controls how sharply guidance is concentrated near the final denoising steps. In particular, $\lambda_T \approx 0$ suppresses guidance for highly noisy states, while $\lambda_1 = \lambda_{\max}$ applies the strongest guidance at the final step.

Sampling. Starting from $s_T \sim \mathrm{Uniform}(\{0,1\}^N)$, we generate samples via $s_{t-1} \sim p_{\mathrm{guide}}(\cdot \mid s_t, t)$ for $t = T, \dots, 1$. This guided process combines the learned structural prior with reward-driven refinement, allowing us to generate multiple high-quality candidate assortments without explicit combinatorial search.

5 Numerical Experiments

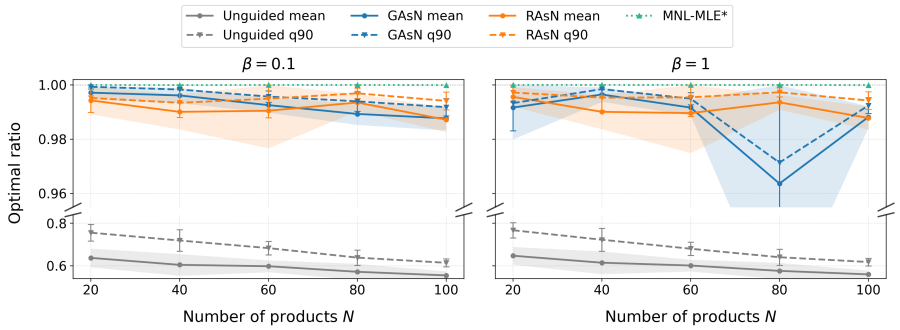
We evaluate D3AO on synthetic offline assortment optimization problems generated from standard choice models. The experiments are designed to answer whether guided diffusion can identify high-reward assortments and remain robust when classical parametric baselines are misspecified. The main text focuses on primary solution-quality performance, while Appendix §B provides implementation details, full numerical tables, and complementary analyses on exact recovery, diversity, sample efficiency, and behavior-policy distribution shift.

Setting. We generate offline datasets from three choice models of increasing complexity: MNL, MCCM, and MMNL; see Appendix §A.1 for details. We vary the number of products as $N \in \{20, 40, 60, 80, 100\}$ and use $n = 10{,}000$ offline samples. Historical assortments are drawn from a Boltzmann behavior policy with inverse temperature $\beta \in \{0.1, 1.0\}$. For each configuration of choice model, N, and β, we generate 10 independent datasets.

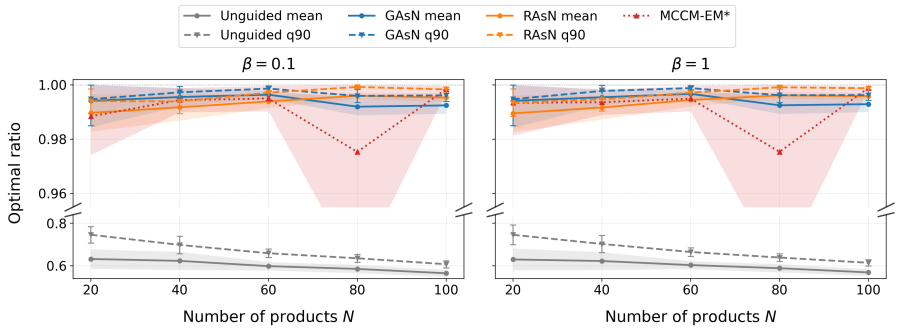
Baselines. We compare with two classical estimate-then-optimize baselines. **MNL-MLE** estimates utilities under the MNL model by maximum likelihood and optimizes the fitted model using revenue-ordered policies (Talluri & Van Ryzin, 2004). **MCCM-EM** estimates MCCM parameters using the EM algorithm (Şimşek & Topaloglu, 2018) and optimizes the fitted model using the LP-based polynomial-time method for MCCMs (Blanchet et al., 2016). MNL-MLE is correctly specified on MNL data, MCCM-EM is correctly specified on MCCM data, and both baselines are misspecified on MMNL data. Each parametric baseline returns a single assortment.

D3AO. We instantiate D3AO using the neural choice models GAsN and RAsN (Wang et al., 2023a) for reward estimation, and then apply guided reverse diffusion to generate 256 candidate assortments per instance. We also include an unguided variant without reward guidance to isolate the effect of the guidance mechanism. Implementation details, including network architectures, diffusion schedules, and guidance parameters, are provided in Appendix §A.2.1 and Appendix §B.

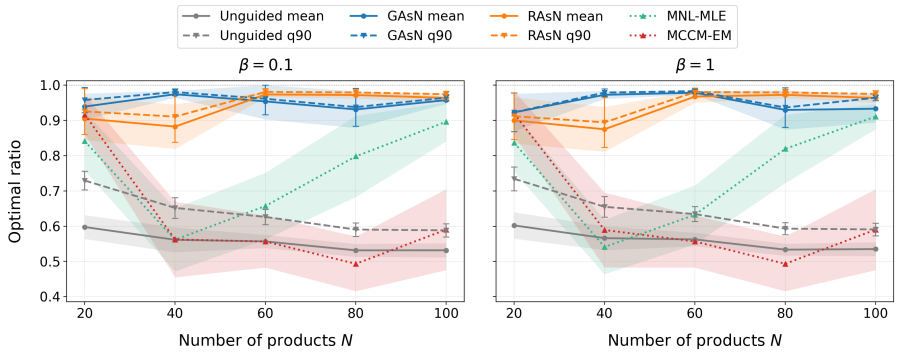
Evaluation protocol. We evaluate solution quality by the optimal ratio $\mathrm{Optimal\,Ratio}(s) := R(s)/R(s^\star)$ for $s \in \mathscr{S}$, where $R(s)$ is the true expected reward under the ground-truth choice model and $s^\star$ is an optimal assortment. Parametric baselines return a single solution, for which we report



(a) MNL data. The oracle baseline MNL-MLE* achieves near-optimal performance, while GAsN- and RAsN-guided methods remain highly competitive.



(b) MCCM data. Both GAsN- and RAsN-guided methods match or outperform the oracle baseline MCCM-EM*, while unguided sampling remains substantially suboptimal.



(c) MMNL data. Under model misspecification, guided diffusion methods significantly outperform parametric baselines and unguided sampling.

Figure 2: Optimal ratio across problem sizes N under different choice models. Solid lines denote the mean optimal ratio, shaded regions indicate 95% confidence intervals for the mean, dashed lines denote the 90th percentile (`q90`), and error bars indicate 95% confidence intervals for `q90`, all computed over 10 runs.

the corresponding optimal ratio. For diffusion-based methods, we report summary statistics over the 256 generated candidates, including maximum, mean, median (`q50`), and 90th percentile (`q90`).

Results. Figure 2 summarizes the main results, while Tables 2–4 in Appendix §B report the detailed numerical values.

On MNL data [Figure 2(a)], the correctly specified oracle baseline MNL-MLE is nearly optimal. Unguided sampling performs much worse, while both guided variants remain highly competitive, with maximum optimal ratios consistently near 100% and strong mean and upper-quantile performance.

On MCCM data [Figure 2(b)], MCCM-EM again performs strongly as the correctly specified oracle baseline, but both guided variants remain close to optimal across problem sizes and sometimes exceed the single solution returned by MCCM-EM. Unguided sampling remains substantially worse.

The largest gap appears on MMNL data [Figure 2(c)], where both parametric baselines are misspecified. In this setting, MNL-MLE and MCCM-EM degrade substantially and become unstable across problem sizes, whereas both guided variants remain consistently strong and clearly outperform the misspecified baselines.

Overall, these results show that reward guidance is essential to D3AO's performance, which enables guided diffusion to match correctly specified parametric baselines while remaining substantially more robust under model misspecification.

6 Discussion and Conclusion

In this paper, we proposed D3AO, a model-agnostic framework for offline assortment optimization based on guided discrete diffusion. By representing assortments as binary vectors, learning a generative behavior prior from historical data, and guiding reverse sampling with reward estimates, D3AO performs stochastic search without explicit combinatorial enumeration or restrictive parametric choice assumptions. Empirically, the method generates high-quality assortments, remains competitive with correctly specified parametric baselines, and is substantially more robust under model misspecification. The generative nature of the approach also enables multiple candidate assortments rather than a single deterministic solution.

More broadly, our work highlights the potential of generative modeling for data-driven combinatorial optimization, while also pointing to several limitations and future directions. In particular, performance depends on the informativeness of the offline data and the accuracy of reward estimation; when the historical assortments provide weak structure, the learned prior may offer limited guidance. Future work includes reducing this dependence on data regularity, extending the method to constrained feasible sets, and exploring connections with continuous relaxation and Langevin-type sampling methods.

AI use statement

In this work, we used generative AI tools for implementing methods, in the form of assistance with writing and editing the experimental code. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, or interpreting results, and translation, cleaning and reformatting data sets, and qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code and for editing the paper to improve readability. We have reviewed all AI-assisted work. All AI-assisted code was reviewed line by line and tested for correctness by the authors, and all AI-edited text was checked by the authors to ensure that it preserves the intended technical meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work studies offline assortment optimization using synthetic datasets generated from standard customer choice models. It does not involve human participants or the collection of personal data. Although the proposed method may improve assortment decisions, deployment using real customer data could reproduce biases in historical choices, and revenue maximization alone does not ensure fairness or consumer welfare. Practical applications would therefore require assessment of data privacy, distributional effects, and appropriate operational constraints. Our experiments are limited to synthetic settings and do not establish these deployment properties.

Reproducibility statement

The main text describes the proposed method and evaluation protocol. Appendix A.2.1 describes the model architectures, and Appendix B.1 provides hyperparameters, training configurations, and computational resources; additional experimental details are given in Appendix B. The synthetic choice models are described in Appendix A.1, and complete proofs of the theoretical results are provided in Appendix C.

References

Ahmed Aloui, Junyi Liao, Ali Hasan, Jose Blanchet, and Vahid Tarokh. Score-based metropolis-hastings for fractional langevin algorithms. *arXiv preprint arXiv:2602.00835*, 2026.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 843–852, 2023.

Dimitris Bertsimas and Velibor V Mišic. Data-driven assortment optimization. *Management Science*, 1:1–35, 2015.

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

Juan José Miranda Bront, Isabel Méndez-Díaz, and Gustavo Vulcano. A column generation algorithm for choice-based network revenue management. *Operations research*, 57(3):769–784, 2009.

Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *arXiv preprint arXiv:2404.07771*, 2024.

Yi-Chun Chen and Velibor V Mišić. Decision forest: A nonparametric approach to modeling irrational choice. *Management Science*, 68(10):7090–7111, 2022.

Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Constrained assortment optimization under the markov chain–based choice model. *Management Science*, 66(2):698–721, 2020.

Juncheng Dong, Weibin Mo, Zhengling Qi, Cong Shi, Ethan X Fang, and Vahid Tarokh. Pasta: A unified framework for offline assortment learning. *arXiv preprint arXiv:2510.01693*, 2025.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

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

Qi Feng, J George Shanthikumar, and Mengying Xue. Consumer choice models and estimation: A review and extension. *Production and Operations Management*, 31(2):847–867, 2022.

Marshall Fisher and Ramnath Vaidyanathan. A demand estimation procedure for retail assortment optimization with results from implementations. *Management Science*, 60(10):2401–2415, 2014.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Xiao-Yue Gong, Vineet Goyal, Garud N Iyengar, David Simchi-Levi, Rajan Udwani, and Shuangyu Wang. Online assortment optimization with reusable resources. *Management Science*, 68(7):4772–4785, 2022.

Yuxuan Han, Han Zhong, Miao Lu, Jose Blanchet, and Zhengyuan Zhou. Learning an optimal assortment policy under observational data. *arXiv preprint arXiv:2502.06777*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Srikanth Jagabathula. Assortment optimization under general choice. *Available at SSRN 2512831*, 2014.

Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science*, 66(8):3635–3656, 2020.

Thorsten Joachims, Adith Swaminathan, and Maarten De Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

A Gürhan Kök, Marshall L Fisher, and Ramnath Vaidyanathan. Assortment planning: Review of literature and industry practice. *Retail supply chain management: Quantitative models and empirical studies*, pp. 99–153, 2008.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Shukai Li, Qi Luo, Zhiyuan Huang, and Cong Shi. Online learning for constrained assortment optimization under markov chain choice model. *Operations research*, 73(1):109–138, 2025.

Yicheng Liu, Xiao Alison Chen, Yan Liu, and Zizhuo Wang. Simultaneous vs. sequential: Optimal assortment recommendation in multistore retailing. *Manufacturing & Service Operations Management*, 27(3):825–842, 2025.

Charles F Manski, Daniel McFadden, et al. *Structural analysis of discrete data with econometric applications*. MIT press Cambridge, MA, 1981.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In Paul Zarembka (ed.), *Fontiers in Econometrics*, pp. 105–142. Academic press, 1974.

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

Isabel Méndez-Díaz, Juan José Miranda-Bront, Gustavo Vulcano, and Paula Zabala. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics*, 164:246–263, 2014.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE transactions on neural networks and learning systems*, 35(8):10237–10257, 2023.

Meng Qi, Ho-Yin Mak, and Zuo-Jun Max Shen. Data-driven research in retail operations—a review. *Naval Research Logistics (NRL)*, 67(8):595–616, 2020.

Robert P Rooderkerk, Harald J Van Heerde, and Tammo HA Bijmolt. Optimizing retail assortments. *Marketing Science*, 32(5):699–715, 2013.

Paat Rusmevichientong, Zuo-Jun Max Shen, and David B Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research*, 58(6):1666–1680, 2010.

Paat Rusmevichientong, David Shmoys, Chaoxu Tong, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23 (11):2023–2039, 2014.

Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. *arXiv preprint arXiv:2406.01661*, 2024.

Denis Sauré and Assaf Zeevi. Optimal dynamic assortment planning with demand learning. *Manufacturing & Service Operations Management*, 15(3):387–404, 2013.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Alexander Rush, Volodymyr Kuleshov, Hugo Dalla-Torre, Sam Boshar, Bernardo P de Almeida, and Thomas Pierrot. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 44153, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423, 1948.

Tatsuhiro Shimizu, Koichi Tanaka, Ren Kishimoto, Haruka Kiyohara, Masahiro Nomura, and Yuta Saito. Effective off-policy evaluation and learning in contextual combinatorial bandits. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 733–741, 2024.

A Serdar Şimşek and Huseyin Topaloglu. An expectation-maximization algorithm to estimate the parameters of the markov chain choice model. *Operations Research*, 66(3):748–760, 2018.

Aaron J Snoswell, Surya PN Singh, and Nan Ye. Revisiting maximum entropy inverse reinforcement learning: New perspectives and algorithms. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 241–249. IEEE, 2020.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Arne K Strauss, Robert Klein, and Claudius Steinhardt. A review of choice-based revenue management: Theory and methods. *European journal of operational research*, 271(2):375–387, 2018.

Bradley Sturt. The value of robust assortment optimization under ranking-based choice models. *Management Science*, 71(5):4246–4265, 2025.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in neural information processing systems*, 36:3706–3731, 2023.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International conference on machine learning*, pp. 814–823. PMLR, 2015.

Kalyan Talluri and Garrett Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management science*, 50(1):15–33, 2004.

Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

Hanzhao Wang, Zhongze Cai, Xiaocheng Li, and Kalyan Talluri. A neural network based choice model for assortment optimization. *arXiv preprint arXiv:2308.05617*, 2023a.

Hanzhao Wang, Xiaocheng Li, and Kalyan Talluri. Transformer choice net: A transformer neural network for choice prediction. *arXiv preprint arXiv:2310.08716*, 2023b.

Shiyue Wang, Ziao Guo, Changhong Lu, and Junchi Yan. Fractional langevin dynamics for combinatorial optimization via polynomial-time escape. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

Hang Zhao, Kexiong Yu, Yuhang Huang, Renjiao Yi, Chenyang Zhu, and Kai Xu. Disco: Efficient diffusion solver for large-scale combinatorial optimization problems. *arXiv preprint arXiv:2406.19705*, 2024.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Model Details

A.1 Choice Models

In this subsection, we introduce several benchmark choice models based on different assumptions about customer behavior. Given an assortment $s \in \mathscr{S}$, each model defines a choice policy $p(\cdot \mid s)$ over $s \cup \{0\}$, where 0 denotes the no-purchase option.

Multinomial Logit (MNL). The MNL model is a classical random-utility model (McFadden, 1974; Talluri & Van Ryzin, 2004; Train, 2009). For each $i \in [N] \cup \{0\}$, let

$$U_i = u_i + \varepsilon_i,$$

where u_i is the deterministic utility component and the random noise terms $\{\varepsilon_i\}$ are i.i.d. type-I extreme-value (Gumbel). Since only utility differences matter, one typically normalizes the outside option utility, e.g., $u_0 = 0$. Under this assumption, the choice probabilities admit the closed-form expression

$$p(i \mid s) = \frac{\exp(u_i)}{\sum_{j \in s \cup \{0\}} \exp(u_j)}, \qquad i \in s \cup \{0\}.$$

A key advantage of the MNL model is that the assortment optimization problem admits an efficient solution. In particular, it is well known that an optimal assortment can be obtained via a revenue-ordered (RO) policy, which selects a prefix of products sorted by revenue (Talluri & Van Ryzin, 2004). This structural property enables polynomial-time optimization and makes MNL a widely used benchmark in assortment optimization.

Parameter generation. To generate synthetic MNL instances, we assign utilities $\{u_i\}_{i=0}^{N}$ independently. The no-purchase option is assigned zero utility, i.e., $u_0 = 0$, while product utilities are sampled as

$$u_1, \cdots, u_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\, 1).$$

This construction yields a homogeneous preference structure across products, consistent with the standard MNL model.

Markov Chain Choice Model (MCCM). The Markov chain choice model is introduced by Blanchet et al. (2016), which defines a discrete-time Markov chain on the state space $[N] \cup \{0\}$. A customer starts from an initial state distributed according to $\lambda \in \Delta([N] \cup \{0\})$, i.e.

$$\sum_{i \in [N] \cup \{0\}} \lambda_i = 1, \quad \lambda_0, \lambda_1, \cdots, \lambda_N \geq 0.$$

If the current state belongs to $s \cup \{0\}$, the process stops and that state is chosen; otherwise, the customer transitions according to a transition matrix ρ, where $\rho_{ij} \geq 0$ and $\sum_{j \in [N] \cup \{0\}} \rho_{ij} = 1$. Let $\{X_t\}_{t \geq 0}$ denote the chain and define the first hitting time

$$\tau = \inf\{t \geq 0 : X_t \in s \cup \{0\}\}.$$

Then the induced choice probabilities are

$$p(i \mid s) = \mathbb{P}(X_\tau = i), \qquad i \in s \cup \{0\}.$$

Compared with MNL, the MCCM framework can capture richer substitution patterns, while the MNL model can be viewed as a special case of MCCM (Blanchet et al., 2016). Parameter estimation for MCCM can be efficiently performed via expectation-maximization (EM) algorithms (Şimşek & Topaloglu, 2018). Moreover, Blanchet et al. (2016) shows that the assortment optimization problem under MCCM admits a polynomial-time solution via a linear programming (LP)-based formulation, making it a flexible yet tractable extension of classical choice models.

Parameter generation. To generate synthetic MCCM instances, we randomly sample the initial distribution λ and transition matrix ρ. Specifically, we sample λ from a symmetric Dirichlet distribution:

$$\lambda \sim \text{Dirichlet}(\alpha \mathbf{1}_{N+1}),$$

where $\alpha > 0$ is a concentration parameter.

The transition matrix ρ is constructed row-wise. The no-purchase state is made absorbing, i.e., $\rho_{00} = 1$. For each product state $i \in [N]$, we sample transition probabilities over all other states using a Dirichlet distribution:

$$(\rho_{i,0}, \rho_{i,1}, \cdots, \rho_{i,i-1}, \rho_{i,i+1}, \cdots, \rho_{i,N}) \sim \text{Dirichlet}(\alpha \mathbf{1}_N), \qquad \rho_{ii} = 0.$$

This construction provides a flexible and unbiased way to generate valid stochastic matrices while avoiding degenerate structures. The symmetric Dirichlet distribution induces diverse transition patterns across products, enabling rich context-dependent substitution effects. In our experiments, we set $\alpha = 1$ to obtain moderately heterogeneous yet well-conditioned instances.

Mixed Multinomial Logit (MMNL). MMNL extends MNL by allowing customer-level preference heterogeneity (McFadden & Train, 2000; Train, 2009). In our experiments, we use a version with a finite number of customer types, which is also standard in assortment optimization (Méndez-Díaz et al., 2014). Let $\mathcal{C}$ be the set of customer types, let α_c be the probability of type c, and let $u_{c,i}$ be the utility of item i for type c. Then

$$p(i \mid s) = \sum_{c \in \mathcal{C}} \alpha_c \frac{\exp(u_{c,i})}{\sum_{j \in s \cup \{0\}} \exp(u_{c,j})}, \qquad i \in s \cup \{0\},$$

where $\alpha_c \geq 0$ and $\sum_{c \in \mathcal{C}} \alpha_c = 1$. In other words, the population is modeled as a mixture of finitely many MNL-type customer groups with different utility parameters.

While MMNL provides a significantly more expressive model than MNL, it also introduces substantial computational challenges. For small to moderate problem sizes, or for instances with special structure, exact or near-exact solutions can be obtained via mixed-integer or branch-and-bound methods (Méndez-Díaz et al., 2014; Rusmevichientong et al., 2014). However, in general, assortment optimization under MMNL is NP-hard and lacks efficient exact solution methods (Bront et al., 2009; Rusmevichientong et al., 2014).

Parameter generation. To generate synthetic MMNL instances, we consider a finite mixture model with $|\mathcal{C}| = 5$ customer types and uniform mixing weights $\alpha_c = 0.2$ for all $c \in \mathcal{C}$. For each type $c \in \mathcal{C}$, we specify type-dependent utilities $\{u_{c,i}\}_{i=0}^{N}$ as follows:

$$u_{c,i} = \begin{cases} 0, & i = 0, \\ \mathcal{N}(c + N/5,\, 1), & (c-1)\frac{N}{5} + 1 \leq i \leq \frac{cN}{5}, \\ \mathcal{N}(-1,\, 1), & \text{otherwise.} \end{cases}$$

In other words, the no-purchase option is assigned zero utility for every customer type, and the N products are partitioned into five disjoint groups of equal size. For each type c, products in the c-th group are assigned higher utilities, while all remaining products receive lower utilities.

This construction induces structured preference heterogeneity, where each customer type strongly favors a distinct subset of products while maintaining mild overlap across types. As a result, the induced choice behavior exhibits complex substitution patterns that cannot be captured by homogeneous models such as MNL.

A.2 Neural Choice Models

Neural choice models replace a fixed parametric substitution structure by a flexible function class learned directly from data (Wang et al., 2023a;b). For neural models, we encode each assortment $s \in \mathscr{S}$ by its binary incidence vector in $\{0,1\}^N$, where the ith entry equals 1 if and only if item i is offered. Given this representation, a neural network outputs logits for the outside option and all products, followed by a masked softmax that enforces feasibility:

$$\mathbf{Y} = g(s; \theta) \in \Delta([N] \cup \{0\}),$$

where $Y_a = p_\theta(a \mid s)$ for $a \in [N] \cup \{0\}$ and $p_\theta(i \mid s) = 0$ whenever $i \notin s$. This formulation is expressive enough to capture nonlinear interactions and complex substitution effects that are difficult to specify in closed form.

A.2.1 ARCHITECTURES

In our experiments, we mainly tested two architectures from Wang et al. (2023a). The outside option is handled by a separate logit that is always available.

Gated-Assort-Net (GAsN). GAsN is a feed-forward network that maps the assortment vector to product logits, followed by a masking step that enforces zero probability for unavailable products. Let $\mathbf{z}_0 = s$. For hidden layers $l = 1, \dots, L$, we use

$$\mathbf{z}_l = (\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l)^+,$$

where $(\cdot)^+ = \max(\cdot, 0)$ is applied elementwise. At the output layer, the product scores are gated by the assortment mask through a coordinatewise multiplication, so products not contained in s are removed before the final normalization. The resulting masked product scores are then combined with an outside-option logit and passed through a softmax over $[N] \cup \{0\}$. Therefore, products outside the offered assortment automatically receive probability zero.

Residual-Assort-Net (RAsN). RAsN uses the same masked output layer as GAsN, but replaces the hidden transformation by residual blocks:

$$\mathbf{z}_l = (\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l)^+ + \mathbf{z}_{l-1}.$$

The skip connection helps preserve information from earlier layers and stabilizes training. As in GAsN, the output layer applies a coordinatewise multiplication with the assortment mask before normalization, which retains the support constraint $p_\theta(i \mid s) = 0$ for unavailable items.

Discrete Diffusion Model. The diffusion model follows the discrete denoising formulation introduced in Austin et al. (2021) and detailed in Section 4, specialized to the binary state space $\{0, 1\}^N$. In our setting, the forward kernel factorizes across coordinates as

$$q_t(s_t \mid s_{t-1}) = \prod_{i=1}^{N} \left[(1 - \beta_t) \mathbf{1}\{s_{t,i} = s_{t-1,i}\} + \frac{\beta_t}{2} \right],$$

while the reverse model is parameterized as

$$p_\phi(s_{t-1} \mid s_t, t) = \prod_{i=1}^{N} \mathrm{Bernoulli}\Big(\sigma\big(g_\phi^{(i)}(s_t, t)\big) \Big).$$

This is the baseline binary diffusion structure inherited from Austin et al. (2021); our modification is to add the reward-guidance term to the reverse logits during sampling, as described in Section 4.

B ADDITIONAL NUMERICAL EXPERIMENTS AND RESULTS

This appendix supports and extends the main numerical results in §5; an overview of its structure is provided in Table 1. We first provide implementation details, hyperparameters, and runtime information. We then report the full optimal-ratio statistics underlying Figure 2, which quantify the overall quality of the generated candidate distribution. Beyond reward quality, we further analyze exact recovery of the global optimum and diversity among generated assortments. Exact recovery provides a strict measure of whether the sampler identifies the true optimum, while diversity characterizes the breadth of the generated candidate set, especially in larger problems where many near-optimal assortments may coexist. Finally, we evaluate sample efficiency and behavior-policy distribution shift to test the robustness of D3AO under different data regimes.

B.1 IMPLEMENTATION DETAILS

We use GAsN with a single hidden layer of width N and RAsN with a single residual block of width N, where N is the number of products. The diffusion model uses a two-layer MLP with hidden sizes $[128, 128]$ and $T = 100$ diffusion steps. The forward corruption schedule is linear, with β_t ranging from 10^{-4} to 0.1. For guided sampling, we use the schedule equation 4 with $\lambda_{\max} = 1000$ and $\gamma = 3$.

Table 1: Overview of the additional numerical experiments in Appendix §B.

Subsection	Description
B.1 Implementation Details and Runtime	Specifies model architectures, training hyperparameters, guidance settings, hardware, and representative runtimes.
B.2 Detailed Optimal-Ratio Results	Reports the full optimal-ratio statistics underlying Figure 2, including results across choice models, problem sizes, behavior policies, and methods.
B.3 Exact Recovery and Diversity	Evaluates exact recovery of the global optimum and diversity among generated assortments, complementing the optimal-ratio results.
B.4 Sample Efficiency	Studies how performance changes as the number of offline samples varies under MNL and MMNL data.
B.5 Distribution Shift	Tests robustness when historical assortments are collected from a shifted sampling policy while the underlying choice model is fixed.

All neural models, including GAsN, RAsN, and the diffusion model, are trained for 50 epochs with batch size 128, learning rate 10^{-3}, weight decay 0, and validation fraction 0.2. These hyperparameters are fixed across all experiments and are not tuned separately for individual settings.

All experiments are run on a machine equipped with NVIDIA GeForce RTX 3090 GPUs with 24GB memory. We report representative runtimes for the largest setting $N = 100$; smaller problem sizes are faster. Training each neural choice model takes approximately 1 minute, and training the diffusion model also takes approximately 1 minute. Guided sampling with batch size 256 and $T = 100$ reverse steps takes approximately 0.5 seconds per instance. The MNL-MLE baseline is very fast, taking approximately 0.5 seconds per instance. The most time-consuming baseline is MCCM-EM, for which a single instance can take up to approximately 1 minute.

B.2 Detailed Optimal-Ratio Results

In this subsection, we report the detailed numerical results for the optimal-ratio experiment summarized in the main text Section 5. We compare D3AO with classical parametric baselines on synthetic offline assortment optimization problems generated from three standard choice models: MNL, MCCM, and MMNL. These settings allow us to examine both performance under correct specification and robustness under model misspecification.

Setting. To evaluate different methods under controlled conditions, we generate synthetic datasets from three widely used choice models with increasing levels of complexity:

- **Multinomial Logit (MNL):** assumes homogeneous customer preferences and follows a standard softmax-based choice rule.
- **Markov Chain Choice Model (MCCM):** captures substitution effects through pairwise interactions among products.
- **Mixed Multinomial Logit (MMNL):** accounts for customer heterogeneity by introducing latent customer types.

See Appendix §A.1 for details on choice models. We vary the number of products as $N \in \{20, 40, 60, 80, 100\}$ and sample historical assortments from a Boltzmann policy with inverse temperature $\beta \in \{0.1, 1.0\}$. Our evaluation protocol is specified in §5.

Results on MNL Data. Table 2 reports numerical results on MNL data, where MNL-MLE serves as the correctly specified oracle baseline. Its near-perfect performance is expected, as the MNL model only requires estimating N item utilities, with $N \ll n$. In our setting with $n = 10{,}000$ offline samples, these parameters can be estimated very accurately. In contrast, unguided sampling performs much worse, and both its maximum and average optimal ratios decrease as N grows.

Once reward guidance is introduced, both GAsN-guided and RAsN-guided improve sharply and remain near-optimal across almost all settings, with maximum optimal ratios consistently close to 100%. Their mean, median, and 90th percentile statistics are also uniformly high, indicating that the improvement is not limited to a few best samples. Overall, guided diffusion remains highly effective even against a correctly specified model-based baseline.

Results on MCCM Data. Table 3 reports numerical results on synthetic data generated from the MCCM, where MCCM-EM is the correctly specified oracle parametric baseline. As in the MNL setting, MCCM-EM achieves a high optimal ratio across all problem sizes because it matches the true data-generating model, although its performance is slightly less perfect than in the MNL case due to the greater complexity of the MCCM. Unguided sampling again performs much worse, with both the maximum and average optimal ratios decreasing as N increases. In contrast, both guided variants improve dramatically and remain close to optimal across nearly all settings. Notably, in several cases, the best found solutions slightly exceed the single solution returned by MCCM-EM. Overall, these results show that guided diffusion remains highly effective in the more structured MCCM setting.

Results on MMNL Data under Model Misspecification. Table 4 reports numerical results on MMNL data, where both MNL-MLE and MCCM-EM are misspecified parametric baselines. In this setting, the model-based baselines degrade substantially and become much less stable across problem sizes, while unguided sampling also performs poorly. In contrast, both guided variants remain consistently strong and substantially outperform the misspecified baselines across all settings. This is consistent with the fact that our method does not rely on a fixed parametric assumption on the underlying choice model, and is therefore less sensitive to model mismatch. Overall, these results show that guided diffusion is substantially more robust to model misspecification and can recover high-quality assortments without requiring explicit assumptions on the data-generating model.

B.3 Exact Recovery and Diversity

While the optimal-ratio results show that guided diffusion consistently generates high-reward assortments, they do not reveal whether these samples exactly recover the global optimum, nor how the sampler behaves when exact recovery becomes difficult. In relatively simple settings, a strong method should place substantial probability mass on the true optimal assortment and recover it frequently. As the problem becomes more challenging, however, exact recovery may no longer be realistic, even when the generated assortments remain near-optimal in reward. In that regime, it becomes important to ask a complementary question: whether the generative procedure still produces a sufficiently diverse set of competitive candidate solutions, rather than collapsing to a small number of repeated outputs. Motivated by this, we next evaluate guided sampling from two perspectives beyond reward quality: exact recovery of the optimal assortment and diversity among generated assortments.

Exact Recovery of Optimal Assortments. While the optimal-ratio results quantify the reward quality of generated assortments, they do not reveal whether guided sampling actually recovers the true global optimum. To address this, we evaluate the *exact recovery rate* among generated samples. For each problem instance, suppose a method generates $M = 256$ assortments $s^{(1)}, \dots, s^{(M)}$, and let $s^\star$ denote an optimal assortment under the ground-truth model. We define

$$\mathrm{Rec}(S) = \frac{1}{M} \sum_{i=1}^{M} \mathbf{1}\{s^{(i)} = s^\star\},$$

that is, the proportion of generated samples that exactly match the optimal solution. Unlike reward-based metrics, exact recovery provides a stricter criterion that directly measures whether the generative procedure places sufficient probability mass on the true optimum.

Results on Exact Recovery. Table 5 reports the exact recovery rates under different data-generating models and problem sizes. In both the MNL and MCCM settings, the recovery rate decreases markedly as N increases, indicating that exact identification of the global optimum becomes increasingly difficult as the combinatorial search space grows. Nevertheless, both guided variants retain nontrivial recovery probabilities in low- and medium-dimensional regimes. For example, on MCCM data with $\beta = 1.0$, GAsN-guided achieves exact recovery rates of 68.16% at $N = 20$ and 32.73% at

Table 2: Optimal ratio statistics of generated assortments on MNL data. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. The ground-truth choice model is MNL, and MNL-MLE is the correctly specified parametric baseline. For this baseline, we report the quality of the single solution returned by the method; for GAsN and RAsN-guided methods, we report summary statistics over generated 256 assortments. * marks the oracle parametric model (i.e., the model class matches the ground-truth choice model).

(a) $\beta = 1.0$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MNL-MLE*	exact sol.	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	99.99 ± 0.01
Unguided	max	89.95 ± 6.12	81.14 ± 7.54	76.67 ± 4.82	72.24 ± 3.35	68.95 ± 2.72
	mean	64.73 ± 6.34	61.43 ± 8.16	60.12 ± 4.26	57.60 ± 5.78	55.95 ± 2.83
	q50	65.11 ± 6.71	62.26 ± 8.59	60.77 ± 4.83	57.75 ± 6.08	55.87 ± 2.91
	q90	76.71 ± 5.43	72.25 ± 8.22	68.01 ± 4.87	64.02 ± 5.84	61.76 ± 2.81
GAsN-guided	max	**99.93** ± 0.12	**99.96** ± 0.13	**99.78** ± 0.16	97.62 ± 6.07	99.56 ± 0.28
	mean	99.17 ± 1.79	**99.65** ± 0.41	**99.17** ± 0.44	96.37 ± 7.75	**98.82** ± 0.61
	q50	99.11 ± 1.98	**99.63** ± 0.46	99.23 ± 0.44	96.42 ± 7.52	98.85 ± 0.66
	q90	99.33 ± 1.56	**99.85** ± 0.27	99.50 ± 0.34	97.14 ± 6.93	99.24 ± 0.45
RAsN-guided	max	99.80 ± 0.55	99.83 ± 0.35	99.59 ± 1.09	**99.91** ± 0.14	**99.80** ± 0.28
	mean	**99.56** ± 0.55	99.01 ± 0.97	98.96 ± 2.25	**99.36** ± 0.41	98.78 ± 0.65
	q50	**99.62** ± 0.58	99.06 ± 0.96	**99.39** ± 1.04	**99.39** ± 0.40	**98.87** ± 0.77
	q90	**99.72** ± 0.55	99.53 ± 0.73	**99.54** ± 1.07	**99.73** ± 0.25	**99.43** ± 0.50

(b) $\beta = 0.1$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MNL-MLE*	exact sol.	100.00 ± 0.00	99.99 ± 0.02	100.00 ± 0.00	100.00 ± 0.00	99.99 ± 0.01
Unguided	max	90.45 ± 5.54	80.89 ± 7.65	76.40 ± 4.96	71.35 ± 5.95	68.80 ± 3.99
	mean	63.72 ± 6.73	60.43 ± 7.78	59.79 ± 4.27	57.17 ± 5.79	55.47 ± 2.93
	q50	64.28 ± 7.18	60.26 ± 7.95	60.30 ± 4.54	57.33 ± 5.91	55.42 ± 2.95
	q90	75.57 ± 6.07	71.88 ± 7.71	68.31 ± 4.92	63.84 ± 5.55	61.39 ± 2.82
GAsN-guided	max	**99.93** ± 0.12	**99.93** ± 0.14	99.74 ± 0.04	99.86 ± 0.16	99.89 ± 0.18
	mean	**99.72** ± 0.39	**99.61** ± 0.51	99.26 ± 0.42	98.93 ± 0.61	98.77 ± 0.67
	q50	**99.77** ± 0.33	**99.63** ± 0.53	99.27 ± 0.46	98.93 ± 0.65	98.80 ± 0.73
	q90	**99.93** ± 0.12	**99.83** ± 0.34	99.57 ± 0.32	99.60 ± 0.55	99.62 ± 0.33
RAsN-guided	max	99.65 ± 0.81	99.54 ± 0.68	**99.85** ± 0.28	**99.98** ± 0.03	**99.98** ± 0.03
	mean	98.97 ± 1.05	99.18 ± 0.75	**99.39** ± 0.37	**99.60** ± 0.16	**99.54** ± 0.25
	q50	99.05 ± 1.07	99.14 ± 0.78	**99.42** ± 0.47	**99.67** ± 0.23	**99.56** ± 0.31
	q90	99.40 ± 0.69	99.40 ± 0.69	**99.71** ± 0.43	**99.93** ± 0.07	**99.84** ± 0.15

$N = 40$. In more challenging settings, especially under MMNL model misspecification, exact recovery becomes rare or disappears altogether. Combined with the high optimal-ratio results in Tables 2 and 3, these findings suggest that even when exact recovery is difficult, guided diffusion continues to generate high-quality near-optimal assortments reliably.

Diversity of Generated Assortments. As shown above, exact recovery becomes increasingly difficult as the problem size grows, even when guided diffusion continues to generate near-optimal assortments. In such regimes, it is important to further understand whether the sampler still explores a nontrivial set of competitive solutions, rather than collapsing to a small number of repeated outputs. To quantify this behavior, we evaluate the diversity of generated assortments using two complementary metrics: the *unique sample ratio* and the *average pairwise Hamming distance*. Suppose each method generates M assortments $s^{(1)}, \ldots, s^{(M)}$, where each $s^{(i)} \in 0, 1^N$. Let U denote the number of distinct assortments in this set. We define the *unique sample ratio* as

$$\mathrm{Uniq}(S) = \frac{U}{M},$$

Table 3: Optimal ratio statistics of generated assortments on MCCM data. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. The ground-truth choice model is MCCM, and MCCM-EM is the correctly specified parametric baseline. For this baseline, we report the quality of the single solution returned by the method; for GAsN and RAsN-guided methods, we report summary statistics over generated 256 assortments. * marks the oracle parametric model (i.e., the model class matches the ground-truth choice model).

(a) $\beta = 1.0$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MCCM-EM*	exact sol.	99.34 ± 1.83	99.36 ± 0.71	99.50 ± 0.71	97.54 ± 6.74	99.89 ± 0.09
Unguided	max	86.25 ± 4.90	78.53 ± 6.38	72.91 ± 4.10	69.58 ± 3.31	67.00 ± 2.15
	mean	62.97 ± 7.88	62.24 ± 6.41	60.32 ± 2.88	58.85 ± 2.95	56.88 ± 2.40
	q50	63.22 ± 8.34	62.39 ± 6.43	60.24 ± 2.87	58.81 ± 2.89	56.94 ± 2.41
	q90	74.57 ± 7.11	70.26 ± 6.20	66.48 ± 3.06	63.87 ± 2.73	61.41 ± 2.30
GAsN-guided	max	99.50 ± 1.49	**99.92** ± 0.14	**99.97** ± 0.04	99.82 ± 0.32	99.91 ± 0.11
	mean	**99.42** ± 1.51	**99.55** ± 0.46	**99.67** ± 0.20	99.25 ± 0.48	99.29 ± 0.44
	q50	**99.44** ± 1.48	**99.54** ± 0.49	**99.70** ± 0.22	99.27 ± 0.52	99.37 ± 0.43
	q90	**99.48** ± 1.49	**99.78** ± 0.32	**99.88** ± 0.10	99.62 ± 0.41	99.63 ± 0.30
RAsN-guided	max	**99.65** ± 0.46	99.60 ± 0.67	99.87 ± 0.22	**99.99** ± 0.02	**99.99** ± 0.02
	mean	98.95 ± 1.03	99.17 ± 0.67	99.45 ± 0.35	**99.61** ± 0.18	**99.59** ± 0.19
	q50	98.97 ± 1.28	99.23 ± 0.60	99.51 ± 0.36	**99.68** ± 0.27	**99.66** ± 0.18
	q90	99.35 ± 0.77	99.47 ± 0.66	99.71 ± 0.43	**99.92** ± 0.10	**99.88** ± 0.09

(b) $\beta = 0.1$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MCCM-EM*	exact sol.	98.84 ± 2.17	99.44 ± 0.69	99.50 ± 0.71	97.53 ± 6.74	99.84 ± 0.18
Unguided	max	86.22 ± 5.42	78.12 ± 6.09	72.25 ± 3.87	69.71 ± 3.22	65.56 ± 2.40
	mean	63.20 ± 7.00	62.31 ± 6.51	59.83 ± 3.01	58.52 ± 2.95	56.47 ± 2.56
	q50	63.56 ± 7.28	62.49 ± 6.61	59.77 ± 3.06	58.50 ± 2.97	56.49 ± 2.60
	q90	74.59 ± 5.94	69.82 ± 6.31	65.90 ± 3.04	63.53 ± 2.90	60.73 ± 2.58
GAsN-guided	max	99.48 ± 1.49	**99.93** ± 0.14	**99.97** ± 0.04	99.86 ± 0.16	99.89 ± 0.18
	mean	**99.41** ± 1.48	**99.55** ± 0.51	**99.64** ± 0.22	99.20 ± 0.48	99.25 ± 0.46
	q50	**99.44** ± 1.48	**99.60** ± 0.53	**99.67** ± 0.26	99.23 ± 0.51	99.31 ± 0.47
	q90	**99.48** ± 1.49	**99.73** ± 0.34	**99.87** ± 0.12	99.60 ± 0.37	99.62 ± 0.33
RAsN-guided	max	**99.65** ± 0.46	99.54 ± 0.68	99.85 ± 0.28	**99.98** ± 0.03	**99.98** ± 0.03
	mean	98.97 ± 1.05	99.18 ± 0.75	99.39 ± 0.37	**99.60** ± 0.16	**99.54** ± 0.25
	q50	99.05 ± 1.07	99.14 ± 0.78	99.42 ± 0.47	**99.67** ± 0.23	**99.56** ± 0.31
	q90	99.40 ± 0.69	99.40 ± 0.69	99.71 ± 0.43	**99.93** ± 0.07	**99.84** ± 0.15

which measures the proportion of distinct solutions among all generated samples. A higher value indicates that the generative procedure avoids collapsing to only a few repeated assortments.

To further measure structural variation among generated solutions, we consider the *average pairwise Hamming distance*:

$$\mathrm{Ham}(S) = \frac{2}{M(M-1)} \sum_{1 \le i < j \le M} \frac{1}{N} \left\| s^{(i)} - s^{(j)} \right\|_1.$$

This metric measures how different two generated assortments are in terms of product inclusion, normalized by the number of products. While the unique ratio reflects *support diversity*, namely how many distinct solutions are produced, the Hamming distance reflects *geometric diversity*, namely how far apart these solutions are in the combinatorial space. Together, these two metrics provide a more complete picture of whether guided sampling maintains meaningful variation among high-quality candidate assortments.

Table 4: Optimal ratio statistics of generated assortments on MMNL data. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. The ground-truth choice model is MMNL, while MNL-MLE and MCCM-EM are misspecified parametric baselines. For these baselines, we report the quality of the single solution returned by the method; for GAsN and RAsN-guided methods, we report summary statistics over generated 256 assortments.

(a) $\beta = 1.0$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MNL-MLE	exact sol.	83.64 ± 12.57	54.10 ± 11.80	63.22 ± 12.77	81.92 ± 14.78	91.09 ± 5.44
MCCM-EM	exact sol.	91.94 ± 9.64	58.91 ± 16.22	55.65 ± 11.30	49.32 ± 11.86	59.03 ± 17.56
Unguided	max	86.52 ± 4.06	75.54 ± 4.15	73.52 ± 3.03	66.47 ± 3.56	65.37 ± 3.16
	mean	60.72 ± 6.15	55.66 ± 5.66	56.32 ± 3.31	53.21 ± 2.53	54.37 ± 3.20
	q50	60.85 ± 6.62	55.50 ± 5.92	56.44 ± 3.41	53.31 ± 2.58	54.46 ± 3.26
	q90	72.27 ± 6.47	64.92 ± 5.80	64.02 ± 3.57	59.24 ± 2.76	60.41 ± 3.44
GAsN-guided	max	**95.78** ± 5.45	**98.58** ± 1.43	**98.77** ± 0.61	97.23 ± 1.45	97.39 ± 0.66
	mean	**92.27** ± 8.10	**97.26** ± 1.76	**97.85** ± 0.81	92.96 ± 8.51	93.31 ± 5.99
	q50	**92.08** ± 8.26	**97.27** ± 1.70	**97.76** ± 0.94	93.09 ± 8.59	93.64 ± 6.04
	q90	**92.30** ± 8.44	**97.89** ± 1.61	**98.31** ± 0.70	93.64 ± 8.58	96.43 ± 1.26
RAsN-guided	max	92.54 ± 9.98	94.71 ± 6.17	98.72 ± 1.14	**98.33** ± 1.04	**98.31** ± 0.79
	mean	89.99 ± 9.99	87.48 ± 9.70	96.74 ± 2.58	**97.26** ± 1.53	**96.36** ± 2.08
	q50	90.23 ± 9.49	86.98 ± 9.60	97.10 ± 1.62	**97.44** ± 1.65	**96.33** ± 2.17
	q90	91.18 ± 10.02	89.43 ± 10.82	98.02 ± 1.75	**97.94** ± 1.44	**97.50** ± 1.48

(b) $\beta = 0.1$

Method	Statistic	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
MNL-MLE	exact sol.	84.10 ± 12.77	56.15 ± 13.67	65.72 ± 14.42	79.80 ± 17.58	89.62 ± 8.48
MCCM-EM	exact sol.	91.66 ± 8.46	56.24 ± 16.40	55.65 ± 11.30	49.32 ± 11.86	59.03 ± 17.56
Unguided	max	89.25 ± 4.62	75.62 ± 4.33	72.15 ± 3.25	66.68 ± 3.00	65.77 ± 3.11
	mean	61.75 ± 5.56	56.13 ± 5.58	56.78 ± 2.98	53.52 ± 2.63	53.17 ± 3.16
	q50	61.92 ± 5.89	56.02 ± 5.79	56.89 ± 3.08	53.58 ± 2.67	53.26 ± 3.19
	q90	73.79 ± 6.04	65.63 ± 5.94	63.73 ± 3.36	59.67 ± 2.93	59.02 ± 3.23
GAsN-guided	max	**95.78** ± 5.45	**98.81** ± 1.34	98.82 ± 0.56	97.52 ± 1.35	97.52 ± 0.69
	mean	**93.91** ± 5.67	**97.42** ± 1.52	95.38 ± 7.80	93.10 ± 7.97	95.75 ± 1.59
	q50	**93.48** ± 5.40	**97.32** ± 1.78	95.25 ± 8.25	93.17 ± 8.33	95.73 ± 1.65
	q90	**95.78** ± 5.45	**98.03** ± 1.45	96.13 ± 6.85	93.76 ± 8.20	96.43 ± 1.32
RAsN-guided	max	93.32 ± 10.20	94.18 ± 7.44	**98.77** ± 1.15	**98.36** ± 1.02	**98.25** ± 0.83
	mean	90.52 ± 9.35	88.24 ± 9.64	**97.35** ± 1.14	**97.20** ± 1.52	**96.37** ± 1.95
	q50	90.34 ± 9.58	87.86 ± 9.64	**97.19** ± 1.48	**97.28** ± 1.64	**96.42** ± 2.05
	q90	92.54 ± 9.98	91.05 ± 11.03	**98.09** ± 1.46	**97.93** ± 1.46	**97.43** ± 1.38

Results on Diversity. Table 6 reports the diversity of generated assortments across different data-generating models and problem sizes. Overall, both the unique sample ratio and the average pairwise Hamming distance tend to increase as N grows, indicating that guided sampling becomes more diverse in larger combinatorial spaces. By contrast, when N is small, the generated assortments are much less diverse. This is consistent with the exact-recovery results: in simpler regimes, the optimal assortment is easier to identify, so the sampling process tends to concentrate more strongly around the optimum. As the problem becomes harder, exact recovery becomes less likely, and the sampler correspondingly spreads its probability mass over a broader set of competitive candidate solutions rather than repeatedly producing the same assortment.

Table 5: Proportion of exact optimal assortments. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs.

(a) MNL data

β	Method	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
1.0	GAsN-guided	**45.20** ± 44.34	**30.27** ± 43.19	**12.42** ± 26.33	0.70 ± 1.41	0.43 ± 1.29
	RAsN-guided	30.23 ± 34.04	31.02 ± 35.48	8.44 ± 13.29	**7.46** ± 16.14	**0.55** ± 1.40
0.1	GAsN-guided	**44.38** ± 44.13	**31.17** ± 39.82	**13.09** ± 27.40	1.25 ± 3.26	0.20 ± 0.59
	RAsN-guided	34.45 ± 36.26	28.63 ± 39.36	13.05 ± 20.52	**7.73** ± 15.96	**0.98** ± 2.18

(b) MCCM data

β	Method	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
1.0	GAsN-guided	**68.16** ± 44.35	**32.73** ± 33.71	**5.98** ± 8.93	2.97 ± 7.91	0.98 ± 1.77
	RAsN-guided	36.45 ± 44.19	26.21 ± 37.13	2.73 ± 3.67	**7.15** ± 10.42	**7.89** ± 17.11
0.1	GAsN-guided	**67.77** ± 44.59	**35.82** ± 38.43	**6.41** ± 13.77	2.77 ± 7.01	1.13 ± 1.76
	RAsN-guided	36.76 ± 44.77	27.23 ± 36.83	2.97 ± 4.76	**6.64** ± 13.34	**6.60** ± 15.46

(c) MMNL data

β	Method	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
1.0	GAsN-guided	13.48 ± 30.65	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	RAsN-guided	**20.16** ± 37.58	0.00 ± 0.00	**3.24** ± 9.73	0.00 ± 0.00	**1.48** ± 4.45
0.1	GAsN-guided	6.29 ± 14.52	**0.08** ± 0.23	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	RAsN-guided	**20.12** ± 36.98	0.00 ± 0.00	**3.13** ± 9.38	0.00 ± 0.00	**0.63** ± 1.88

B.4 Sample Efficiency

Historical choice data may be limited in practice, and the reliability of offline assortment optimization can depend strongly on the amount of observed data. Motivated by this, We next evaluate how performance changes with the amount of available offline data. While the main experiments use a fixed dataset size, here we vary the number of offline samples to examine the sample efficiency of D3AO.

Sample-size setting. To evaluate sample efficiency, we vary the size of the offline dataset while keeping the underlying assortment optimization problem fixed. Specifically, for each setting, we generate historical data under the same choice model and compare the performance of different methods across a range of sample sizes, with

$$n \in \{100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000\}.$$

As in the main experiments, we consider both MNL and MMNL data, and report results for two problem sizes, $N = 40$ and $N = 80$. For each configuration, all results are averaged over 10 independent runs, and we report both the mean and variability across runs. For sample efficiency, performance is measured in terms of *Regret*, defined as

$$\text{Regret} = 1 - \text{Optimal Ratio},$$

so that lower values indicate solutions closer to the true optimum. Figure 3 summarizes how regret changes with the amount of available offline data for the diffusion-based methods and the parametric baseline.

Results on sample size. Figure 3, Table 7 and Table 8 show that the effect of sample size differs substantially across data-generating models. In the MNL setting, regret is uniformly low and generally decreases with n, while the correctly specified MNL-MLE baseline quickly approaches zero regret. In contrast, under MMNL data, the misspecified MNL-MLE baseline exhibits substantially larger

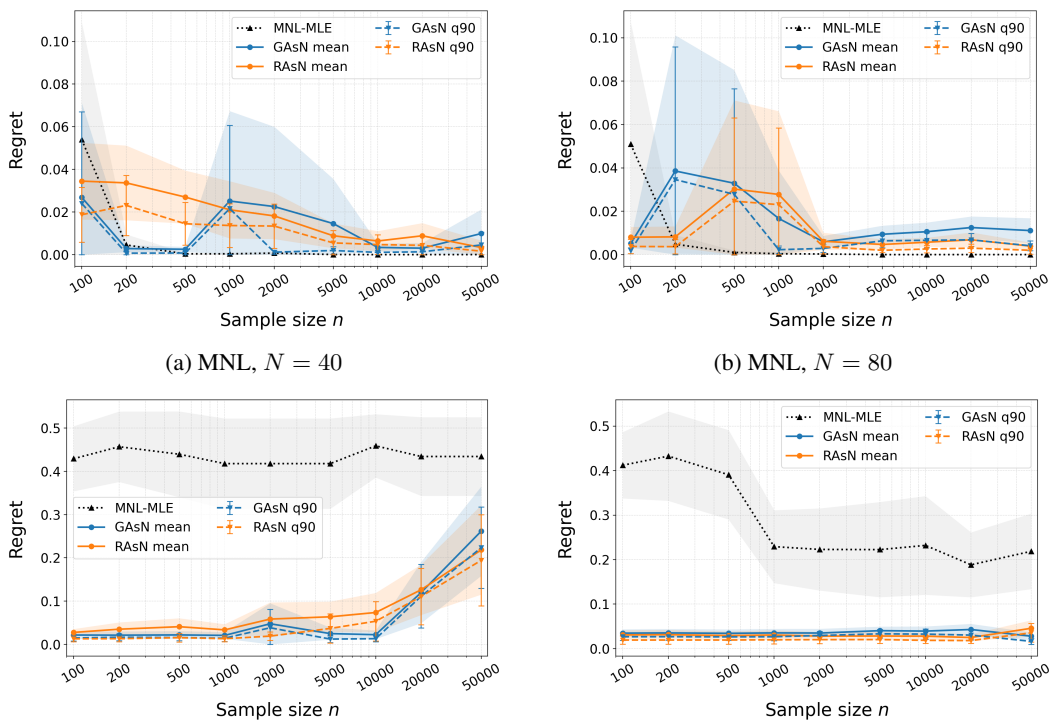
Table 6: Diversity of generated assortments measured by unique sample ratio and average pairwise Hamming distance. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs, based on 256 generated assortments per run.

(a) Unique ratio

β	Data model	Method	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
1.0	MNL	GAsN	**1.21** ± 1.07	**5.31** ± 4.67	**14.88** ± 11.78	25.00 ± 26.20	28.63 ± 13.82
		RAsN	**1.21** ± 0.56	3.71 ± 3.97	14.38 ± 12.72	**27.19** ± 23.93	**50.74** ± 28.23
	MCCM	GAsN	0.78 ± 0.55	**2.81** ± 1.59	9.73 ± 4.02	**17.73** ± 10.18	25.74 ± 14.73
		RAsN	**1.09** ± 0.67	2.15 ± 1.12	**12.11** ± 16.51	13.44 ± 11.88	**27.42** ± 20.25
	MMNL	GAsN	**0.82** ± 0.41	1.29 ± 0.63	6.64 ± 9.54	**18.59** ± 16.13	**44.77** ± 27.14
		RAsN	**0.82** ± 0.32	**3.63** ± 4.21	**6.68** ± 9.03	16.64 ± 15.58	38.75 ± 27.39
0.1	MNL	GAsN	**1.09** ± 0.80	**5.27** ± 4.22	10.98 ± 5.97	16.17 ± 9.30	27.77 ± 13.38
		RAsN	0.98 ± 0.44	3.59 ± 3.99	**12.03** ± 11.26	**24.06** ± 21.44	**52.30** ± 27.45
	MCCM	GAsN	0.74 ± 0.37	**2.93** ± 1.65	9.38 ± 4.90	**19.30** ± 10.26	**29.38** ± 16.42
		RAsN	**0.90** ± 0.50	1.84 ± 1.05	**12.42** ± 18.19	13.75 ± 12.46	27.30 ± 19.85
	MMNL	GAsN	**0.86** ± 0.38	1.99 ± 1.00	**13.83** ± 22.48	**22.89** ± 23.31	36.13 ± 26.08
		RAsN	**0.86** ± 0.34	**3.12** ± 3.27	4.96 ± 7.67	15.66 ± 15.57	**40.31** ± 27.90

(b) Average pairwise Hamming distance

β	Data model	Method	$N = 20$	$N = 40$	$N = 60$	$N = 80$	$N = 100$
1.0	MNL	GAsN	**1.56** ± 1.66	**2.27** ± 1.49	2.73 ± 1.33	2.98 ± 2.55	2.44 ± 0.64
		RAsN	1.44 ± 0.93	1.88 ± 1.95	**2.83** ± 1.41	**3.10** ± 1.73	**4.28** ± 2.28
	MCCM	GAsN	0.50 ± 0.66	**1.51** ± 0.99	2.24 ± 0.80	**2.19** ± 0.93	2.13 ± 0.84
		RAsN	**1.50** ± 1.83	1.37 ± 0.99	**2.31** ± 1.84	1.93 ± 1.01	**2.25** ± 1.04
	MMNL	GAsN	0.62 ± 0.85	0.61 ± 0.70	1.53 ± 1.39	2.22 ± 1.21	**5.65** ± 5.84
		RAsN	**1.42** ± 2.11	**3.10** ± 4.28	**2.28** ± 3.47	**2.40** ± 1.22	3.59 ± 2.05
0.1	MNL	GAsN	1.21 ± 1.59	**2.19** ± 1.26	**2.25** ± 0.76	2.14 ± 0.86	2.36 ± 0.63
		RAsN	**1.22** ± 1.09	1.51 ± 1.77	2.24 ± 1.31	**2.93** ± 1.63	**4.44** ± 2.29
	MCCM	GAsN	0.63 ± 0.81	**1.34** ± 0.94	2.10 ± 0.92	**2.40** ± 0.87	**2.30** ± 0.92
		RAsN	**1.29** ± 1.54	1.00 ± 0.81	**2.29** ± 1.79	1.98 ± 1.23	2.29 ± 1.07
	MMNL	GAsN	**1.25** ± 0.92	0.78 ± 0.79	**2.77** ± 4.29	**2.76** ± 2.29	2.76 ± 1.46
		RAsN	1.19 ± 2.16	**2.34** ± 2.90	1.39 ± 1.14	2.24 ± 1.10	**3.65** ± 2.05



(a) MNL, $N = 40$

(b) MNL, $N = 80$

(c) MMNL, $N = 40$

(d) MMNL, $N = 80$

Figure 3: Suboptimality (regret) versus sample size n under MNL and MMNL choice models. Diffusion-based methods (GAsN, RAsN) achieve consistently lower regret compared to the MNL-MLE baseline, especially under model misspecification (MMNL).

regret across all sample sizes. The diffusion-based methods remain markedly stronger in this setting, but their performance does not improve monotonically with more data. Overall, these results suggest that a main advantage of guided diffusion lies in its robustness under model misspecification across different data regimes, rather than in a uniformly monotonic improvement with increasing sample size.

Table 7: Optimal ratio statistics on MNL data across varying offline sample sizes. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. For MNL-MLE, we report the quality of the single returned solution; for GAsN-guided and RAsN-guided methods, we report summary statistics over 256 generated assortments.

N	Method	Statistic	$n = 100$	$n = 200$	$n = 500$	$n = 1000$	$n = 2000$	$n = 5000$	$n = 10000$	$n = 20000$	$n = 50000$
40	MNL-MLE	exact sol.	94.60 ± 8.87	99.55 ± 0.79	99.96 ± 0.08	99.96 ± 0.08	99.93 ± 0.10	100.00 ± 0.01	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.01
	GAsN	max	99.98 ± 0.05	99.99 ± 0.04	100.00 ± 0.01	98.18 ± 5.40	99.98 ± 0.05	99.92 ± 0.15	99.96 ± 0.10	99.97 ± 0.09	99.85 ± 0.37
		mean	97.31 ± 7.16	99.72 ± 0.26	99.75 ± 0.19	97.49 ± 6.81	97.75 ± 6.04	98.54 ± 3.39	99.66 ± 0.30	99.70 ± 0.39	99.01 ± 1.82
		q50	97.24 ± 7.56	99.75 ± 0.35	99.80 ± 0.21	97.50 ± 6.88	97.52 ± 6.91	98.29 ± 4.27	99.69 ± 0.29	99.71 ± 0.41	98.93 ± 2.18
		q90	97.60 ± 6.94	99.93 ± 0.11	99.93 ± 0.12	97.84 ± 6.31	99.88 ± 0.15	99.81 ± 0.29	99.90 ± 0.17	99.87 ± 0.26	99.55 ± 0.96
	RAsN	max	99.29 ± 0.85	99.29 ± 0.83	99.72 ± 0.62	99.65 ± 0.60	99.72 ± 0.62	99.72 ± 0.62	99.74 ± 0.62	99.67 ± 0.56	99.87 ± 0.32
		mean	96.55 ± 2.91	96.63 ± 2.82	97.30 ± 2.01	97.90 ± 2.16	98.18 ± 1.75	99.11 ± 0.94	99.36 ± 0.75	99.12 ± 0.96	99.67 ± 0.53
		q50	96.91 ± 2.51	96.73 ± 2.75	97.00 ± 2.35	98.41 ± 1.57	98.28 ± 1.97	99.10 ± 1.00	99.46 ± 0.71	99.31 ± 0.81	99.72 ± 0.61
		q90	98.13 ± 2.07	97.69 ± 2.28	98.55 ± 1.62	98.64 ± 1.66	98.66 ± 1.67	99.45 ± 0.93	99.53 ± 0.74	99.56 ± 0.73	99.84 ± 0.35
80	MNL-MLE	exact sol.	94.91 ± 9.07	99.52 ± 0.68	99.90 ± 0.14	99.96 ± 0.11	99.97 ± 0.08	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	GAsN	max	99.97 ± 0.04	99.94 ± 0.11	97.61 ± 6.98	99.90 ± 0.14	99.87 ± 0.17	99.69 ± 0.30	99.64 ± 0.35	99.67 ± 0.32	99.84 ± 0.24
		mean	99.49 ± 0.30	96.14 ± 10.10	96.71 ± 8.43	98.33 ± 3.56	99.41 ± 0.44	99.06 ± 0.64	98.94 ± 0.68	98.75 ± 0.83	98.89 ± 0.92
		q50	99.56 ± 0.33	96.10 ± 10.42	96.78 ± 8.46	99.45 ± 0.48	99.48 ± 0.46	99.08 ± 0.72	99.00 ± 0.77	98.81 ± 0.79	99.05 ± 0.71
		q90	99.82 ± 0.22	96.54 ± 9.88	97.22 ± 7.86	99.77 ± 0.29	99.71 ± 0.29	99.36 ± 0.51	99.34 ± 0.55	99.32 ± 0.48	99.59 ± 0.37
	RAsN	max	99.89 ± 0.17	99.86 ± 0.17	99.86 ± 0.17	99.81 ± 0.21	99.87 ± 0.17	99.92 ± 0.13	99.88 ± 0.20	99.90 ± 0.16	99.97 ± 0.07
		mean	99.20 ± 0.76	99.18 ± 0.78	96.98 ± 6.60	97.22 ± 6.19	99.39 ± 0.64	99.53 ± 0.38	99.43 ± 0.34	99.32 ± 0.55	99.61 ± 0.27
		q50	99.35 ± 0.82	99.34 ± 0.82	97.02 ± 6.80	97.22 ± 6.26	99.34 ± 0.82	99.58 ± 0.44	99.47 ± 0.33	99.40 ± 0.47	99.68 ± 0.29
		q90	99.62 ± 0.53	99.63 ± 0.53	97.53 ± 6.19	97.70 ± 5.71	99.63 ± 0.53	99.80 ± 0.16	99.74 ± 0.27	99.70 ± 0.31	99.80 ± 0.29

Table 8: Optimal ratio statistics on MMNL data across varying offline sample sizes. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. For MNL-MLE, we report the quality of the single returned solution; for GAsN-guided and RAsN-guided methods, we report summary statistics over 256 generated assortments.

N	Method	Statistic	$n = 100$	$n = 200$	$n = 500$	$n = 1000$	$n = 2000$	$n = 5000$	$n = 10000$	$n = 20000$	$n = 50000$
40	MNL-MLE	exact sol.	57.07 ± 12.12	54.30 ± 13.13	56.04 ± 15.97	58.20 ± 16.83	58.20 ± 16.83	58.20 ± 16.83	54.10 ± 11.80	56.56 ± 14.67	56.56 ± 14.67
	GAsN	max	99.12 ± 0.52	99.19 ± 0.53	99.26 ± 0.55	99.26 ± 0.55	99.13 ± 0.92	99.12 ± 0.91	99.13 ± 1.09	95.74 ± 4.41	83.28 ± 11.67
		mean	97.84 ± 1.41	97.88 ± 1.42	97.82 ± 1.47	97.93 ± 1.22	95.24 ± 7.65	97.51 ± 2.25	97.77 ± 1.58	88.00 ± 11.42	73.81 ± 16.65
		q50	97.76 ± 1.58	97.94 ± 1.70	97.74 ± 1.68	97.81 ± 1.27	95.61 ± 6.93	98.14 ± 1.73	98.29 ± 1.32	88.04 ± 11.90	73.62 ± 18.33
		q90	98.52 ± 1.16	98.36 ± 1.32	98.43 ± 1.20	98.58 ± 1.25	96.16 ± 6.88	98.78 ± 1.29	98.69 ± 1.18	88.87 ± 11.85	77.63 ± 15.16
	RAsN	max	99.28 ± 1.03	99.02 ± 0.99	99.06 ± 1.02	99.06 ± 1.02	99.06 ± 1.02	96.41 ± 5.51	96.09 ± 7.10	90.61 ± 10.79	82.75 ± 18.14
		mean	97.23 ± 1.28	96.50 ± 2.55	95.92 ± 3.16	96.65 ± 2.13	94.15 ± 6.21	93.63 ± 5.68	92.63 ± 7.31	87.37 ± 9.40	78.15 ± 16.66
		q50	97.56 ± 1.58	96.77 ± 2.29	95.91 ± 3.67	96.88 ± 2.25	94.43 ± 6.09	94.49 ± 6.36	93.18 ± 6.93	87.48 ± 9.69	77.46 ± 16.91
		q90	98.71 ± 1.14	98.69 ± 1.17	98.45 ± 1.57	98.69 ± 1.17	98.10 ± 1.57	96.31 ± 5.46	94.65 ± 7.35	88.93 ± 10.51	80.56 ± 17.02
80	MNL-MLE	exact sol.	58.80 ± 12.01	56.75 ± 16.20	60.94 ± 16.17	77.09 ± 13.20	77.72 ± 14.88	77.76 ± 17.25	76.81 ± 17.88	81.17 ± 11.74	78.14 ± 13.68
	GAsN	max	98.35 ± 0.92	98.35 ± 0.98	98.24 ± 1.06	98.35 ± 0.91	97.96 ± 1.19	97.39 ± 1.35	97.66 ± 1.43	97.78 ± 1.42	98.80 ± 1.19
		mean	96.53 ± 1.42	96.48 ± 1.45	96.57 ± 1.46	96.47 ± 1.49	96.50 ± 1.55	95.97 ± 1.49	96.07 ± 1.69	95.71 ± 2.05	97.21 ± 1.39
		q50	96.45 ± 1.56	96.40 ± 1.63	96.51 ± 1.56	96.46 ± 1.59	96.50 ± 1.69	95.93 ± 1.52	96.18 ± 1.81	95.88 ± 1.77	97.15 ± 1.87
		q90	97.34 ± 1.33	97.33 ± 1.28	97.37 ± 1.29	97.34 ± 1.46	97.05 ± 1.47	96.68 ± 1.53	96.75 ± 1.85	96.97 ± 1.75	98.37 ± 1.09
	RAsN	max	99.08 ± 1.20	98.76 ± 1.49	99.08 ± 1.20	98.79 ± 1.44	98.85 ± 1.11	98.35 ± 1.10	98.74 ± 0.78	98.92 ± 1.17	97.26 ± 2.78
		mean	96.93 ± 1.75	96.86 ± 1.73	97.07 ± 1.57	96.94 ± 1.72	97.19 ± 1.47	97.17 ± 1.48	97.21 ± 1.24	97.51 ± 1.24	95.43 ± 2.91
		q50	97.38 ± 1.30	97.29 ± 1.57	97.42 ± 1.29	97.22 ± 1.68	97.38 ± 1.37	97.03 ± 1.70	97.03 ± 1.44	97.62 ± 1.45	95.60 ± 2.95
		q90	98.08 ± 1.50	98.08 ± 1.49	98.07 ± 1.47	98.03 ± 1.48	97.99 ± 1.46	97.97 ± 1.35	98.14 ± 1.11	98.20 ± 1.04	96.45 ± 3.37

B.5 DISTRIBUTION SHIFT

To further evaluate robustness beyond the data-collection setting used in the main experiments, we consider a distribution-shift setting in which the historical assortments are generated from a different sampling policy. This experiment is motivated by the fact that offline datasets may not be collected under the same behavior policy across applications. A method that relies too strongly on the observed assortment distribution may therefore be sensitive to how the offline data were collected, even when the underlying choice behavior remains unchanged.

Distribution-shift setting. Instead of sampling historical assortments from the Boltzmann policy, we use a `uniform_size_then_subset` procedure. For each sample, we first draw the assortment size k uniformly from $\{1, \ldots, N\}$, and then uniformly sample one subset of size k from the combinatorial space. This produces an assortment distribution that is not Boltzmann. Given each sampled assortment, the observed choice is generated according to the selected ground-truth choice model. Under MNL, choices are sampled from the standard softmax distribution over the offered products and the outside option. Under MMNL, choices are sampled from a mixture of type-specific MNL models with uniformly weighted latent customer types. Thus, the underlying choice mechanism is kept unchanged, while the distribution of observed assortments is shifted.

Results under distribution shift. Table 9 reports the optimal-ratio statistics under this shifted assortment-generation distribution. Compared with the main experiments under the Boltzmann distribution, performance drops noticeably in both the MNL and MMNL settings, especially in the mean, median, and upper-quantile statistics. Nevertheless, the maximum optimal ratios remain relatively high, indicating that guided diffusion can still recover highly competitive solutions under distribution shift. Overall, these results suggest that although the proposed approach is affected by changes in the assortment distribution, it remains reasonably robust, particularly in preserving strong best-case solution quality.

Table 9: Optimal ratio statistics of generated assortments under a distribution-shift setting. Historical assortments are generated from a policy different from that used in the main experiments, while observed choices are still sampled from the ground-truth choice model. All entries are reported in percentage units (%) and shown as mean ± standard deviation over 10 runs. For each diffusion-based method, we report summary statistics over 256 generated assortments per run.

Method	Statistic	MNL		MMNL	
		$N = 40$	$N = 80$	$N = 40$	$N = 80$
Unguided	max	80.06 ± 6.09	88.91 ± 3.73	65.47 ± 5.71	69.68 ± 7.34
	mean	57.70 ± 0.98	55.68 ± 0.39	38.76 ± 0.91	46.57 ± 0.52
	q50	63.24 ± 0.07	55.96 ± 0.14	46.10 ± 0.18	49.13 ± 0.08
	q90	64.43 ± 0.18	68.21 ± 1.25	48.92 ± 0.98	51.10 ± 0.18
GAsN-guided	max	97.60 ± 1.18	99.17 ± 0.16	97.89 ± 1.62	**91.34** ± 7.95
	mean	66.93 ± 0.25	**67.28** ± 0.75	60.57 ± 0.78	**55.18** ± 0.30
	q50	65.08 ± 0.14	**63.28** ± 0.18	57.51 ± 0.82	**54.23** ± 0.29
	q90	69.87 ± 0.72	**91.12** ± 10.95	71.68 ± 9.81	**60.45** ± 0.85
RAsN-guided	max	**97.79** ± 0.50	**99.39** ± 0.34	**98.30** ± 1.77	80.36 ± 7.74
	mean	**67.92** ± 0.25	65.07 ± 0.53	**66.05** ± 0.63	53.98 ± 0.27
	q50	**65.95** ± 0.25	62.95 ± 0.20	**62.96** ± 0.22	53.20 ± 0.31
	q90	**72.26** ± 0.78	67.26 ± 0.84	**78.87** ± 6.78	58.38 ± 0.64

C Theoretical Results and Proofs

C.1 Entropy-Regularized Assortment Optimization

Proof of Theorem 1. We write the objective function as

$$\Phi(\mu) := \sum_{s\in\mathscr{S}} \mu(s)R(s) + \frac{1}{\beta}\mathcal{H}(\mu) = \sum_{s\in\mathscr{S}} \mu(s)R(s) - \frac{1}{\beta}\sum_{s\in\mathscr{S}} \mu(s)\log\mu(s),$$

where $\mu \in \Delta(\mathscr{S})$, i.e.,

$$\mu(s) \geq 0 \quad \text{for all } s \in \mathscr{S}, \qquad \sum_{s\in\mathscr{S}} \mu(s) = 1.$$

We wish to solve

$$\max_{\mu\in\Delta(\mathscr{S})} \Phi(\mu).$$

We first note that the objective is *strictly concave* in μ, because the term $\sum_s \mu(s)R(s)$ is linear in μ, while the entropy term $-\sum_s \mu(s)\log\mu(s)$ is strictly concave on the probability simplex. Therefore, $\Phi(\mu)$ is strictly concave, and hence it admits at most one maximizer. It thus suffices to characterize the stationary point. To enforce the normalization constraint $\sum_s \mu(s) = 1$, consider the Lagrangian

$$\mathcal{L}(\mu,\lambda) = \sum_{s\in\mathscr{S}} \mu(s)R(s) - \frac{1}{\beta}\sum_{s\in\mathscr{S}} \mu(s)\log\mu(s) + \lambda\left(\sum_{s\in\mathscr{S}} \mu(s) - 1\right).$$

For each $s \in \mathscr{S}$, differentiate $\mathcal{L}$ with respect to $\mu(s)$. Using $\frac{d}{dx}(x\log x) = 1 + \log x$, we obtain

$$\frac{\partial\mathcal{L}}{\partial\mu(s)} = R(s) - \frac{1}{\beta}(1 + \log\mu(s)) + \lambda.$$

At an optimum, the first-order condition gives

$$R(s) - \frac{1}{\beta}(1 + \log\mu(s)) + \lambda = 0.$$

Rearranging yields

$$\log\mu(s) = \beta R(s) + \beta\lambda - 1.$$

Hence there exists a constant $C > 0$, independent of s, such that

$$\log\mu(s) = \beta R(s) + C.$$

Exponentiating both sides gives

$$\mu(s) = e^C e^{\beta R(s)}.$$

Now use the constraint that μ is a probability distribution:

$$1 = \sum_{s\in\mathscr{S}} \mu(s) = e^C \sum_{s\in\mathscr{S}} e^{\beta R(s)}.$$

Therefore,

$$e^C = \left(\sum_{s'\in\mathscr{S}} e^{\beta R(s')}\right)^{-1}.$$

Substituting this back into the expression for $\mu(s)$, we obtain

$$\mu^\star(s) = \frac{e^{\beta R(s)}}{\sum_{s'\in\mathscr{S}} e^{\beta R(s')}}, \quad s \in \mathscr{S}.$$

Finally, since Φ is strictly concave over $\Delta(\mathscr{S})$, this stationary point $\mu^\star$ is the unique maximizer. This completes the proof. □

C.2 FORWARD CORRUPTION

We first justify the use of the uniform distribution as the terminal reference distribution for the forward process. The forward corruption kernel independently either preserves each bit or resamples it from a uniform Bernoulli distribution. Therefore, the amount of information retained from the initial assortment can be tracked explicitly through the cumulative retention factor

$$\bar{\alpha}_t := \prod_{\tau=1}^{t}(1-\beta_\tau).$$

The following proposition gives a finite-time bound showing that the forward process approaches the uniform distribution as $\bar{\alpha}_t$ becomes small.

Proposition 3 (Forward corruption approaches the uniform distribution)**.** *Let $(s_t)_{t\geq 0}$ be the forward corruption process on $\{0,1\}^N$ defined by*

$$q_t(s_t \mid s_{t-1}) = \prod_{i=1}^{N}\left[(1-\beta_t)\mathbf{1}\{s_{t,i}=s_{t-1,i}\}+\frac{\beta_t}{2}\right],$$

where $\beta_t \in (0,1)$. Define

$$\bar{\alpha}_t := \prod_{\tau=1}^{t}(1-\beta_\tau).$$

Then, for any initial state $s_0 \in \{0,1\}^N$ and any coordinate $i \in [N]$,

$$\mathbb{P}(s_{t,i}=1 \mid s_0) = \frac{1}{2}+\bar{\alpha}_t\left(s_{0,i}-\frac{1}{2}\right).$$

Equivalently, the conditional law of s_t given s_0 factorizes as

$$q_t(s_t \mid s_0) = \prod_{i=1}^{N}\left[\frac{1}{2}+\bar{\alpha}_t\left(s_{0,i}-\frac{1}{2}\right)(2s_{t,i}-1)\right].$$

In particular, if u denotes the uniform distribution on $\{0,1\}^N$, then

$$\|q_t(\cdot \mid s_0)-u\|_{\mathrm{TV}} \leq \frac{N}{2}\bar{\alpha}_t.$$

Remark 4. Proposition 3 is a finite-time statement. It shows that the terminal distribution is close to uniform whenever the cumulative retention factor $\bar{\alpha}_t = \prod_{\tau=1}^{t}(1-\beta_\tau)$ is small. In our implementation, we use a finite schedule with $T = 100$ and β_t linearly increasing from 10^{-4} to 0.1, for which $\bar{\alpha}_T$ is small but nonzero. If one considers an infinite schedule satisfying $\sum_{t=1}^{\infty}\beta_t = \infty$, then $\bar{\alpha}_t \to 0$ since

$$\log\bar{\alpha}_t = \sum_{\tau=1}^{t}\log(1-\beta_\tau) \leq -\sum_{\tau=1}^{t}\beta_\tau \to -\infty,$$

and hence the forward process converges to the uniform distribution in total variation.

Proof of Proposition 3. We first analyze one coordinate. Fix $i \in [N]$ and define

$$m_{t,i} := \mathbb{P}(s_{t,i}=1 \mid s_0).$$

By the definition of the corruption kernel, conditional on $s_{t-1,i}$, the next bit is kept with probability $1-\beta_t$ and is resampled uniformly from $\{0,1\}$ with probability β_t. Therefore,

$$m_{t,i} = \left(1-\frac{\beta_t}{2}\right)m_{t-1,i}+(1-m_{t-1,i})\frac{\beta_t}{2} = (1-\beta_t)m_{t-1,i}+\frac{\beta_t}{2}.$$

Subtracting $1/2$ from both sides gives

$$m_{t,i}-\frac{1}{2} = (1-\beta_t)\left(m_{t-1,i}-\frac{1}{2}\right).$$

Iterating this recursion yields

$$m_{t,i} - \frac{1}{2} = \left(\prod_{\tau=1}^{t}(1-\beta_\tau)\right)\left(s_{0,i} - \frac{1}{2}\right) = \bar{\alpha}_t\left(s_{0,i} - \frac{1}{2}\right).$$

Hence

$$\mathbb{P}(s_{t,i} = 1 \mid s_0) = \frac{1}{2} + \bar{\alpha}_t\left(s_{0,i} - \frac{1}{2}\right).$$

Since the forward corruption acts independently across coordinates, conditional on s_0 the coordinates of s_t are independent. Thus,

$$q_t(s_t \mid s_0) = \prod_{i=1}^{N} \mathbb{P}(s_{t,i} \mid s_0).$$

Using the expression above, each coordinate marginal can be written as

$$\mathbb{P}(s_{t,i} \mid s_0) = \frac{1}{2} + \bar{\alpha}_t\left(s_{0,i} - \frac{1}{2}\right)(2s_{t,i} - 1),$$

which proves the factorized expression.

Finally, let $q_{t,i}$ denote the marginal law of $s_{t,i}$ given s_0, and let u_i denote the uniform distribution on $\{0,1\}$. Then

$$\|q_{t,i} - u_i\|_{\mathrm{TV}} = \left|\mathbb{P}(s_{t,i} = 1 \mid s_0) - \frac{1}{2}\right| = \bar{\alpha}_t\left|s_{0,i} - \frac{1}{2}\right| = \frac{\bar{\alpha}_t}{2}.$$

Using the tensorization bound

$$\left\|\bigotimes_{i=1}^{N} q_{t,i} - \bigotimes_{i=1}^{N} u_i\right\|_{\mathrm{TV}} \le \sum_{i=1}^{N} \|q_{t,i} - u_i\|_{\mathrm{TV}},$$

we obtain

$$\|q_t(\cdot \mid s_0) - u\|_{\mathrm{TV}} \le \sum_{i=1}^{N} \frac{\bar{\alpha}_t}{2} = \frac{N}{2}\bar{\alpha}_t.$$

This completes the proof. □

This result formalizes the intuition that the forward process progressively erases the structure of the original assortment. After sufficiently many corruption steps, the terminal distribution becomes close to the uniform reference distribution, which justifies initializing the learned reverse process from $s_T \sim \mathrm{Uniform}(\{0,1\}^N)$.

C.3 Denoising Process

We next characterize what the denoising objective learns in the population limit. The reverse model is trained to predict the clean assortment s_0 from a corrupted state (s_t, t) using coordinate-wise binary cross-entropy. The following result shows that, with infinite data and an unrestricted function class, the optimal predictor recovers the posterior marginal probability that each clean bit equals one.

Proposition 5 (Population denoising learns posterior clean-sample marginals)**.** *Let μ_0 be a distribution over clean assortments $s_0 \in \{0,1\}^N$. For each $t \in \{1, \cdots, T\}$, let $q_t(\cdot \mid s_0)$ denote the forward corruption kernel from s_0 to s_t. Suppose the training data are generated by*

$$s_0 \sim \mu_0, \qquad t \sim \pi, \qquad s_t \sim q_t(\cdot \mid s_0),$$

where π is a distribution over $\{1, \cdots, T\}$ with full support. Consider the population denoising objective over measurable logit functions $g : \{0,1\}^N \times \{1, \cdots, T\} \to \mathbb{R}^N$:

$$\mathcal{L}(g) = \mathbb{E}\left[-\sum_{i=1}^{N}\left\{s_{0,i}\log\sigma(g_i(s_t,t)) + (1-s_{0,i})\log\left(1-\sigma(g_i(s_t,t))\right)\right\}\right].$$

Then any population minimizer $g^\star$ satisfies, for every coordinate $i \in [N]$,

$$\sigma(g_i^\star(s_t,t)) = \mathbb{P}(s_{0,i} = 1 \mid s_t, t)$$

almost surely with respect to the joint distribution of (s_t, t)*. Equivalently, whenever*

$$0 < \mathbb{P}(s_{0,i} = 1 \mid s_t, t) < 1,$$

the optimal logit is

$$g_i^\star(s_t, t) = \log \frac{\mathbb{P}(s_{0,i} = 1 \mid s_t, t)}{\mathbb{P}(s_{0,i} = 0 \mid s_t, t)}.$$

Proof. The loss decomposes over coordinates. Fix a coordinate i and condition on a particular value $(s_t, t) = (z, \tau)$. Let

$$\eta_i(z, \tau) := \mathbb{P}(s_{0,i} = 1 \mid s_t = z, t = \tau).$$

For a predicted probability $a \in (0, 1)$, the conditional binary cross-entropy risk is

$$r(a; z, \tau) = -\eta_i(z, \tau) \log a - (1 - \eta_i(z, \tau)) \log(1 - a).$$

This is minimized over $a \in (0, 1)$ at $a = \eta_i(z, \tau)$. Indeed,

$$\frac{\partial r}{\partial a} = -\frac{\eta_i(z, \tau)}{a} + \frac{1 - \eta_i(z, \tau)}{1 - a},$$

and setting this derivative equal to zero gives

$$a = \eta_i(z, \tau).$$

Moreover, the second derivative is

$$\frac{\partial^2 r}{\partial a^2} = \frac{\eta_i(z, \tau)}{a^2} + \frac{1 - \eta_i(z, \tau)}{(1 - a)^2} > 0,$$

so this minimizer is unique whenever $\eta_i(z, \tau) \in (0, 1)$.

Since $a = \sigma(g_i(z, \tau))$, the population minimizer satisfies

$$\sigma(g_i^\star(z, \tau)) = \eta_i(z, \tau) = \mathbb{P}(s_{0,i} = 1 \mid s_t = z, t = \tau).$$

Taking the logit transform gives

$$g_i^\star(z, \tau) = \log \frac{\eta_i(z, \tau)}{1 - \eta_i(z, \tau)} = \log \frac{\mathbb{P}(s_{0,i} = 1 \mid s_t = z, t = \tau)}{\mathbb{P}(s_{0,i} = 0 \mid s_t = z, t = \tau)},$$

whenever the posterior marginal lies strictly between 0 and 1. □

This proposition formalizes the clean-sample prediction interpretation of our denoising model. The learned probabilities $\sigma(g_i(s_t, t))$ approximate the posterior clean-bit marginals induced by the empirical behavior distribution and the forward corruption process. Thus, before reward guidance is applied, the reverse model acts as a behavior-prior denoiser rather than as an explicit optimizer.

C.4 Reward Guidance as KL-regularized Local Policy Improvement

For completeness, we restate Theorem 2 before giving the proof.

Restatement of Theorem 2. Fix a reverse diffusion step t and a current state $s_t \in \{0, 1\}^N$. Suppose the unguided reverse proposal is a factorized Bernoulli distribution

$$p_\phi(x \mid s_t, t) = \prod_{i=1}^{N} \text{Bernoulli}\left(x_i; \sigma(g_\phi^{(i)}(s_t, t))\right), \qquad x \in \{0, 1\}^N.$$

Let $R : \{0, 1\}^N \to \mathbb{R}$ be a reward function. For each coordinate $i \in [N]$, define the local reward difference

$$\Delta_i R(s_t) := R(s_t^{(i=1)}) - R(s_t^{(i=0)}),$$

where $s_t^{(i=1)}$ and $s_t^{(i=0)}$ denote the vectors obtained from s_t by setting the i-th coordinate to 1 and 0, respectively. Define the coordinate-wise local reward surrogate

$$\ell_{s_t}(x) := \sum_{i=1}^{N} x_i \Delta_i R(s_t).$$

For any $\lambda_t > 0$, consider the KL-regularized local improvement problem

$$\max_{q \in \Delta(\{0,1\}^N)} \left\{ \mathbb{E}_{x \sim q} \left[\ell_{s_t}(x) \right] - \frac{1}{\lambda_t} \mathrm{KL} \left(q \,\|\, p_\phi(\cdot \mid s_t, t) \right) \right\}.$$

Then the unique optimizer is

$$q_t^\star(x \mid s_t) = \frac{p_\phi(x \mid s_t, t) \exp\left(\lambda_t \ell_{s_t}(x)\right)}{Z_t(s_t)},$$

where

$$Z_t(s_t) = \sum_{x \in \{0,1\}^N} p_\phi(x \mid s_t, t) \exp\left(\lambda_t \ell_{s_t}(x)\right).$$

Moreover, $q_t^\star(\cdot \mid s_t)$ factorizes across coordinates, and its coordinate-wise logits satisfy

$$\log \frac{q_t^\star(x_i = 1 \mid s_t)}{q_t^\star(x_i = 0 \mid s_t)} = g_\phi^{(i)}(s_t, t) + \lambda_t \Delta_i R(s_t).$$

Equivalently,

$$q_t^\star(x_i = 1 \mid s_t) = \sigma\left(g_\phi^{(i)}(s_t, t) + \lambda_t \Delta_i R(s_t) \right).$$

Thus, the reward-guided reverse transition is the solution of a KL-regularized local policy improvement problem, where the unguided reverse proposal acts as the reference distribution and ℓ_{s_t} acts as a coordinate-wise local approximation of the reward.

Proof. Fix s_t and t. For notational simplicity, we write

$$p_\phi(x) := p_\phi(x \mid s_t, t), \qquad \ell(x) := \ell_{s_t}(x).$$

The local improvement objective is

$$\mathcal{J}(q) = \mathbb{E}_{x \sim q}[\ell(x)] - \frac{1}{\lambda_t} \mathrm{KL}(q \,\|\, p_\phi).$$

Expanding the KL divergence gives

$$\mathcal{J}(q) = \sum_x q(x) \ell(x) - \frac{1}{\lambda_t} \sum_x q(x) \log \frac{q(x)}{p_\phi(x)}.$$

Rearranging terms, we obtain

$$\mathcal{J}(q) = -\frac{1}{\lambda_t} \sum_x q(x) \log \frac{q(x)}{p_\phi(x) \exp(\lambda_t \ell(x))}.$$

Define the normalizing constant

$$Z := \sum_{x \in \{0,1\}^N} p_\phi(x) \exp(\lambda_t \ell(x)).$$

Then

$$\bar{q}(x) := \frac{p_\phi(x) \exp(\lambda_t \ell(x))}{Z}$$

is a probability distribution over $\{0,1\}^N$. Using this definition, we can rewrite the objective as

$$\mathcal{J}(q) = \frac{1}{\lambda_t} \log Z - \frac{1}{\lambda_t} \mathrm{KL}(q \,\|\, \bar{q}).$$

Since $\mathrm{KL}(q \,\|\, \bar{q}) \geq 0$, with equality if and only if $q = \bar{q}$, the unique maximizer is

$$q_t^\star(x \mid s_t) = \bar{q}(x) = \frac{p_\phi(x \mid s_t, t) \exp(\lambda_t \ell_{s_t}(x))}{Z_t(s_t)}.$$

This proves the first claim.

It remains to show that this optimizer factorizes and has the claimed coordinate-wise logits. By assumption,

$$p_\phi(x \mid s_t, t) = \prod_{i=1}^{N} \sigma(g_\phi^{(i)}(s_t, t))^{x_i} \left(1 - \sigma(g_\phi^{(i)}(s_t, t))\right)^{1-x_i}.$$

Also, since

$$\ell_{s_t}(x) = \sum_{i=1}^{N} x_i \Delta_i R(s_t),$$

we have

$$\exp(\lambda_t \ell_{s_t}(x)) = \prod_{i=1}^{N} \exp\left(\lambda_t x_i \Delta_i R(s_t)\right).$$

Therefore,

$$q_t^\star(x \mid s_t) \propto \prod_{i=1}^{N} \left[\sigma(g_\phi^{(i)}(s_t, t))^{x_i} \left(1 - \sigma(g_\phi^{(i)}(s_t, t))\right)^{1-x_i} \exp\left(\lambda_t x_i \Delta_i R(s_t)\right)\right].$$

Hence $q_t^\star(\cdot \mid s_t)$ factorizes across coordinates. For each coordinate i, the odds ratio under $q_t^\star$ is

$$\frac{q_t^\star(x_i = 1 \mid s_t)}{q_t^\star(x_i = 0 \mid s_t)} = \frac{\sigma(g_\phi^{(i)}(s_t, t))}{1 - \sigma(g_\phi^{(i)}(s_t, t))} \exp\left(\lambda_t \Delta_i R(s_t)\right).$$

Taking logarithms gives

$$\log \frac{q_t^\star(x_i = 1 \mid s_t)}{q_t^\star(x_i = 0 \mid s_t)} = \log \frac{\sigma(g_\phi^{(i)}(s_t, t))}{1 - \sigma(g_\phi^{(i)}(s_t, t))} + \lambda_t \Delta_i R(s_t).$$

Since $\log \frac{\sigma(a)}{1-\sigma(a)} = a$, we obtain

$$\log \frac{q_t^\star(x_i = 1 \mid s_t)}{q_t^\star(x_i = 0 \mid s_t)} = g_\phi^{(i)}(s_t, t) + \lambda_t \Delta_i R(s_t).$$

Equivalently,

$$q_t^\star(x_i = 1 \mid s_t) = \sigma\left(g_\phi^{(i)}(s_t, t) + \lambda_t \Delta_i R(s_t)\right).$$

This is exactly the reward-guided reverse transition. □