Marginal Distribution Adaptation for Discrete Sets via Module-Oriented Divergence Minimization

Hanjun Dai [1] **Mengjiao Yang** [1] **Yuan Xue** [2] **Dale Schuurmans** [1] **Bo Dai** [1]

Abstract

Distributions over discrete sets capture the essential statistics including the high-order correlation among elements. Such information provides powerful insight for decision making across various application domains, *e.g.* product assortment based on product distribution in shopping carts. While deep generative models trained on pre-collected data can capture existing distributions, such pre-trained models are usually not capable of aligning with a target domain in the presence of distribution shift due to reasons such as temporal shift or the change in the population mix. We develop a general framework to adapt a generative model subject to a (possibly counterfactual) target data distribution with both sampling and computation efficiency. Concretely, instead of re-training a full model from scratch, we reuse the learned modules to preserve the correlations between set elements, while only adjusting corresponding components to align with target marginal constraints. We instantiate the approach for three commonly used forms of discrete set distribution—latent variable, autoregressive, and energy based models—and provide efficient solutions for marginal-constrained optimization in either primal or dual forms. Experiments on both synthetic and real-world e-commerce and EHR datasets show that the proposed framework is able to practically align a generative model to match marginal constraints under distribution shift.

1. Introduction

Discrete sets are a common datatype in real world applications, typically encountered, for example, in carts of products in online shopping, sets of diagnosis codes for individual patients in their electronic health records (EHR), or even bag-of-word representations of documents. Understanding correlations between set elements provides essential insight in these domains, and has been a major topic in machine learning and data mining research (Han et al., 2011). Deep generative models, including deep latent variable models (Kingma & Welling, 2014), autoregressive models (Uria et al., 2016), and deep energy-based models (LeCun et al., 2006), have recently provided powerful new tools for capturing high-order correlations between elements co-occurring in a set. Generated samples of discrete sets from such models, such as synthetic online orders, are often used for evaluating downstream decisions in applications like supply chain fulfillment and product assortment decisions.

Generative models have demonstrated success in discrete set modeling for domains such as document (Blei et al., 2003) and language (Vaswani et al., 2017) modeling, but these successes have generally relied on a basic assumption: that the target distribution matches the distribution that generated the training data (Vapnik, 1999). However, distribution shift is prevalent in real-world scenarios, which can cause poor alignment between previously sampled training data and a current target distribution. One typical reason for such drift is seasonality, for example sales in summer differ from those in winter. Another reason is the need to perform counterfactual simulation for purposes like debiasing EHR data or stress-testing logistic systems. Both cases require the generative model to be adapted to satisfy a (possibly counterfactual) target data distribution.

For discrete sets, the most natural statistics of interest are element marginals, *i.e.*, the occurrence frequency of a particular element in the generated sets. In practice, it is also relatively easy to obtain estimates of such marginals, like sales for a certain product or prevalence of a certain disease, compared to obtaining joint occurrence statistics. In this paper, we therefore focus on developing a practical answer to the following question:

How can we efficiently align an existing generative model to match target marginal specifications, while preserving previously learned correlations between elements?

The most straightforward idea would be to retrain an entire generative model from scratch on data that respects

[1]Google Research, Brain Team [2]Google Cloud. Correspondence to: Hanjun Dai <hadai@google.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

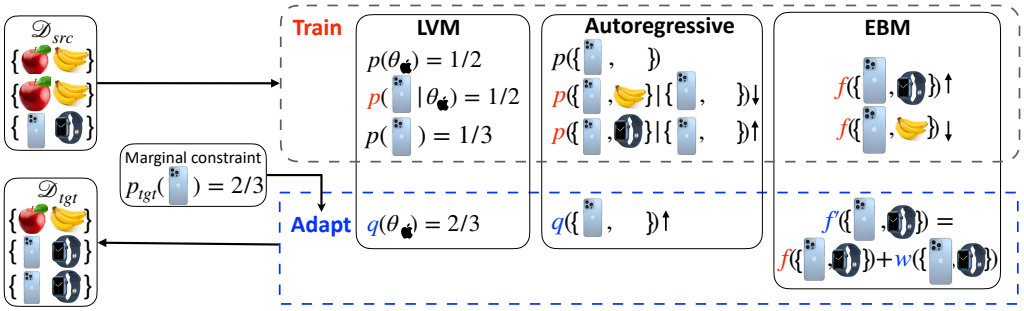

Figure 1. Motivating example and overview of MODEM, the training distribution $\mathcal{D}_{src}$ describes customer orders on a regular day (more fruits than electronics). A store would like to simulate orders around the time of a new iPhone release, while preserving item correlations in the training data (i.e., apples and bananas co-occur, iPhones and watches co-occur). 2/3 of the general population buys new iPhones around the time of release according to some poll (marginal constraint). LVM adapts to the constraint by controlling the latent variable representing the electronics category. Autoregressive model increases the probability that the first generated item is an iPhone. EBM adapts the energy to generate more iPhones. Red denotes modules fixed and reused after training and blue denotes adapted components.

a new marginal specification. However, such a naive approach is maximally inefficient in terms of sample, memory and computational resource use. Fine-tuning a pretrained model (Devlin et al., 2018) is another widely used approach, but this merely uses an existing model as a warm-start, and otherwise retrains a full model on new data reflecting the target distribution. It is not obvious how to bypass such inefficiencies, however, given that all parameters in a model are updated during gradient-based training, and there is no simple mechanism for preserving previous correlations without accessing the original training data. This reveals a delicate trade-off between training efficiency and model reuse.

Our contribution. In this paper, we propose a solution to the question posed, and show how a pre-trained generative model can be adapted, efficiently, to match target marginals while preserving previous correlations. In particular, we maintain previous correlations by explicitly reusing existing modules from the pre-trained model, while to adapt the model to new marginals, we recompose the fixed modules using a simple optimization. This leads to our proposed framework, *Module-Oriented DivErgence Minimization (MODEM)*. More specifically,

- We first introduce the primary contribution, the MODEM framework, from the perspective of constrained divergence minimization in Section 2;
- We instantiate the general framework for specific forms of generative model—latent variable, autoregressive, and energy-based models—in Section 3.1, 3.2, and 3.3, respectively, showing how marginal matching can be efficiently achieved in each case;
- We verify the proposed MODEM framework on the different types of generative model for discrete set modeling, considering both synthetic and four real-world datasets from e-commerce and EHR domains in Section 5. The empirical results demonstrate the effectiveness of MODEM.

We emphasize that although we consider marginal distribution adaptation as the primary motivation, the proposed MODEM framework is far more general and can be easily applied to other distribution alignment problems, which we leave as future work.

2. MODEM Framework

In this section, we first formally introduce the problem setting, then reformulate distribution adaptation as a constrained divergence minimization problem. Key to the approach is to explicitly reuse modules from a pretrained generative model to preserve previously learned correlations. By integrating these two components, we obtain a general framework, *Module-Oriented Divergence Minimization (MODEM)*, that strikes an effective balance between efficiency and model reuse.

Problem Formulation. A discrete set S is defined as a collection of unique elements from a finite domain $X = \{x_1, x_2, \ldots, x_{|X|}\}$. We have $S \in \mathscr{P}(X)$ where $\mathscr{P}(X)$ is the powerset of X. Given a dataset sampled from some unknown source distribution $\mathcal{D}_{src} \sim \hat{p}(S)$, the standard generative modeling task is to learn a model p from a parametrized distribution family $\mathcal{H}$ to approximate the distribution $\hat{p}(S)$. A vast diversity of objectives and methods have been developed for this general problem (Goodfellow et al., 2016).

We will focus on generative model adaption under marginal distribution specification. Concretely, given a learned model $p \in \mathcal{H}$, we would like to find another model $q \in \mathcal{H}$ that satisfies the marginal distribution specification, while the original correlation in p is still preserved in q.

By "marginal distribution specification" we mean

$$|\mathbb{E}_{S\sim q}[\mathbb{I}(e_i \in S)] - t_i| = 0, \forall (e_i, t_i) \in C, \quad (1)$$

where $C = \{c_i = (e_i, t_i)\}_{i=1}^{|C|}$ specifies that a certain ele-

ment $c_i \in X$ would in expectation appear in a $0 \leqslant t_i \leqslant 1$ fraction of all the generated discrete sets.

By "correlation preservation", we mean the high-order moments should be approximately maintained, *i.e.*,

$$|\mathbb{E}_p [\mathbb{I}(A \in S)] - \mathbb{E}_q [\mathbb{I}(A \in S)]| \leqslant \xi, \forall A \in \mathscr{P}(X) \, and \, |A| > 1, \quad (2)$$

with $\xi > 0$ and A_k denotes the subset of S with cardinality equals to k.

Divergence Minimization. The most comprehensive way to approximate a target distribution p with another distribution q would be to approximate all higher order moments. However, such a naive approach can be computationally intractable, given that the number of constraints is more than exponential w.r.t. $|X|$. One can reduce the correlation preservation conditions by only considering the largest differences between high-order moments,

$$\max_{|A|>1} \mathbb{E}_p [\mathbb{I}(A \in S)] - \mathbb{E}_q [\mathbb{I}(A \in S)] \leqslant \xi. \quad (3)$$

But this condition is still with exponential complexity due to the construction of $\{A| \, |A| > 1\}$. However, it establishes the connection to total variation distance, which paves the way for a tractable optimization.

Recall that we can rewrite the total variation distance in a variational form (Gibbs & Su, 2002):

$$d_{TV}(p, q) = \max_{h \in \mathcal{F}_\infty} \mathbb{E}_p [h(S)] - \mathbb{E}_q [h(S)], \quad (4)$$

where $\mathcal{F} = \{h| \, \|h\|_\infty \leqslant 1\}$ denotes the set of functions whose infinity norm is bounded by 1. Therefore, if we relax the requirement that the test set A must be $|A| \geqslant 1$ in correlation preservation condition (3), we directly have $d_{TV}(p, q) \leqslant \xi$ will be a sufficient condition. Thus, following the above derivation, we reformulate the model adaption problem as a constrained optimization:

$$\begin{aligned} \min_{q \in \mathcal{H}} \quad & d_{TV}(p, q) \quad (5) \\ \text{s.t.} \quad & |\mathbb{E}_{S \sim q} [\mathbb{I}(e_i \in S)] - t_i| \leqslant \epsilon, \forall (e_i, t_i) \in C, \end{aligned}$$

where ϵ is a constant for relaxing the constraints, which can be zero if all the marginals must be exactly satisfied.

Eq (5) provides a generic framework for generative adaptation of distributions, where the marginal constraints serve as the hints for target domain. However, it is still difficult to optimize in practice due to the definition of total variation distance over discrete random variables. Therefore, we apply Pinsker's inequality, $d_{TV}(p, q) \leqslant \sqrt{\frac{1}{2} KL(q||p)}$, and obtain a more practical surrogate objective

$$\begin{aligned} \min_{q \in \mathcal{H}} \quad & KL(q||p) \quad (6) \\ \text{s.t.} \quad & |\mathbb{E}_{S \sim q} [\mathbb{I}(e_i \in S)] - t_i| \leqslant \epsilon, \forall (e_i, t_i) \in C. \end{aligned}$$

This optimization view provides a practical way to exploit the pretrained model p to preserve the previously learned correlations as much as possible in q while adapting to the target marginals.

Module Reusable Parametrization. With the proposed divergence minimization view in Eq.(6), one could attempt to apply arbitrary deep probabilistic density models for parametrizing q. However, even though this would be valid in principle, it is suboptimal in terms of sample and computational complexity. Although we do not need to retrain q on previous data $\mathcal{D}_{src}$, which reduces memory complexity by avoiding the revisitation of datasets, the optimization still requires learning a *brand new* model from random initialization, which we have already argued is too inefficient. Instead, we will exploit the structure of specific but still flexible model classes, preserving existing modules in a pretrained model and only incrementally modifying how existing modules are combined, which can dramatically save computational and sample complexity.

For different generative model classes, effective techniques for composing a new model from pretrained modules will be different. In Section 3, we instantiate the proposed MODEM framework on three of the most popular and powerful model classes; namely latent variable models (Section 3.1), autoregressive models (Section 3.2) and energy-based models (Section 3.3) for discrete set modeling. In each case, we derive the efficient algorithms for solving Eq (6), in either the primal or dual forms.

3. Instantiations of MODEM

In this section, we show how the MODEM framework can be concretely and practically instantiated with different classes of generative model. Below we will use blue to highlight a new component that will be learned, and red to highlight the fixed modules that will be frozen from an existing model.

3.1. Latent variable models

Latent variable models (LVMs) have been commonly used for generative modeling of documents (Gu & Kong, 2020) and images (Kingma & Welling, 2014), and conveniently it is also natural to use them for unordered sets. For ease of representation, we use a binary vector B to equivalently represent a set S. That is to say, $B \in \{0, 1\}^{|X|}$ indicates the presence or absence of certain values, such that $B_i = \mathbb{I}(x_i \in S)$. Then, according to the De Finetti's Theorem, any joint distribution can be represented as follows:

$$p(B) = \int_\theta p(\theta) \prod_{i=1}^{|X|} p(B_i|\theta) \quad (7)$$

When θ is discrete and the summation is tractable, one can calculate $p(B)$ in a closed form to support efficient maxi-

mum likelihood estimation on a given dataset $\mathcal{D}_{src}$. When θ is in a continuous domain, techniques like VAE (Kingma & Welling, 2014) are needed to optimize the evidence lower-bound. We do not focus on the learning of $(p(\theta), p(B|\theta))$, which can be done by standard techniques; rather, we focus on the adaptation of q from p under the target constraints by implementing MODEM.

Marginal estimation: We first need to consider calculation of the marginal in an LVM model for the constraints in Eq (6). Note that by the conditional independence structure in (7), we have

$$\begin{aligned} p(B_i) &= \sum_{\tilde{B}\in\{0,1\}^{|X|}, \tilde{B}_i = B_i} \int_\theta p(\theta) \prod_{j=1}^{|X|} p(\tilde{B}_j|\theta) \\ &= \int_\theta p(\theta) p(B_i|\theta) \left(\sum_{\tilde{B}} \prod_{j\neq i} p(\tilde{B}_j|\theta) \right) \\ &= \int_\theta p(\theta) p(B_i|\theta) \end{aligned} \tag{8}$$

In above equations, the change from first to second line is based on the interchangeability of summation and integration, while the last step is based on the fact that $\sum_{\{\tilde{B}_j, j\neq i\}} \prod_{j\neq i} p(\tilde{B}_j|\theta) = \prod_{j\neq i} \left(\sum_{\{\tilde{B}_j\}} p(\tilde{B}_j|\theta)\right)$ by independence of each factor and $\sum_{\{\tilde{B}_j\}} p(\tilde{B}_j|\theta) = 1$, $\forall j \neq i$, due to the fact that the summation of probabilities over all events equals to 1. Thus, the marginal for element x_i only involves a subset of components in the overall model (7), which can be efficiently calculated.

Adaptation: To adapt the distribution p to a target domain, we propose to reuse the conditional probability module $p(B|\theta)$, since intuitively we can control the generation process via the control over the latent variable θ. Thus we propose to define $q(B)$ in the following form

$$q(B) = \int_\theta q(\theta) \prod_{i=1}^{|X|} p(B_i|\theta) \tag{9}$$

That is to say, we freeze the conditional components from p while adjusting the prior over θ only.

Plug the module-reused parametrization of $q(B)$ into Eq (6), we obtain the instantiation of MODEM for LVMs as

$$\begin{aligned} \min_{q(\theta)} \quad & KL\left(q(\theta)||p(\theta)\right) \\ \text{s.t.} \quad & \left\| \mathbb{E}_{\theta\sim q(\theta)} \left[p(B_{e_i}|\theta)\right] - t_i \right\|_2 \leqslant \epsilon, \forall (e_i, t_i) \in C. \end{aligned} \tag{10}$$

Note that minimizing $KL(q(B)||p(B))$ between joint distributions is equivalent to minimizing $KL\left(q(\theta)||p(\theta)\right)$, where the latter form has a closed form solution when $p(\theta)$ and $q(\theta)$ are from exponential families, such as the multinomial or Gaussian distributions. Therefore, we can simply solve Eq (10) in its primal form via penalty methods.

Parametrization: When θ is categorical and the integration in Eq. (7) is tractable, we simply use a uniform distribution for $p(\theta)$. When θ is continuous and VAE is employed, we use set type encoders like a Transformer (Vaswani et al., 2017; Ren et al., 2021) or simply an MLP on a binary representation to parameterize the variational posterior.

3.2. Autoregressive models

Since any joint distribution can be factorized in an autoregressive manner, autoregressive models have become quite popular, especially for modeling sequences. Despite the presence of a total ordering, which is not desirable for unordered set modeling, autoregressive models have still proved to be quite powerful for discrete set modeling (Vinyals et al., 2015; Gao et al., 2019). In particular, for this model, one can treat a set S with cardinality L as a sequence of L elements: $S = [s_1, s_2, \ldots, s_L]$. Then an autoregressive model defines the distribution as:

$$p(S|L) = \prod_{i=1}^{L} p(s_i|s_{<i}, L) \tag{11}$$

However it is generally hard to compute the marginals for autoregressive models, due to the exponential growth of marginalization cost with respect to the sequence length. So for our purposes, we need to introduce some special structure to support efficient marginal computation. For discrete sets, one reasonable assumption would be to enforce *permutation invariance*, which is also used in several existing works (Zhang et al., 2019; Kosiorek et al., 2020; Locatello et al., 2020; Carion et al., 2020). Suppose we shuffle the sequence S into S^π with a permutation π, we hope to that the following equation still holds,

$$p(S^\pi|L) = \prod_{i=1}^{L} p(s_{\pi_i}|s_{<\pi_i}, L) = p(S^{\pi'}|L) \tag{12}$$

for any two permutations π and π'.

Introducing permutation invariance into autoregressive models can be difficult, but one reasonably effective strategy is to use the following surrogate objective for p:

$$p = \operatorname*{argmax}_p \mathbb{E}_{S\sim\mathcal{D}_{src}} \left[\mathbb{E}_{\pi\sim\text{Uniform}} \left[p(S^\pi)\right]\right] \tag{13}$$

It is also possible to leverage robust learning to further reduce sample complexity, but this is out of scope of this paper, and we leave such refinements as a future extension.

Marginal estimation: With the permutation invariance assumption, the marginals can be calculated efficiently. Specially, for a particular element $x \in X$, we have

$$\begin{aligned} p(x) &= \sum_{L=1}^{|X|} p(L) \sum_{S:|S|=L} p(x \in S|L) \\ &= \sum_{L=1}^{|X|} p(L) \sum_{S:|S|=L} p(s_1 = x|L) \times L \end{aligned} \tag{14}$$

In other words, one can obtain the marginal $p(x)$ simply by accessing the probability of generating x in the first position. We do however have to consider that exact permutation invariance might not have been achieved in p, meaning that the marginal could be improved via additional computation. Note that one can actually unroll Eq (14) further to obtain the marginal via the probability of generating x in either the first or the second positions:

$$p(x) = \sum_{L=1}^{|X|} p(L)\Big(p_1(x|L) + (L-1) \times \sum_{x' \neq x} p_1(x'|L)p_2(x|x', L)\Big) \quad (15)$$

Here we overload the notation a bit to use $p_1(x|L)$ to denote the probability of generating x in the first position in a set of cardinality L, and similarly $p_2(x|x', L)$ is for x at second position given L and first element x'. Unrolling one step increases the computational cost by a factor of $O(|X|)$, which is generally acceptable. Unrolling further quickly becomes impractical, but we found that the second order estimator is sufficient in practice to balance between the estimation quality and computational cost.

Adaptation: From Eq. (15) we can see under the assumption of permutation invariance, we can control the marginal $p(x)$ via the probability of generating x in the first position. This naturally suggests the following adaptation:

$$q(S) = p(|S|)q_1(s_1||S|)\prod_{i=2}^{|S|} p(s_i|s_{<i}, |S|) \quad (16)$$

Thus the marginal estimator for q becomes

$$q(x) = \sum_{L=1}^{|X|} p(L)\Big(q_1(x|L) + (L-1) \times \sum_{x' \neq x} q_1(x'|L)p_2(x|x', L)\Big) \quad (17)$$

Again, in this case the modules in p are preserved and we only need to learn an additional $q_1(\cdot|\cdot)$, which is much easier than learning a full autoregressive model. Note that optimizing Eq (6) can be done effectively as

$$\min_{q_1} \quad \mathbb{E}_{L\sim p(L)} KL\left(q_1(\cdot|L)||p_1(\cdot|L)\right)$$
$$\text{s.t.} \quad \|q(e_i) - t_i\|_2 \leqslant \epsilon, \forall (e_i, t_i) \in C \quad (18)$$

where the KL term is defined over multinomial distributions making it simple to solve. By plugging Eq (17) into Eq (18), we can again solve the above optimization directly in its primal form via penalty methods.

Parametrization: One major property of discrete set modeling is permutation invariance. As Transformers (without positional encoding) are naturally used for modeling permutation invariant data (Kosiorek et al., 2020; Carion et al., 2020), we use these model for parameterization. Note that although this only guarantees permutation invariance for each of the conditional marginals (*i.e.*, $p(s_i|s_{<i}^{\pi}) = p(s_i|s_{<i}^{\pi'})$), we have found it empirically very useful in achieving Eq (12) and obtaining good results.

3.3. Energy-based models

Energy-Based Models (EBMs) are highly expressive for modeling distributions. One only needs to specify an unnormalized score function over the domain, which brings significant flexibility but also incurs significant computational challenges for training and inference in general. In fact, EBMs are particularly convenient for discrete set modeling, as there have been many provably expressive set encoder parameterizations (Zaheer et al., 2017; Yang et al., 2020) proposed in the literature. Similar to LVMs discussed in Section 3.1, here we use a binary vector B to equivalently represent a set S in the same manner. A set distribution can then be simply defined through $f(B)$ as

$$p_f(B) = \frac{\exp(f(B))}{Z_p}, \; Z_f = \sum_{B\in\{0,1\}^{|X|}} \exp(f(B)) \quad (19)$$

where f is the negative energy or score function, which can be a neural network like DeepSets (Zaheer et al., 2017).

Marginal Adaptation: Different from the LVMs and autoregressive models, where the models can be factorized and the module can be extracted explicitly, module factorization can be difficult in EBM from the score function $f(B)$, and thus, making the module reuse becomes non-trivial. However, the module reuse can be naturally derived from the dual form of Eq. 6 with EBMs.

Specifically, given the constraints set C and denote $\phi(B) = [B_{e_1}, B_{e_2}, \ldots, B_{e_{|C|}}]$ and $c = [t_1, t_2, \ldots, t_{|C|}]$, plug this into the optimization (6), we obtain

$$\min_{q\in\mathcal{P}} KL(q||p_f) \quad \text{s.t.} \|\mathbb{E}_q[\phi(B)] - c\|_2 \leqslant \epsilon, \quad (20)$$

The dual form of (20) can be directly obtained via (Altun & Smola, 2006, Theorem 7) as below (with constants omitted),

$$\max_w w^\top c - \log \sum_B \exp(w^\top \phi(B) + f(B)) - \epsilon \|w\|_2, \quad (21)$$

which is equivalent to the MLE for $p_f(B)\, p(c|B)$ with $p(c|B) \propto \exp(w^\top \phi(B))$ with a *single* data point c. Comparing to the primal form (20), which conducts optimization over *all* valid distributions, the dual form (21) is much easier to optimize.

Moreover, we can clearly see from (21) that the whole model $f(B)$ will be frozen and reused during adaptation, while a new component $w^\top \phi(B)$ with w is the only learnable parameter, which has the size equals to the number of constraints. We emphasize due to the equivalence of primal (20) and dual (21), the optimal solution to (20) will be $q(B) \propto \exp(w^\top \phi(B) + f(B))$, which means the module-reuse parametrization does not lose any flexibility.

Parametrization: Although in principle one can use any f to parameterize p, in our experiments we simply use an MLP on the binary representation B without worrying about enforcing permutation invariance explicitly. As learning the discrete set generation for EBMs requires the sampling in discrete space, we leverage the recent advances in sampling from EBMs (Grathwohl et al., 2021) in discrete space for training both p and q, and use the same samplers for generating new samples from the learned models for simulation.

4. Related work

Generative modeling for sets. There has been a growing interest in modeling sets. Most work has focused on learning generative models of sets in continuous domains like point clouds, adapting recent advances in energy based models (Yang et al., 2020), normalizing flows (Rasul et al., 2019) or ODEs (Li et al., 2020). Here the key challenge is to ensure exchangeability. Zhang et al. (2019) leverages a set encoder and gradient to generate the set for permutation invariance, which can be sped up by implicit differentiation (Zhang et al., 2021); Transformers are also permutation invariant by design, which has made them popular as set decoders (Kosiorek et al., 2020; Locatello et al., 2020; Carion et al., 2020) to replace the gradient calculation. From a modeling perspective, these can be seen as EBMs in a continuous domain. However these approaches can not be directly applied to discrete sets. In a discrete domain, (ordered) sets are typically generated in either an autoregressive way (Vinyals et al., 2015; Gao et al., 2019; Emelianenko et al., 2019) or a non-autoregressive manner with LVMs (Gu & Kong, 2020), or under suitable conditional independence assumptions (Rezatofighi et al., 2018; 2017; 2021). Above, we have revisited these three prominent classes of models for discrete set modeling, where we have shown that our proposed marginal adaptation framework can be applied to all of these existing model classes.

Controllable generation. There have been several works in controllable generation for objects like text (Hu et al., 2017) or images (Li et al., 2019). Most such work has focused on controllable generation at the *instance* level, which considers attributes like facial attributes for a human, or sentiments for sentences. In this paper we focus instead on the *population* level control, a more coarse-grained specification than the individual-level control. Population level control is also easier to be specified for the purposes of distribution adaptation. One direct approach to achieving distribution level requirements is via instance control, however due to the interaction between multiple requirements, it is nontrivial to specify such constraints at an instance level.

Posterior Regularization. The posterior regularization (Ganchev et al., 2010; Mann & McCallum, 2010; Zhu et al., 2014) is also introducing constraints on distributions for optimization over densities, which is similar to the proposed MODEM in terms of the objective. However, there are several major differences: **i)**, we are adapting the learned model to align with distribution shift; while the posterior regularization are designed for posterior calculation in variational Bayesian inference; more importantly, **ii)**, we introduce module reusable parametrization for accelerating the adapting in terms of both sample and computational complexity, which was not explored in posterior regularization.

5. Experiment

In this section, we will first validate the correctness of our marginal adaptation framework on synthetic datasets for all the three models in Section 5.1. Then in Section 5.2 we study the effectiveness of the framework in adapting the learned distribution to the target distribution via marginal alignment using real-world datasets. We present the experiment configurations for model architectures, training and evaluation methods used in both sections.

Model configuration: We present the default model configurations here unless later specified.

- **LVM**: for synthetic data, we evaluate both continuous LVMs and discrete LVMs given that the number of latent components is known. For continuous LVMs, we use multilayer perceptron (MLP) with 2 hidden layers of 512 ReLU activated neurons to parameterize both encoders and decoders using VAE objective, where the adaptor $q(\theta)$ is also a Gaussian distribution. For discrete LVMs, the prior $p(\theta)$ is a uniform distribution while $q(\theta)$ is a multinomial distribution in Eq (9).
- **Autoregressive**: We use Transformers (Vaswani et al., 2017) without positional encoding for parameterization of $p(\cdot|L)$ and a learnable logit vector for set size $p(L)$. We use 4 layers with 8 heads in each layer, where the dimensions for embedding and feed-forward layers are 256 and 512, respectively. The adaptor learns q_1 which is in tabular form of size $\texttt{MaxSetSize} \times |X|$.
- **EBM**: We use an MLP with 2 hidden layers of 512 ReLU activated neurons for f used in p. The adaptor consists of an additional parameter w that has the same size as the number of marginal constraints.

Training configuration: By default we train all the base

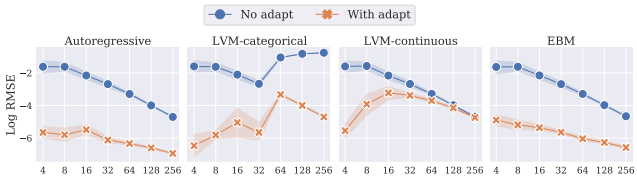

Figure 2. Marginal RMSE on synthetic datasets.

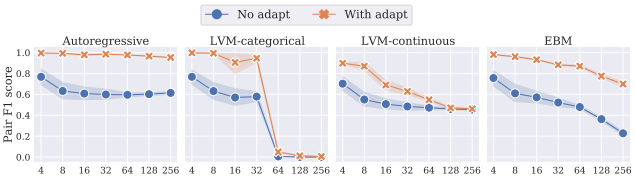

Figure 3. Pairwise-F1 on synthetic datasets.

models p and adapted models q on a single Nvidia V100 GPU with batch size 128, using Adam optimizer. For EBMs training we leverage the PCD framework (Tieleman, 2008) that employs a replay buffer inspired by Du & Mordatch (2019). We use GWG-sampler (Grathwohl et al., 2021) for MCMC sampling. The number of MCMC steps per gradient update varies within $\{50, 100, 200\}$.

Evaluation metrics: Given a set of generated sets $\mathcal{D}_{gen}$ and the sets from $\mathcal{D}_{tgt}$, we use the following two metrics to evaluate the quality:

- **Pairwise-F1** To verify whether the learned adaptation still preserves the learned correlations among elements in a set, we consider the co-occurrence between pairs of elements. Higher-order statistics would be more accurate but would be infeasible to calculate due to the exponential growth of the correlation calculation. Let $c2(x, y; \mathcal{D})$ be the number of sets in $\mathcal{D}$ that contains both x and y, and $c2(\mathcal{D}) = \sum_{x,y} c2(x, y; \mathcal{D})$ be the total counts, then we define the precision as

$$\text{Precision} = \frac{\sum_{x,y} \min\{c2(x, y; \mathcal{D}_{gen}), c2(x, y; \mathcal{D}_{tgt})\}}{c2(\mathcal{D}_{gen})}$$

  and the recall as:

$$\text{Recall} = \frac{\sum_{x,y} \min\{c2(x, y; \mathcal{D}_{gen}), c2(x, y; \mathcal{D}_{tgt})\}}{c2(\mathcal{D}_{tgt})}$$

  and the pairwise F1 as $\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$.
- **Marginal RMSE** Given the marginal specifications $C = \{(e_i, t_i)\}$, the RMSE$(\mathcal{D}, C)$ is computed as

$$\sqrt{\frac{1}{|C|} \sum_{(e_i,t_i)\in C} \left(t_i - \frac{\sum_{S\in\mathcal{D}} \mathbb{I}(e_i \in S)}{|\mathcal{D}|} \right)} \qquad (22)$$

  which measures how faithful the generated set is to the marginal constraints. For better visualization purpose we report log-RMSE (*i.e.*, log of Eq (22)) in the figures.

Remark: Pairwise-F1 is a surrogate to Eq (2) when $|A| = 2$ ($F_1 = 1$ means 0 error in Eq (2)) while Marginal RMSE measures $|A| = 1$. As the computation grows exponentially with $|A|$, it is not practical to evaluate $|A| \geqslant 3$.

5.1. Synthetic experiments

We use the synthetic data to verify the correctness of the proposed adaptation algorithms. In this setup, we construct the domain $X = \{1, \ldots, N\}$ where N is an even number. Each set in this dataset contains only a single pair of digits. There are totally $N/2$ unique sets, where the i-th set contains the pair of digits $(i, i + N/2)$. In this way, we can easily identify whether the model preserves this specific pattern of pairwise correlations or not.

As there are only $N/2$ possible sets in the dataset, we have $|\mathcal{D}_{src}| = N/2$. In other words, each set gets equal probability of being generated in the source data. To construct the counterfactual target set, we randomly generate the marginal distribution $c = [t_1, t_2, \ldots, t_{N/2}]$ for elements $1, \ldots, N/2$, where $\sum_{i=1}^{N/2} t_i = 1$. We vary the dimension $N \in \{4, 8, 16, 32, 64, 128, 256\}$ to see how different models perform when the dimension grows. For each setting of N, we repeat the experiment 10 times with different generated marginal constraints and report the average results.

From Figure 2 we can see that in all the cases, the learned adaptation q would achieve much lower marginal RMSE or better alignment with respect to the target marginal distributions, compared to directly using model p trained on source dataset. We can also see from Figure 3 that when $N = 4$, all the adaptations are able to achieve almost perfect pairwise F1 score. This indicates that our proposed marginal adaptation framework is indeed able to reshape the marginal distribution p while preserving the learned pairwise relationships between elements. When the dimension gets higher, some models like categorical LVM or EBM would suffer from the difficulty of optimization. Improving the efficiency of learning these discrete models is a long standing research problem and is beyond the scope of the paper. Despite of it, we can still see the relative improvements after adaptation in all these settings.

5.2. Real-world experiments

We collect several datasets from both e-commerce and EHR where the discrete sets are core for representation, and the marginal estimations like product sales or disease populations are commonly used. We provide introductions to these datasets below, and the statistics in Table 1.

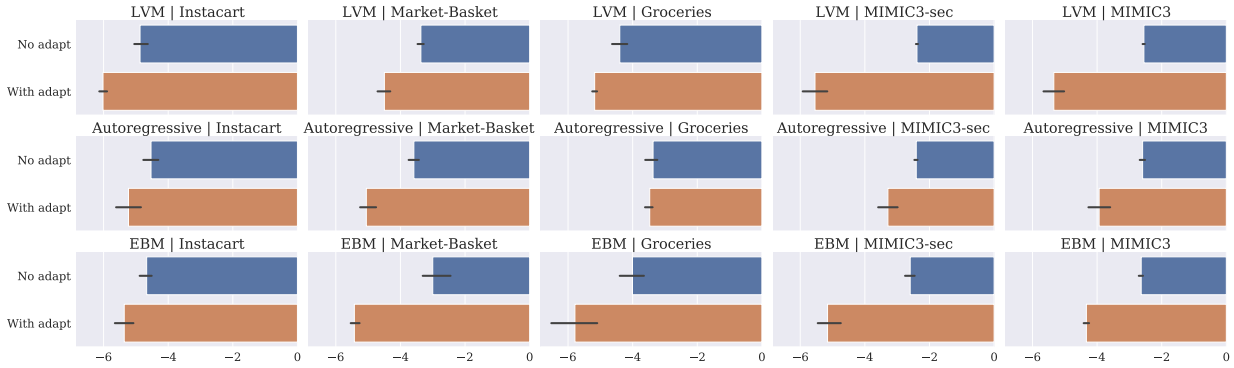

Figure 4. Marginal log-RMSE for models before and after marginal adaptations on real-world datasets.

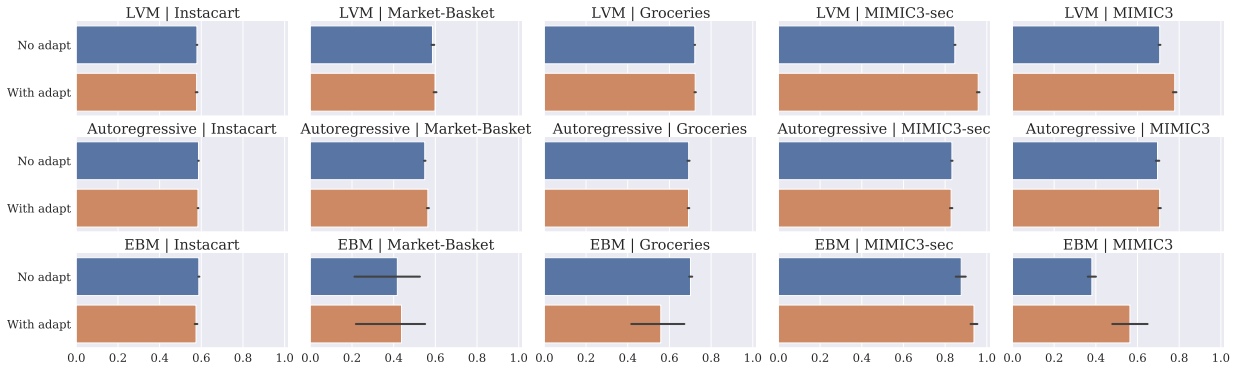

Figure 5. Pairwise-F1 scores for for models before and after marginal adaptations on real-world datasets.

Table 1. Real-world dataset statistics.

Dataset	$\lvert\mathcal{D}_{src}\rvert$	$\lvert\mathcal{D}_{tgt}\rvert$	$\lvert X\rvert$	`MaxSetSize`
Groceries	8,851	984	169	32
Market-Basket	13,466	1,497	167	10
MIMIC3	53,030	5,893	1,070	39
MIMIC3-sec	53,030	5,893	19	16
Instacart	2,963,177	119,533	1,000	79

- Groceries: This dataset consists of transactions from grocery shopping. Each set object in this case represents items like milk, sausage, *etc.*in an order.
- Market-Basket: This one is similar to the Groceries dataset, except that all the transactions are timestamped.
- MIMIC3: We curate this dataset based on the encounter ICD9 diagnosis codes from MIMIC-III (Johnson et al., 2016), an open source EHR dataset. Each record consists of a set of diagnosis code for a patient visit.
- MIMIC3-sec: This dataset is similar to MIMIC3, except that the diagnosis codes are encoded using the 3-digit prefix (chapter level codes) of the ICD9 code.
- Instacart (ins): This dataset comes from the Kaggle Instacart Market Basket Analysis competition. Each online order from a customer is represented as a set of products. We select the top 1,000 popular products for generation and control experiments.

Without timing information, we randomly split the Groceries dataset into $\mathcal{D}_{src}$ and $\mathcal{D}_{tgt}$ with ratio 9:1. For Instacart, we use its own `prior` set as $\mathcal{D}_{src}$ and `train` as $\mathcal{D}_{tgt}$. For all the others with timing information, we sort the datasets according to the timestamp and then use the first 90% as $\mathcal{D}_{src}$ and rest 10% as $\mathcal{D}_{tgt}$. As such, we expect all the datasets except the Groceries to have distribution shift issues, a situation where MODEM would help.

5.2.1. Distribution Shift

To show how the distributions of $\mathcal{D}_{src}$ and $\mathcal{D}_{tgt}$ differ, we visualize the marginal distribution shifts in Figure 6. We compute empirical marginals $p(x_i)$ on source/target sets respectively, and report $p_{tgt}(x_i) - p_{src}(x_i)$ (top row) and $\frac{p_{tgt}(x_i) - p_{src}(x_i)}{p_{src}(x_i)}$ (bottom row) with top-20 largest absolute values. The largest difference can be 12.5% or relatively $2,500\%$, as the dataset is split before/after a given date, following what is commonly done in practice.

5.2.2. Main Results

The marginal adaptation results with 4 marginal constraints are displayed in Figure 4, where similar results are observed with more constraints (see Appendix A). We can see that in all the datasets, all three generative models have achieved lower marginal RMSE after performing the adaptation, es-

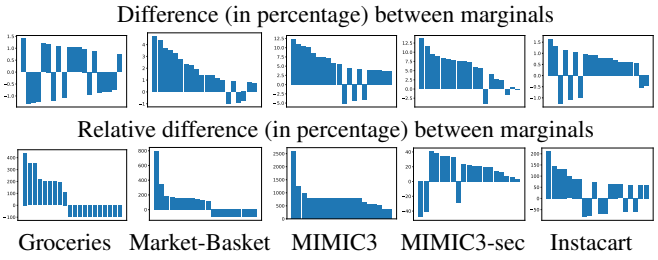

Figure 6. Marginal distribution shifts between training and test.

Table 2. # parameters updated with different methods.

	LVM-continuous	Autoregressive	EBM
(re)training	1,091,239	2,196,657	611,841
MODEM(ours)	512	1,670	167

pecially for models like LVMs and EBMs where the error reduction gets several magnitudes at the best. This in general aligns with our expectation that the constrained optimization would eventually tune the model towards the marginal guidance. Besides the marginal statistics, we are also interested in whether the adaptation preserves the learned correlations between elements or not. We summarize the corresponding results in Figure 5. We can see that the adapted distribution q maintains or even improves the correlation metric. One exception is the EBM on groceries dataset, as this dataset split is not expected to have the distribution shift due to the random data partition, additional adaptation training for EBMs using PCD would not be able to help. Nevertheless, we can see the LVMs and autoregressive models can consistently maintain or improve the distribution alignment regardless of the target data distribution characteristics.

5.2.3. Efficiency

One main advantage of MODEM framework is the efficiency of adaptation. Compared to retraining the entire model, the potential efficiency gain comes from the facts that (a) fewer parameters being updated; (b) fewer number of updates needed.

To validate (a) we report the number of parameters updated for Market-Basket dataset (and other datasets show similar ratios) in Table 2, where MODEM updates fewer than 0.1% of original parameters.

To validate (b), we report the number of steps needed until convergence as this metric determines the number of large model evaluations. In many cases, it only requires 10% steps compared to (re)training from scratch.

5.2.4. Alternative adaptation methods

The problem setting focused in the paper is different from the typical domain adaptation for supervised learning, where one usually has access to unlabeled data of target distribution. Thus most of the domain adaptation methods are not directly applicable. One potentially feasible but inefficient way of doing generative model adaptation with marginal constraints is to re-weight the training samples to match the marginal specification, and retrain the underlying model with re-weighting. Specifically, one needs to first solve

$$\begin{aligned} &\min_{w\in\mathbb{R}^{|\mathcal{D}_{src}|}, w\geqslant 0, |w|=1} H(w) \\ s.t. \quad &\sum_{(S\in\mathcal{D}_{src})} \mathbb{I}(e_i \in S) w_S = t_i, \forall (e_i, t_i) \in C \end{aligned}$$

to obtain the weights of training samples w, and re-train the model p with samples weighted by w. The objective $H(w)$ is the entropy that guarantees the uniqueness of solution. This baseline is expensive since 1) the above optimization can be expensive; 2) the entire model needs to be re-trained. While this alternative method indeed shows benefits in Table 4, it is less effective than our proposed MODEM.

Table 3. # train/adapt steps until convergence.

(train/adapt)	Groceries	Market-Basket	MIMIC3	MIMIC3-sec	Instacart
LVM	18k/1k	10k/1k	32k/1k	24k/1k	23k/1k
Autoregressive	43k/3k	30k/21k	45k/40k	40k/36k	45k/35k
EBM	99k/14k	60k/10k	62k/12k	95k/5k	105k/12k

Table 4. log-RMSE of marginals (averaged over different models).

Methods	Groceries	Market-Basket	MIMIC3	MIMIC3-sec	Instacart
No adapt	-3.95	-3.50	-2.63	-2.53	-4.57
Reweighting	-4.04	-3.33	-2.89	-3.73	-4.82
MODEM(ours)	**-4.80**	**-4.91**	**-4.43**	**-4.32**	**-5.41**

6. Conclusion

In this paper, we proposed MODEM that is able to adapt a trained generative model according to given marginal constraints, in the scenario of discrete set generation. This adaptation framework alleviates the distribution shift issue when applying the generative model for target distribution simulation. Experiments on both synthetic and real-world e-commerce and EHR datasets show that our approach is able to improve the marginal distribution alignment while maintaining the learned co-occurence relationships among elements in a set. Our framework is also generic where the optimization can be potentially applied to domains beyond discrete sets, or handling constraints that are more complicated than single-dimensional marginal distributions. We will explore these extensions in our future work.

Acknowledgements

The authors would like to thank Hieu Pham and anonymous reviewers for their valuable feedbacks on the paper draft, and thank Bethany Wang for her effort in realizing part of this research outcome to industrial applications.

References

Instacart market basket analysis. https://www.kaggle.com/c/instacart-market-basket-analysis/data.

Altun, Y. and Smola, A. Unifying divergence minimization and statistical inference via convex duality. In *International Conference on Computational Learning Theory*, pp. 139–153. Springer, 2006.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, 3: 993–1022, 2003.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Emelianenko, D., Voita, E., and Serdyukov, P. Sequence modeling with unconstrained generation order. *arXiv preprint arXiv:1911.00176*, 2019.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049, 2010.

Gao, T., Chen, J., Chenthamarakshan, V., and Witbrock, M. A sequential set generation method for predicting set-valued outputs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2835–2842, 2019.

Gibbs, A. L. and Su, F. E. On choosing and bounding probability metrics. *International statistical review*, 70 (3):419–435, 2002.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. J. Oops i took a gradient: Scalable sampling for discrete distributions. *arXiv preprint arXiv:2102.04509*, 2021.

Gu, J. and Kong, X. Fully non-autoregressive neural machine translation: Tricks of the trade. *arXiv preprint arXiv:2012.15833*, 2020.

Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2017.

Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *stat*, 1050:10, 2014.

Kosiorek, A. R., Kim, H., and Rezende, D. J. Conditional set generation with transformers. *arXiv preprint arXiv:2006.16841*, 2020.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Li, B., Qi, X., Lukasiewicz, T., and Torr, P. H. Controllable text-to-image generation. *arXiv preprint arXiv:1909.07083*, 2019.

Li, Y., Yi, H., Bender, C. M., Shan, S., and Oliva, J. B. Exchangeable neural ode for set modeling. *arXiv preprint arXiv:2008.02676*, 2020.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020.

Mann, G. S. and McCallum, A. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(2), 2010.

Rasul, K., Schuster, I., Vollgraf, R., and Bergmann, U. Set flow: A permutation invariant normalizing flow. *arXiv preprint arXiv:1909.02775*, 2019.

Ren, H., Dai, H., Dai, Z., Yang, M., Leskovec, J., Schuurmans, D., and Dai, B. Combiner: Full attention transformer with sparse computation cost. *Advances in Neural Information Processing Systems*, 34, 2021.

Rezatofighi, H., Zhu, T., Kaskman, R., Motlagh, F. T., Shi, Q., Milan, A., Cremers, D., Leal-Taixe, L., and Reid, I. Learn to predict sets using feed-forward neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Rezatofighi, S. H., BG, V. K., Milan, A., Abbasnejad, E., Dick, A., and Reid, I. Deepsetnet: Predicting sets with

deep neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5257–5266. IEEE, 2017.

Rezatofighi, S. H., Milan, A., Shi, Q., Dick, A., and Reid, I. Joint learning of set cardinality and state distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17 (1):7184–7220, 2016.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 1999.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vinyals, O., Bengio, S., and Kudlur, M. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Yang, M., Dai, B., Dai, H., and Schuurmans, D. Energy-based processes for exchangeable data. In *International Conference on Machine Learning*, pp. 10681–10692. PMLR, 2020.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

Zhang, Y., Hare, J., and Prugel-Bennett, A. Deep set prediction networks. *Advances in Neural Information Processing Systems*, 32:3212–3222, 2019.

Zhang, Y., Zhang, D. W., Lacoste-Julien, S., Burghouts, G. J., and Snoek, C. G. Multiset-equivariant set prediction with approximate implicit differentiation. *arXiv preprint arXiv:2111.12193*, 2021.

Zhu, J., Chen, N., and Xing, E. P. Bayesian inference with posterior regularization and applications to infinite latent svms. *The Journal of Machine Learning Research*, 15(1): 1799–1847, 2014.

A. More experimental results

A.1. Effect on the number of marginal constraints

In the main paper we presented the real-world experimental results with 4 marginal constraints (*i.e.*, placing marginal constraints on 4 most popular elements). Here we place more constraints and see how the alignment would change when more hints are added.

Figure 8 and Figure 7 show the results with 8 marginal constraints. Figure 10 and Figure 9 show the results with 16 marginal constraints. We can see overall the adapted model still achieve much lower marginal RMSE compared to the original model before adaptation. The change to the F1 score when more constraints are added is not very significant. One possible reason is that as we select the most popular several elements for the constraints, the effect of the constraints diminishes when the popularity of newly added items decreases. Nevertheless, in all these settings we can still see the effectiveness of the adaptation framework in improving the alignment of different generative models to the target distribution.

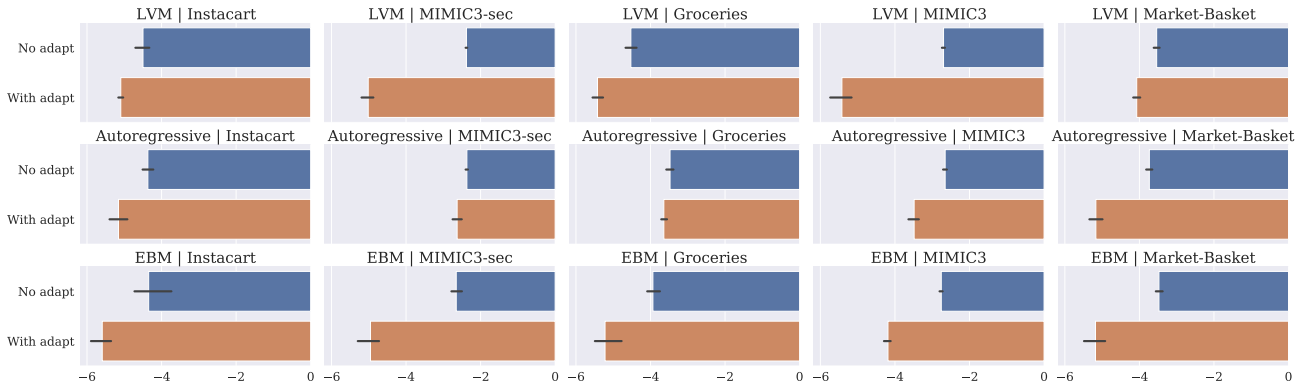

Figure 7. Marginal log-RMSE for models before and after marginal adaptations with 8 marginal constraints.

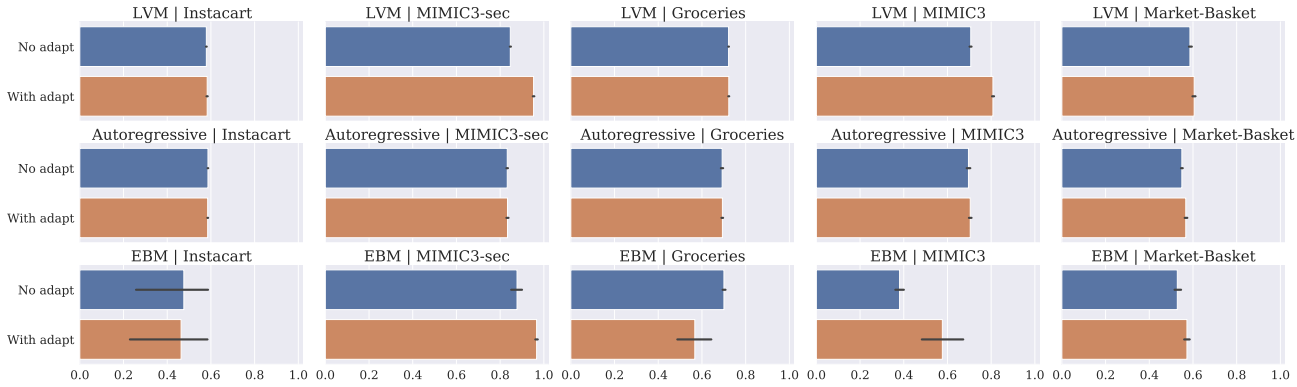

Figure 8. Pairwise-F1 scores for for models before and after marginal adaptations with 8 marginal constraints.

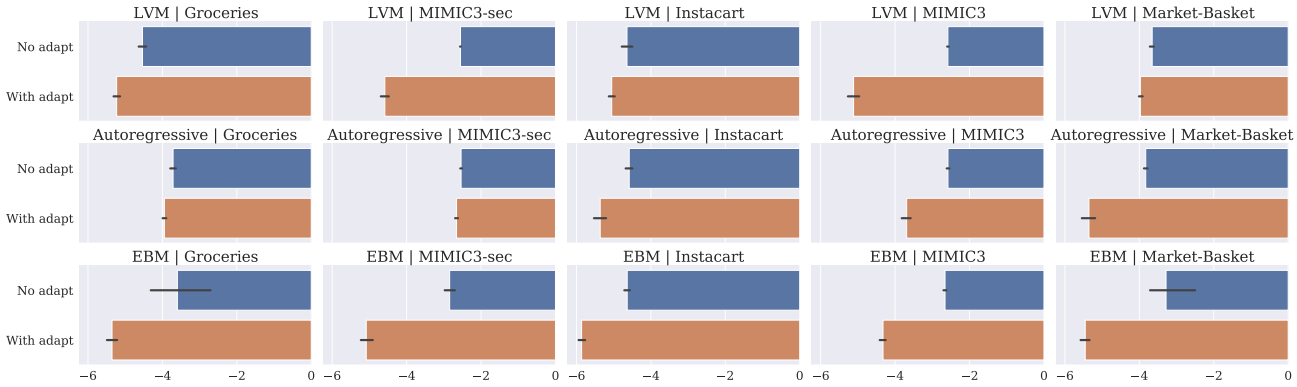

Figure 9. Marginal log-RMSE for models before and after marginal adaptations with 16 marginal constraints.

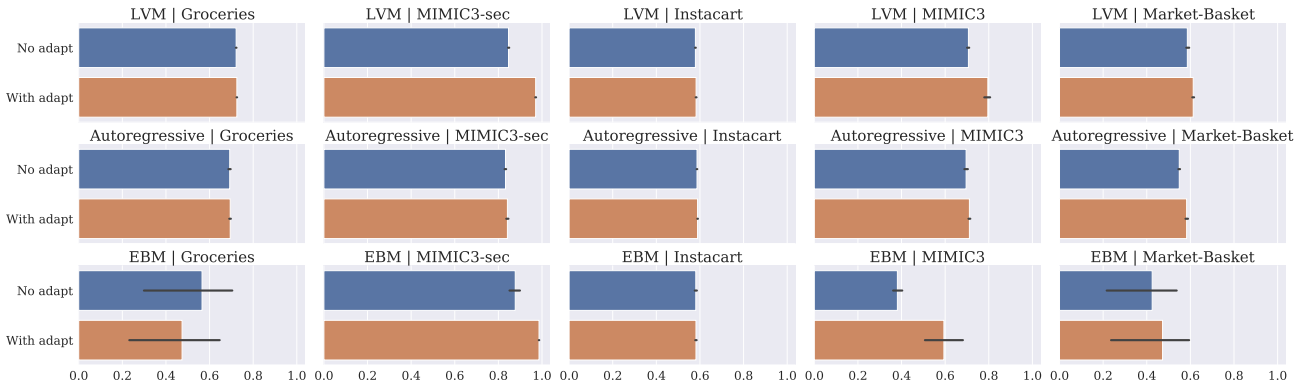

Figure 10. Pairwise-F1 scores for for models before and after marginal adaptations with 16 marginal constraints.