Parameter Estimation in DAGs from Incomplete Data via Optimal Transport

Vy Vo [1 2] **Trung Le** [1] **Tung-Long Vuong** [1 3] **He Zhao** [2] **Edwin V. Bonilla** [2] **Dinh Phung** [1 3]

Abstract

Estimating the parameters of a probabilistic directed graphical model from incomplete data is a long-standing challenge. This is because, in the presence of latent variables, both the likelihood function and posterior distribution are intractable without assumptions about structural dependencies or model classes. While existing learning methods are fundamentally based on likelihood maximization, here we offer a new view of the parameter learning problem through the lens of optimal transport. This perspective licenses a general framework that operates on any directed graphs without making unrealistic assumptions on the posterior over the latent variables or resorting to variational approximations. We develop a theoretical framework and support it with extensive empirical evidence demonstrating the versatility and robustness of our approach. Across experiments, we show that not only can our method effectively recover the ground-truth parameters but it also performs comparably or better than competing baselines on downstream applications.

1. Introduction

Learning probabilistic directed graphical models (DGMs, also known as Bayesian networks) with latent variables is an ongoing challenge in machine learning and statistics. This paper focuses on parameter learning, i.e., estimating the parameters of a DGM with its structure known. Learning DGMs has a long history, dating back to classical indirect likelihood-maximization approaches such as expectation maximization (EM, Dempster et al., 1977). Despite all its success stories, EM is known to suffer from local optima issues. More importantly, EM becomes inapplicable when the posterior distribution is intractable, which arises fairly often in practice. Furthermore, EM is originally a batch algorithm, thereby converging slowly on large datasets (Liang & Klein, 2009). Subsequently, researchers have explored combining EM with approximate inference along with other strategies to improve efficiency (Wei & Tanner, 1990; Neal & Hinton, 1998; Delyon et al., 1999; Beal & Ghahramani, 2006; Cappé & Moulines, 2009; Liang & Klein, 2009; Neath et al., 2013). A large family of approximation algorithms based on variational inference (VI, Jordan et al., 1999; Hoffman et al., 2013) have demonstrated tremendous potential, where the evidence lower bound (ELBO) is not only used for posterior approximation but also for point estimation of the model parameters. Such an approach has proved effective and robust to overfitting, especially when having a small number of parameters. VI has recently taken a leap forward by embracing amortized inference (Amos, 2022), which performs black-box optimization in a considerably more efficient way.

Prior to parameter estimation, both EM and VI consist of an inference step which ultimately requires carrying out expectations of the commonly intractable posterior over the latent variables. In order to address this challenge, a large spectrum of methods have been proposed in the literature and we refer the reader to Ambrogioni et al. (2021) for an excellent discussion of these approaches. Here we characterize them between two extremes. At one extreme, restrictive assumptions about the structure (e.g., as in mean-field approximations) or the model class (e.g., using conjugate exponential families) must be made to simplify the task. At the other extreme, when no assumptions are made, most existing black-box methods exploit very little information about the structure of the known probabilistic model, e.g., in black-box and stochastic VI (Ranganath et al., 2014; Hoffman et al., 2013), hierarchical approaches (Ranganath et al., 2016) or normalizing flows (Papamakarios et al., 2021). Section 2 summarizes the progression of VI research towards this extreme. Since the ultimate goal of VI is posterior inference, parameter estimation has been treated as a by-product of the optimization process where the model parameters are cast as global latent variables. As the complexity of the graph increases, parameter estimation in VI becomes computationally challenging.

A natural question arises as to whether one can learn the parameters of any DGMs with hidden nodes without explicitly solving inference nor assuming any structural independen-

[1]Monash University, Australia [2]CSIRO's Data61, Australia [3]VinAI Research, Vietnam. Correspondence to: Vy Vo <v.vo@monash.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

cies. In this work, we revisit the classic problem of learning graphical models from the viewpoint of optimal transport (OT, Villani et al., 2009), which permits a scalable and general framework that addresses the above criterion.

OT as an alternative to MLE. Estimating the model parameters is essentially about learning a probability density from empirical data. EM and VI are fundamentally based on maximum likelihood estimation (MLE), which amounts to, asymptotically, minimizing the KL (Kullback-Leibler) divergence between the true data and model distribution. We here propose to find a point estimate that minimizes the Wasserstein (WS) distance (Kantorovich, 1960) between these two distributions. The motivations of using WS distance to this problem are three-fold.

First, the measurability and consistency of the minimum Wasserstein estimators have been rigorously studied in prior research, notably in Bassetti et al. (2006); Bernton et al. (2019). **Second,** WS distance is a metric, thus serving as a more sensible measure of distance between two distributions, especially those that are supported on low dimensional manifolds with negligible intersection of support, where standard metrics such as the KL or JS (Jensen-Shannon) divergences are either infinite or undefined (Peyré et al., 2017; Ambrogioni et al., 2018).

Finally, we substantiate the motivation of using OT for graphical learning with an additional experiment of learning GMM under mis-specifications. The task is to estimate the means of a mixture of two bi-variate Gaussian distributions with unit variance i.e., $\sigma_1 = \sigma_2 = \mathbf{I}$. The means of one Gaussian are $\mu_{11}, \mu_{12} \sim \mathcal{U}(0, 2)$ and the means of the other are $\mu_{21}, \mu_{22} \sim \mathcal{U}(0, 2)$. The mixture weight is $\pi \sim \mathcal{U}(0.50, 0.70)$. Figure 1 illustrates the mean absolute errors when (1) the variances are mis-specified at $\varepsilon_\sigma \mathbf{I}$ where $\varepsilon_\sigma \sim \mathcal{U}(1, 2)$; (2) the weights are mis-specified at $\varepsilon_\pi \sim \mathcal{U}(0, 1)$; (3) both are mis-specified. We compare EM with our proposed method that estimates the means by the minimum Wasserstein estimators. The figures show that while EM plateaus early on, our method continues to converge over training. This reaffirms that minimum Wasserstein estimators tend to be more reliable and robust under mis-specifications (Bernton et al., 2019). Despite the above desirable properties of the WS distance, the application of OT to estimating the parameters of a general DGM remains underexplored. Our work is proposed to fill in this gap.

Contributions. In this work, we introduce **OTP-DAG**, an **O**ptimal **T**ransport framework for **P**arameter Learning in **D**irected **A**cyclic **G**raphical models[1]. OTP-DAG is a flexible framework applicable to any type of variables and graphical structures. Our theoretical development renders a tractable formulation of the Wasserstein objective for models with latent variables, which is established as a generalization for the WAE model. We further provide empirical evidence demonstrating the versatility of our method on various graphical structures, where OTP-DAG is shown to successfully recover the ground-truth parameters and achieve comparable or better performance than competing methods across a range of downstream applications.

[1]Our code is published at https://github.com/isVy08/OTP.

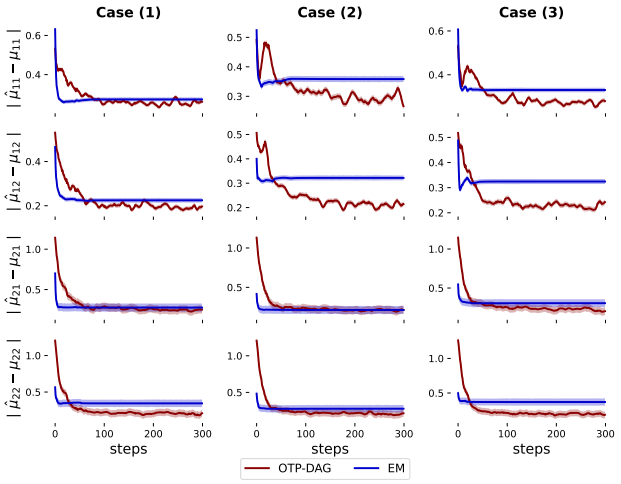

Figure 1. Visualization of mean absolute errors of the inferred means $\hat{\mu}$ and the true values μ for 300 steps, averaged over 100 simulations. μ_{ki} indicates the mean of the component k at dimension i. The red line represents **our method OTP-DAG**. The blue line represents **EM**. Three mis-specified cases are studied: **Case (1)** mis-specified variances, **Case (2)** mis-specified weights and **Case (3)** mis-specified both variances and weights.

2. Related work

Variational Inference. As part of parameter learning, both EM and VI entail an inference sub-process for posterior estimation. If the posteriors cannot be computed exactly, approximate inference is the go-to solution. In this section, we focus on variational algorithms and their computational challenges. Along this line, research efforts have concentrated on ensuring tractability of the ELBO via the mean-field assumption (Bishop & Nasrabadi, 2006) and its relaxation known as structured mean field (Saul & Jordan, 1995). Scalability has been one of the main challenges facing the early VI formulations since the optimization is done on a per-sample basis. This has triggered the development of stochastic variational inference (SVI, Hoffman et al., 2013; Hoffman & Blei, 2015; Foti et al., 2014; Johnson & Willsky, 2014; Anandkumar et al., 2012; 2014) which applies stochastic optimization to solve VI objectives.

Another line of work is collapsed VI that explicitly integrates out certain model parameters or latent variables in an analytic manner (Hensman et al., 2012; King & Lawrence, 2006; Teh et al., 2006; Lázaro-Gredilla et al., 2012). Without a closed form, one could resort to Markov chain Monte Carlo (Gelfand & Smith, 1990; Gilks et al., 1995; Hammersley, 2013), which however tends to be slow. More accurate variational posteriors also exist, namely, through hierarchical variational models (Ranganath et al., 2016), implicit posteriors (Titsias & Ruiz, 2019; Yin & Zhou, 2018; Molchanov et al., 2019; Titsias & Ruiz, 2019), normalizing flows (Kingma et al., 2016), or copulas (Tran et al., 2015).

To avoid computing ELBO analytically, one can obtain an unbiased gradient estimator using re-parameterization tricks (Ranganath et al., 2014; Xu et al., 2019). Extensions of VI to other divergence measures such as $\alpha-$divergence or $f-$divergence, also exist in Li & Turner (2016); Hernandez-Lobato et al. (2016); Wan et al. (2020). A thorough review the above approaches can be found in Ambrogioni et al. (2021, §6). In the causal inference literature, a related direction is to learn both the graphical structure and parameters of the corresponding structural equation model (Yu et al., 2019; Geffner et al., 2022). These frameworks are often limited to additive noise models while assuming no latent confounders.

Optimal Transport. Optimal transport (OT) studies the optimal transportation of mass from one distribution to another (Villani et al., 2009). Through the notion of Wasserstein distance, OT offers a geometrically meaningful distance between probability distributions, proving effectiveness in various machine learning domains (Huynh et al., 2020; Zhao et al., 2020; Nguyen et al., 2021; Wang et al., 2022; Bui et al., 2021; Nguyen et al., 2022; Vuong et al., 2023; Ye et al., 2024; Gao et al., 2024; Luong et al., 2024; Vo et al., 2024).

Particularly, there has been a surge in OT application to generative models, namely Wasserstein GANs (WGAN, Adler & Lunz, 2018; Arjovsky et al., 2017) and Wasserstein Auto-encoders (WAE, Tolstikhin et al., 2017). Underlying WAE is basically a two-node graphical model with one observed node (i.e., the data) and one latent node (i.e., often a Gaussian prior). There has also been application of OT for learning Gaussian mixture models (GMM), which are also two-node graphical models where the latent variable (i.e., the mixture weight) is categorical. Mena et al. (2020) proposes an algorithm named Sinkhorn EM using entropic OT loss which yields faster convergence rate than vanilla EM. Kolouri et al. (2018) studies the extension of the Wasserstein mean problem (Ho et al., 2017) to learn a GMM, showing that the Wasserstein energy landscape is smoother and less sensitive to the initial point than that of the negative log likelihood.

3. Preliminaries

We first introduce the notations and basic concepts used throughout the paper. We reserve bold capital letters (e.g., $\mathbf{G}$) for notations related to graphs. We use calligraphic letters (e.g., $\mathcal{X}$) for spaces, italic capital letters (e.g., X) for random variables, and lower case letters (e.g., x) for values.

Directed Graphical Models. A directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ consists of a set of nodes $\mathbf{V}$ and an edge set $\mathbf{E} \subseteq \mathbf{V}^2$ of ordered pairs of nodes with $(v, v) \notin \mathbf{E}$ for any $v \in \mathbf{V}$ (one without self-loops). For a pair of nodes i, j with $(i, j) \in \mathbf{E}$, there is an arrow pointing from i to j and we write $i \to j$. Two nodes i and j are adjacent if either $(i, j) \in \mathbf{E}$ or $(j, i) \in \mathbf{E}$. If there is an arrow from i to j then i is a parent of j and j is a child of i. A Bayesian network structure $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is a **directed acyclic graph** (DAG), in which the nodes represent random variables $X = [X_i]_{i=1}^n$ with index set $\mathbf{V} := \{1, ..., n\}$. Let PA_{X_i} denote the set of variables associated with parents of node i in $\mathbf{G}$. In this work, we tackle the classic problem of learning the parameters of a directed graph from *partially observed data*. Let $\mathbf{O} \subseteq \mathbf{V}$ and $X_{\mathbf{O}} = [X_i]_{i \in \mathbf{O}}$ be the set of observed nodes and $\mathbf{H} := \mathbf{V} \backslash \mathbf{O}$ be the set of hidden nodes. Let P_θ and P_d respectively denote the distribution induced by the graphical model and the empirical one induced by the *complete* (yet unknown) data. Given a fixed graphical structure $\mathbf{G}$ and some set of i.i.d data points, we aim to find the point estimate θ^* that best fits the observed data $X_{\mathbf{O}}$. The conventional approach is to minimize the KL divergence between the model distribution and the *empirical* data distribution over observed data i.e., $\mathrm{KL}(P_d(X_{\mathbf{O}}), P_\theta(X_{\mathbf{O}}))$, which is equivalent to maximizing the likelihood $P_\theta(X_{\mathbf{O}})$ w.r.t θ. In the presence of latent variables, the marginal likelihood, given as $P_\theta(X_{\mathbf{O}}) = \int_{X_{\mathbf{H}}} P_\theta(X) dX_{\mathbf{H}}$, is generally intractable.

Optimal transport. Let $\alpha = \sum_{j=1}^n a_j \delta_{x_j}$ be a discrete measure with weights $\boldsymbol{a}$ and particles $x_1, \cdots, x^n \in \mathcal{X}$ where $\boldsymbol{a} \in \Delta^n$, a $(n-1)-$dimensional probability simplex. Let $\beta = \sum_{j=1}^n b_j \delta_{y_j}$ be another discrete measure defined similarly. The Kantorovich (Kantorovich, 2006) formulation of the OT distance between two discrete distributions α and β is

$$W_c(\alpha, \beta) := \inf_{\mathbf{P} \in \mathbb{U}(\boldsymbol{a}, \boldsymbol{b})} \langle \mathbf{C}, \mathbf{P} \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius dot-product; $\mathbf{C} \in \mathbb{R}_+^{n \times n}$ is the cost matrix of the transport; $\mathbf{P} \in \mathbb{R}_+^{n \times n}$ is the transport matrix/plan; $\mathbb{U}(\boldsymbol{a}, \boldsymbol{b}) := \left\{ \mathbf{P} \in \mathbb{R}_+^{n \times n} : \mathbf{P}\mathbf{1}_n = \boldsymbol{a}, \mathbf{P}\mathbf{1}_n = \boldsymbol{b} \right\}$ is the transport polytope of $\boldsymbol{a}$ and $\boldsymbol{b}$; $\mathbf{1}_n$ is the $n-$dimensional column of vector of ones. For arbitrary measures, Eq. (1) can be generalized as

$$W_c(\alpha; \beta) := \inf_{\Gamma \sim \mathcal{P}(X \sim \alpha, Y \sim \beta)} \mathbb{E}_{(X,Y) \sim \Gamma} \left[c(X, Y) \right], \tag{2}$$

where the infimum is taken over the set of all joint distributions $\mathcal{P}(X \sim \alpha, Y \sim \beta)$ with marginals α and β respectively. $c : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is any measurable cost function. If $c(x, y) = D^p(x, y)$ is a distance for $p \leq 1$, W_p, the p-root of W_c, is called the p-Wasserstein distance. Finally, for a measurable map $T : \mathcal{X} \mapsto \mathcal{Y}$, $T\#\alpha$ denotes the push-forward measure of α that, for any measurable set $B \subset \mathcal{Y}$, satisfies $T\#\alpha(B) = \alpha(T^{-1}(B))$. For discrete measures, the push-forward operation is $T\#\alpha = \sum_{j=1}^{n} a_j \delta_{T(x_j)}$.

4. Optimal Transport for Learning Directed Graphical Models

We consider a DAG $\mathbf{G}(\mathbf{V}, \mathbf{E})$ over random variables $X = [X_i]_{i=1}^n$ that represents the data generative process of an underlying system. The system consists of X as the set of endogenous variables and $U = \{U_i\}_{i=1}^n$ as the set of exogenous variables representing external factors affecting the system. Associated with every X_i is an exogenous variable U_i whose values are sampled from a prior distribution $P(U)$ independently from the other exogenous variables. For the purpose of theoretical development, our framework operates on an extended graph consisting of both endogenous and exogenous nodes (See Figure 2). In the graph $\mathbf{G}$, U_i is represented by a node with no ancestors that has an outgoing arrow towards its associated endogenous variable X_i. Every distribution $P_{\theta_i}\big(X_i|\mathrm{PA}_{X_i}\big)$ can be reparameterized into a deterministic assignment

$$X_i := \psi_i\big(\mathrm{PA}_{X_i}, U_i\big), \text{ for } i = 1, ..., n.$$

The ultimate goal is to estimate $\theta = \{\theta_i\}_{i=1}^n$ as the parameters of the set of deterministic functions $\psi = \{\psi_i\}_{i=1}^n$. We will use the notation ψ_θ to emphasize this connection from now on. Given the empirical data distribution $P_d(X_\mathbf{O})$ and the model distribution $P_\theta(X_\mathbf{O})$ over the observed set $\mathbf{O}$, the optimal transport goal is to find the parameter set θ that minimizes the cost of transport between these two distributions. The Kantorovich's formulation of the problem is given by

$$W_c\big(P_d; P_\theta\big) := \inf_{\Gamma \sim \mathcal{P}(X \sim P_d, Y \sim P_\theta)} \mathbb{E}_{(X,Y)\sim\Gamma}\big[c(X, Y)\big], \tag{3}$$

where $\mathcal{P}(X \sim P_d, Y \sim P_\theta)$ is a set of all joint distributions of $\big(P_d; P_\theta\big)$; $c : \mathcal{X}_\mathbf{O} \times \mathcal{X}_\mathbf{O} \mapsto \mathcal{R}_+$ is any measurable cost function over $\mathcal{X}_\mathbf{O}$ (i.e., the product space of the spaces of observed variables) defined as $c(X_\mathbf{O}, Y_\mathbf{O}) := \sum_{i\in\mathbf{O}} c_i(X_i, Y_i)$ where c_i is a measurable cost function over a space of an observed variable.

Since P_θ is intractable due to the latent factor, the formulation in Eq. (3) cannot be directly optimized. We now propose our solution to this optimization problem (OP).

Let $P_\theta(\mathrm{PA}_{X_i}, U_i)$ denote the joint distribution of PA_{X_i} and U_i factorized according to the graphical model. Let $\mathcal{U}_i$ denote the space over random variable U_i. The key ingredient of our theoretical development is local backward mapping. For every observed node $i \in \mathbf{O}$, we define a stochastic "backward" map $\phi_i : \mathcal{X}_i \mapsto \Pi_{k\in\mathrm{PA}_{X_i}} \mathcal{X}_k \times \mathcal{U}_i$ such that $\phi_i \in \mathfrak{C}(X_i)$ where $\mathfrak{C}(X_i)$ is the constraint set given as

$$\mathfrak{C}(X_i) := \big\{\phi_i : \phi_i\#P_d(X_i) = P_\theta(\mathrm{PA}_{X_i}, U_i)\big\};$$

that is, every backward $\phi_i\#$ defines a push forward operator such that the samples from $\phi_i(X_i)$ follow the marginal distribution $P_\theta(\mathrm{PA}_{X_i}, U_i)$. Let $P_{\phi_i}(\mathrm{PA}_{X_i}, U_i) = \mathbb{E}_{X_i}\left[\phi_i(\mathrm{PA}_{X_i}, U_i|X_i)\right]$ denote the marginal distribution induced by every ϕ_i.

We will show that the OP in (3) amounts to minimizing the reconstruction error between the observed data and the data generated from P_θ. To understand how the reconstruction works, let us examine the right illustration in Figure 2. With a slight abuse of notations, for every X_i, we extend its parent set PA_{X_i} to include an exogenous variable and possibly some other endogenous variables. Given X_1 and X_3 as observed nodes, we first sample $X_1 \sim P_d(X_1), X_3 \sim P_d(X_3)$ and then construct backward maps ϕ_1, ϕ_3. The next step is to sample $\mathrm{PA}_{X_1} \sim \phi_1(\mathrm{PA}_{X_1}|X_1)$ and $\mathrm{PA}_{X_3} \sim \phi_3(\mathrm{PA}_{X_3}|X_3)$, where $\mathrm{PA}_{X_1} = \{X_2, X_4, U_1\}$ and $\mathrm{PA}_{X_3} = \{X_4, U_3\}$, which are plugged back to the model ψ_θ to obtain the reconstructions $\widetilde{X_1} = \psi_{\theta_1}(\mathrm{PA}_{X_1})$ and $\widetilde{X_3} = \psi_{\theta_3}(\mathrm{PA}_{X_3})$. We wish to learn θ such that X_1 and X_3 are reconstructed correctly. For a general graphical model, this optimization objective is formalized as

Theorem 4.1. *For every ϕ_i as defined above and fixed ψ_θ,*

$$W_c\big(P_d(X_\mathbf{O}); P_\theta(X_\mathbf{O})\big) = \inf_{\left[\phi_i \in \mathfrak{C}(X_i)\right]_{i\in\mathbf{O}}} \mathbb{E}_{X_\mathbf{O}\sim P_d, \mathrm{PA}_{X_\mathbf{O}}\sim\phi(X_\mathbf{O})}\big[c\big(X_\mathbf{O}, \psi_\theta(\mathrm{PA}_{X_\mathbf{O}})\big)\big], \tag{4}$$

where $\mathrm{PA}_{X_\mathbf{O}} := \big[[X_{ij}]_{j\in\mathrm{PA}_{X_i}}\big]_{i\in\mathbf{O}}$.

Proof. See Appendix A.

By Theorem 4.1, the estimation of OT cost is reduced to finding the optimal conditional $\phi(\mathrm{PA}_{X_i}|X_i)$ for every observed node X_i such that the "backward" marginal $P_\phi(\mathrm{PA}_{X_i})$ is identical to the the "forward" marginal $P_\theta(\mathrm{PA}_{X_i})$. While Theorem 4.1 set ups a tractable form for our optimization solution, our OP is constrained, where every backward function ϕ_i must satisfy its push-forward condition defined by $\mathfrak{C}$. In the above example, the backward maps ϕ_i and ϕ_3 must be constructed such that $\phi_1\#P_d(X_1) = P_\theta(X_2, X_4, U_1)$ and $\phi_3\#P_d(X_3) = P_\theta(X_4, U_3)$. We propose to relax the constraints by adding a penalty to the objective (4).

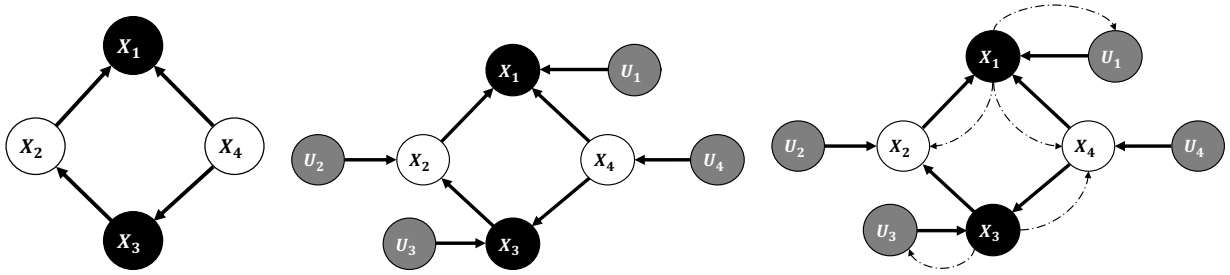

Figure 2. **(Left)** A DAG represents a system of 4 endogenous variables where X_1, X_3 are observed (black-shaded) and X_2, X_4 are hidden variables (non-shaded). **(Middle)** The extended DAG includes an additional set of independent exogenous variables U_1, U_2, U_3, U_4 (grey-shaded) acting on each endogenous variable. $U_1, U_2, U_3, U_4 \sim P(U)$ where $P(U)$ is a prior product distribution. **(Right)** Visualization of our backward-forward algorithm, where the dashed arcs represent the backward maps involved in optimization.

The **final optimization objective** is therefore given as

$$\inf_{\theta} \inf_{\phi} \mathbb{E}_{X_{\mathbf{O}} \sim P_d, \mathrm{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} \big[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\big] + \eta\, D\big(P_\phi, P_\theta\big) \tag{5}$$

where D is any arbitrary divergence measure and $\eta > 0$ is a trade-off hyper-parameter. $D\big(P_\phi, P_\theta\big)$ is a short-hand for divergence between all pairs of backward and forward marginals over the parents of the observed nodes.

Remark. Eq. (5) renders an optimization-based approach in which we leverage reparameterization and amortization (Amos, 2022) for solving it efficiently via stochastic gradient descent. This theoretical result provides our OTP-DAG with two interesting properties: (1) all model parameters are optimized simultaneously within a single framework whether the variables are continuous or discrete, and (2) the computational process can be automated without the need for analytic lower bounds (as in EM and traditional VI), specific graphical structures (as in mean-field VI), or priors over variational distributions on latent variables (as in hierarchical VI). The flexibility our method exhibits is akin to auto-encoding models, and OTP-DAG in fact serves as an extension of WAE for learning general directed graphical models. Our formulation thus inherits a desirable characteristic from WAE, which specifically helps mitigate the posterior collapse issue notoriously occurring to VAE. Appendix B explains this behavior in more detail. Particularly in Section 5.3, we will empirically show that OTP-DAG effectively alleviates the codebook collapse issue in discrete representation learning. Algorithm 1 provides the pseudo-code for OTP-DAG learning procedure.

Algorithm 1 OTP-DAG Algorithm

Input: Directed graph $\mathbf{G}$ with observed nodes $\mathbf{O}$, noise distribution $P(U)$, regularization coefficient η, reconstruction cost function c, and divergence measure D.
Output: Point estimate θ.
Initialize a set of deterministic assignments $\psi_\theta = \{\psi_{\theta_i}\}_{i \in \mathbf{O}}$ where $X_i := \psi_{\theta_i}(\mathrm{PA}_{X_i}, U_i)$ and $U_i \sim P(U)$;
Initialize the stochastic maps $\phi = \{\phi_i(X_i)\}_{i \in \mathbf{O}}$;
while (ϕ, θ) *not converged* **do**
for $i \in \mathbf{O}$,

- Sample batch $X_i^B = \{x_i^1, ..., x_i^B\}$;
- Sample $\widetilde{\mathrm{PA}}_{X_i^B}$ from $\phi_i(X_i^B)$;
- Sampling U_i from the prior $P(U)$;
- Evaluate $\widetilde{X}_i^B = \psi_{\theta_i}(\widetilde{\mathrm{PA}}_{X_i^B}, U_i)$.

Update ϕ and θ alternately by descending

$$\frac{1}{B} \sum_{b=1}^{B} \sum_{i \in \mathbf{O}} c\big(x_i^b, \widetilde{x}_i^b\big) + \eta\, D\big[P_{\phi_i}(\mathrm{PA}_{X_i^B}), P_\theta(\mathrm{PA}_{X_i^B})\big]$$

end while

5. Applications

In this section, we illustrate the practical applications of the OTP-DAG algorithm. We consider various directed probabilistic models with different types of latent variables (continuous and discrete) and for different types of data (texts, images, and time series). In all tables, we report the average results over 5 random initializations and the best/second-best ones are bold/underlined. ↑, ↓ indicate higher/lower performance is better, respectively.

Baselines. We compare OTP-DAG with two groups of parameter learning methods towards the two extremes: (1) EM and SVI where analytic derivation is required; (2) variational auto-encoding frameworks where black-box optimization is permissible. We leave the discussion of the formulation and technicalities in Appendix C.

Experimental setup. We begin with (1) Latent Dirichlet Allocation (Blei et al., 2003), a popular task of topic modeling where traditional methods like EM or SVI can

solve. We then consider learning a (2) Hidden Markov Model (HMM), which remains fairly challenging, where existing optimization/inference algorithms (e.g., Baum-Welch algorithm) are often too computationally costly to be used in practice. We conclude with a more challenging setting: (3) Discrete Representation Learning (Discrete RepL) that cannot simply be solved by EM or MAP (maximum a posteriori). It in fact invokes deep generative modeling via a pioneering development called Vector Quantization Variational Auto-Encoder (VQ-VAE, Van Den Oord et al., 2017). We attempt to apply OTP-DAG for learning discrete representations by grounding it into a parameter learning problem. We note again that for standard (continuous) representation learning, OTP-DAG reduces to WAE (Tolstikhin et al., 2017), which readers can refer to for extensive empirical evidence. Identifiability of the parameters in latent variable models is of critical concern. In the task of recovering the true parameters, we experiment with the LDA setting and Poisson HMM where the parameters are identifiable up to permutations (Teicher, 1960; Wang, 2019), and we resolve the ambiguity by sorting out the estimations.

Figure 3 illustrates the empirical DAG structures of 3 applications. Unlike the standard visualization where the parameters are considered hidden nodes, our graph separates model parameters from latent variables and only illustrates random variables and their dependencies (except the special setting of discrete representation learning). We also omit the exogenous variables associated with the hidden nodes for visibility, since only those acting on the observed nodes are relevant for computation. There is also a noticeable difference between Figures 3 and 2: the empirical version does not require learning the backward maps for the exogenous variables. It is observed across our experiments that sampling the noise from an appropriate prior distribution suffices to yield accurate estimation, which is in fact beneficial in that the training time can be greatly reduced.

Remark. In the following, we show that in the simulated settings where the models are well-specified, OTP-DAG performs equally well as the baseline methods, while exhibits superior efficiency over EM on such a complex graph as HMM. Furthermore, OTP-DAG is shown to achieve better performance on real-world downstream tasks, which substantiates the robustness of the minimum Wasserstein estimators in practical settings. Finally, throughout the experiments, we also aim to demonstrate the versatility of OTP-DAG where our method can be harnessed for a wide range of purposes in a single learning procedure.

5.1. Latent Dirichlet Allocation

Let us consider a corpus $\mathcal{D}$ of M independent documents where each document is a sequence of N words denoted by $W_{1:N} = (W_1, W_2, \cdots, W_N)$. Documents are represented as random mixtures over K latent topics, each of which is characterized by a distribution over words. Let V be the size of a vocabulary indexed by $\{1, \cdots, V\}$. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) dictates the following generative process for every document in the corpus:

1. Sample $\theta \sim \text{Dir}(\alpha)$ with $\alpha < 1$,
2. Sample $\gamma_k \sim \text{Dir}(\beta)$ where $k \in \{1, \cdots, K\}$,
3. For each of the word positions $n \in \{1, \cdots, N\}$,
   - Sample a topic $z_n \sim \text{Multi-nominal}(\theta)$,
   - Sample a word $w_n \sim \text{Multi-nominal}(\gamma_{kn})$,

where Dir(.) is a Dirichlet distribution. θ is a $K-$dimensional vector that lies in the $(K-1)-$simplex and γ_k is a $V-$dimensional vector represents the word distribution corresponding to topic k. In the standard model, α, β, K are hyper-parameters and θ, γ are learnable parameters. Throughout the experiments, the number of topics K is assumed known and fixed.

Parameter estimation. To test whether OTP-DAG can recover the true parameters, we generate synthetic data in the setting: the word probabilities are parameterized by a $K \times V$ matrix γ where $\gamma_{kn} := P(W_n = 1 | Z_n = 1)$; γ is now a fixed quantity to be estimated. We set $\alpha = 1/K$ uniformly and generate small datasets for different number of topics K and sample size N. Following (Griffiths & Steyvers, 2004), for every topic k, the word distribution γ_k can be represented as a square grid where each cell, corresponding to a word, is assigned an integer value of either 0 and 1, indicating whether a certain word is allocated to the kth topic or not. As a result, each topic is associated with a specific pattern. For simplicity, we represent topics using horizontal or vertical patterns (See Figure 4). According to the above generative model, we sample data w.r.t 3 sets of configuration triplets $\{K, M, N\}$. We compare OTP-DAG with Batch EM and SVI and Prod LDA - a variational auto-encoding topic model (Srivastava & Sutton, 2017).

Table 1 reports the fidelity of the estimation of γ. OTP-DAG consistently achieves high-quality estimates by both Hellinger and Wasserstein distances. It is not surprising that the baselines are superior by the KL metric, as it is what they implicitly minimize. While it is inconclusive from the numerical estimations, the qualitative results complete the story. Figure 4 illustrates the distributions of individual words to the topics from each method after 300 training epochs. OTP-DAG successfully recovers the true patterns and as well as EM and SVI, while Prod LDA mis-detects several patterns, despite the competitive numerical results. More qualitative examples for the other settings are presented in Figures 7 and 8 where OTP-DAG is shown to recover almost all true patterns.

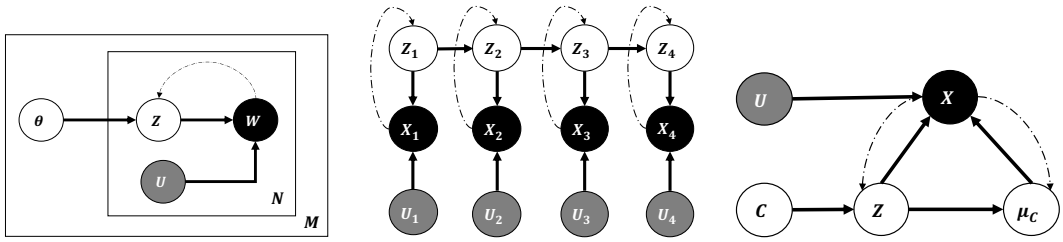

Figure 3. Empirical structures of **(left)** latent Dirichlet allocation model (in plate notation), **(middle)** standard hidden Markov model, and **(right)** discrete representation learning.

Table 1. Fidelity of estimates of the topic-word distribution γ across 3 settings. Fidelity is measured by KL divergence, Hellinger (HL) (Hellinger, 1909) and Wasserstein distance with the true γ.

Metric ↓	K	M	N	**OTP-DAG** (Ours)	**Batch EM**	**SVI**	**Prod LDA**
HL	10	1,000	100	**2.327** $\pm$ **0.009**	2.807 $\pm$ 0.189	2.712 $\pm$ 0.087	2.353 $\pm$ 0.012
KL	10	1,000	100	1.701 $\pm$ 0.005	1.634 $\pm$ 0.022	**1.602** $\pm$ **0.014**	1.627 $\pm$ 0.027
WS	10	1,000	100	**0.027** $\pm$ **0.004**	0.058 $\pm$ 0.000	0.059 $\pm$ 0.000	0.052 $\pm$ 0.001
HL	20	5,000	200	3.800 $\pm$ 0.058	4.256 $\pm$ 0.084	4.259 $\pm$ 0.096	**3.700** $\pm$ **0.012**
KL	20	5,000	200	2.652 $\pm$ 0.080	**2.304** $\pm$ **0.004**	2.305 $\pm$ 0.003	2.316 $\pm$ 0.026
WS	20	5,000	200	**0.010** $\pm$ **0.001**	0.022 $\pm$ 0.000	0.022 $\pm$ 0.001	0.018 $\pm$ 0.000
HL	30	10,000	300	4.740 $\pm$ 0.029	5.262 $\pm$ 0.077	5.245 $\pm$ 0.035	**4.723** $\pm$ **0.017**
KL	30	10,000	300	2.959 $\pm$ 0.015	**2.708** $\pm$ **0.002**	2.709 $\pm$ 0.001	2.746 $\pm$ 0.034
WS	30	10,000	300	**0.005** $\pm$ **0.001**	0.012 $\pm$ 0.000	0.012 $\pm$ 0.000	0.009 $\pm$ 0.000

Figure 4. Topic-word distributions inferred by each method from the 1st set of synthetic data after 300 training epochs.

Topic Inference. We now demonstrate the effectiveness of OTP-DAG on downstream applications[2]. We here use OTP-DAG to infer the topics of 3 real-world datasets: 20 News Group[3], BBC News (Greene & Cunningham, 2006) and DBLP[4]. We revert to the original generative process where the topic-word distributions follows a Dirichlet distribution parameterized by the concentration parameters β, instead of having γ as a fixed quantity. β is now initialized as a matrix of real values $\left(\beta \in \mathbb{R}^{K \times V}\right)$ representing the log concentration values.

For every topic k, we select top 10 most related words according to γ_k to represent it. Table 2 reports the quality of the inferred topics, which is evaluated via the diversity and coherence of the selected words. Diversity refers to the proportion of unique words, whereas Coherence is measured with normalized pointwise mutual information (Aletras & Stevenson, 2013), reflecting the extent to which the words in a topic are associated with a common theme. There exists a trade-off between Diversity and Coherence: words that are excessively diverse greatly reduce coherence, while a set of many duplicated words yields higher coherence yet harms diversity. A well-performing topic model would strike a good balance between these metrics (Zhao et al., 2021). If we consider two metrics comprehensively, our method consistently achieves better performance across different settings. Qualitative results of the inferred topics can be found in Table 5.

5.2. Hidden Markov Models

This application deals with time-series data following a **Poisson hidden Markov model**. Given a time series of T

[2] https://github.com/MIND-Lab/OCTIS. We use OCTIS to standardize evaluation for all models on the topic inference task. Note that the computation of topic coherence score in OCTIS is different than in Srivastava & Sutton (2017).

[3] http://qwone.com/~jason/20Newsgroups/.

[4] https://github.com/shiruipan/TriDNR/.

Table 2. Coherence and Diversity of the inferred topics for the 3 real-world datasets ($K = 10$).

Metric (%) ↑	**OTP-DAG** (Ours)	**Batch EM**	**SVI**	**Prod LDA**
		20 News Group		
Coherence	**10.45 ± 0.56**	6.71 ± 0.16	5.90 ± 0.51	4.78 ± 2.64
Diversity	92.00 ± 2.65	72.33 ± 1.15	85.33 ± 5.51	**92.67 ± 4.51**
		BBC News		
Coherence	**9.12 ± 0.81**	8.67 ± 0.62	7.84 ± 0.49	2.17 ± 2.36
Diversity	87.67 ± 2.65	86.00 ± 1.00	**92.33 ± 2.31**	87.67 ± 3.79
		DBLP		
Coherence	**7.66 ± 0.44**	4.52 ± 0.53	1.47 ± 0.39	2.91 ± 1.70
Diversity	97.33 ± 1.53	81.33 ± 1.15	92.67 ± 2.52	**98.67 ± 1.53**

steps, the task is to segment the data stream into 4 different states, each of which follows a Poisson distribution with rate λ_k sampled from a Uniform hyper-prior. The distributions and the observation at each step t are given as

$$P(X_t|Z_t = k) = \text{Poi}(X_t|\lambda_k), \quad \text{for } k = 1, \cdots, 4,$$

where $\lambda_1 \sim U(10, 20)$, $\lambda_2 \sim U(30, 40)$, $\lambda_3 \sim U(50, 60)$ and $\lambda_4 \sim U(80, 90)$. We further impose a uniform prior over the initial state. The Markov chain stays in the current state with probability p and otherwise transitions to one of the other three states uniformly at random. The transition distribution is given as

$$z_1 \sim \text{Cat}\left(\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right),$$
$$z_t|z_{t-1} \sim \text{Cat}\left(\left\{\begin{array}{cc} \pi & \text{if } Z_t = Z_{t-1} \\ \frac{1-\pi}{4-1} & \text{otherwise} \end{array}\right\}\right)$$

We randomly generate 200 datasets of $50,000$ observations each. For each dataset, we train the models for 50 epochs with learning rate of 0.05 at 5 different initializations. We would like to learn the concentration parameters $\lambda_{1:4}$ through which segmentation can be realized, assuming that the number of states is known. The other experimental configuration is reported in Appendix C.2.

Table 3 reports mean error of the estimates of the parameters along with runtime of OTP-DAG and EM. As the absolute values can be misleading, we report the errors in relative terms, where we apply min-max normalization to scale the λ values to $[0, 1]$. Figure 9 additionally visualizes the distribution of the estimations from OTP-DAG and EM to show the alignment with the generative uniform distributions.

5.3. Learning Discrete Representations

Learning latent discrete representations of data is an important problem, which can be useful for planning and symbolic reasoning tasks. Viewing discrete representation learning as a parameter learning problem, we endow it with a probabilistic generative process as illustrated in Figure 5. The

Table 3. Estimates of the concentration parameters $\lambda_{1:4}$ of the Poisson HMM, measured by mean absolute error with the true values.

Method	**OTP-DAG** (Ours)	**EM**
λ_1	0.040 ± 0.129	**0.022 ± 0.042**
λ_2	**0.079 ± 0.088**	0.088 ± 0.105
λ_3	**0.148 ± 0.119**	0.166 ± 0.171
λ_4	**0.084 ± 0.099**	0.101 ± 0.008
Runtime (50 steps)	≈ **3 mins**	≈ 20 mins

problem deals with a latent space $\mathcal{C} \in \mathbb{R}^{K \times D}$ composed of K discrete latent sub-spaces of D dimensionality. The probability a data point belongs to a discrete sub-space $c \in \{1, \cdots, K\}$ follows a $K-$way categorical distribution $\pi = [\pi_1, \cdots, \pi_K]$. In the language of VQ-VAE, each c is referred to as a *codeword* and the set of codewords is called a *codebook*. Let $Z \in \mathbb{R}^D$ denote the latent variable in a sub-space. On each sub-space, we impose a Gaussian distribution parameterized by μ_c, Σ_c where Σ_c is diagonal. The generative process is as follows:

1. Sample $c \sim \text{Cat}(\pi)$ and $z \sim \mathcal{N}(\mu_c, \Sigma_c)$
2. Quantize $\mu_c = Q(z)$,
3. Generate $x = \psi_\theta(z, \mu_c)$,

where ψ is a highly non-convex function with unknown parameters θ and often parameterized by a deep neural network. Q refers to the quantization of z to μ_c defined as $\mu_c = Q(z)$ where $c = \text{argmin}_c\, d_z(z; \mu_c)$ and $d_z = \sqrt{(z - \mu_c)^T \Sigma_c^{-1} (z - \mu_c)}$ is the Mahalanobis distance. The goal is to learn the parameter set $\{\pi, \mu, \Sigma, \theta\}$ with $\mu = [\mu_k]_{k=1}^K, \Sigma = [\Sigma_k]_{k=1}^K$ such that the learned representation captures the key properties of the data. Following VQ-VAE, our practical implementation considers Z as an $M-$component latent embedding.

We experiment with images in this application and compare OTP-DAG with VQ-VAE on CIFAR10[5], MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011) and CELEBA datasets (Liu et al., 2015). Since the true parameters are unknown, we assess how well the latent space characterizes the input data through the quality of the reconstruction of the original images. Table 4 reports our superior performance in preserving high-quality information of the input images. VQ-VAE suffers from poorer performance mainly due to *codebook collapse* (Yu et al., 2021) where most of latent vectors are quantized to limited discrete codewords. Meanwhile, our framework allows to control the number of latent representations, ensuring all codewords are utilized. In Appendix C.3, we detail the formulation of our method

[5] https://www.cs.toronto.edu/~kriz/cifar.html.

Table 4. Quality of the image reconstructions from the vector quantized models ($K = 512$).

Dataset	Method	Latent Size	SSIM ↑	PSNR ↑	LPIPS ↓	rFID ↓	Perplexity ↑
CIFAR10	**VQ-VAE**	8×8	0.70	23.14	0.35	77.3	69.8
	OTP-DAG (Ours)	8×8	**0.80**	**25.40**	**0.23**	**56.5**	**498.6**
MNIST	**VQ-VAE**	8×8	**0.98**	33.37	0.02	4.8	47.2
	OTP-DAG (Ours)	8×8	**0.98**	**33.62**	**0.01**	**3.3**	**474.6**
SVHN	**VQ-VAE**	8×8	0.88	26.94	0.17	38.5	114.6
	OTP-DAG (Ours)	8×8	**0.94**	**32.56**	**0.08**	**25.2**	**462.8**
CELEBA	**VQ-VAE**	16×16	0.82	27.48	0.19	19.4	48.9
	OTP-DAG (Ours)	16×16	**0.88**	**29.77**	**0.11**	**13.1**	**487.5**

and provide qualitative examples. We also showcase therein our competitive performance against a recent advance called SQ-VAE (Takida et al., 2022) without introducing any additional complexity.

6. Discussion and Conclusion

Discussion. The key message across our experiments is that OTP-DAG is a scalable and versatile framework readily applicable to learning any directed graphs with latent variables. Similar to amortized VI, on one hand, our method employs amortized optimization and assumes one can sample from the priors or more generally, the model marginals over latent parents. OTP-DAG requires continuous relaxation through reparameterization of the underlying model distribution to ensure the gradients can be back-propagated effectively. Note that this specification is not unique to OTP-DAG: VAE also relies on the reparameterization trick to compute the gradients w.r.t the variational parameters. For discrete distributions and for non-reparameterizable continuous ones (e.g., Gamma distribution), the reparameterization trick cannot be easily applied. To this end, a proposal on *Generalized Reparameterization Gradient* (Ruiz et al., 2016) can be a viable solution.

On the other hand, different from VI, our global OT cost minimization is achieved by characterizing local densities through the backward maps from the observed nodes to their parents. This localization strategy makes it easier to find a good approximation compared to VI, where the variational distribution is defined over all hidden variables and should ideally characterize the entire global dependencies in the graph. To model the backward distributions, we utilize the expressive power of deep neural networks. Based on the universal approximation theorem (Hornik et al., 1989), the gap between the backward and forward marginals can be assumed to be smaller than an arbitrary constant ϵ given enough data, network complexity, and training time.

Limitations. Theoretically, our algorithm can scale up to more complex graphs since we make no assumptions about the graphical structure. Our algorithm remains applicable to different graph sizes, where the computation is localized to the dependencies between an observed node and its (direct) parents. However, larger graphs indeed induce more operations where ancestral sampling to evaluate the model marginals over the related parent nodes can be computational expensive. The increased complexity has little impact on our evaluation of reconstruction loss, which only involves forward operations. However, an immediate trade-off arises as it introduces additional computational complexity to the optimization of the divergence measures. Fortunately our framework allows D to be chosen flexibly depending on applications. Here we analyze some promising candidates. A popular option is to choose D as the Jensen–Shannon divergence and estimate it with GAN-based training (Goodfellow et al., 2020). However, this choice is clearly inappropriate as it necessitates training additional discriminators, not to mention that GANs are known for their instability. Wasserstein (WS) distance and maximum mean discrepancy (MMD) are two other candidates that can be estimated with empirical samples. While the exact computation of WS has high complexity in high-dimensional space, efficient and high-quality approximations exist such as Sinkhorn divergences (Cuturi, 2013) or Sliced WS distance (Bonneel et al., 2015). MMD is also practically viable, whose sample complexity does not depend on the dimension. Furthermore, there is a kernel-based closed form to compute an unbiased estimator with reasonable choices of the kernel (Gretton et al., 2012).

Future research. The proposed algorithm lays the cornerstone for an exciting paradigm shift in the realm of graphical learning and inference. Looking ahead, this fresh perspective unlocks a wealth of promising avenues for future application of OTP-DAG to large-scale inference problems or other learning tasks such as for undirected graphical models, or structural learning where edge existence and directionality can be parameterized as part of the model parameters.

Acknowledgments

Trung Le and Dinh Phung were supported by ARC DP23 grant DP230101176 and by the Air Force Office of Scientific Research under award number FA2386-23-1-4044. This does not imply endorsement by the funding agency of the research findings or conclusions. Any errors or misinterpretations in this paper are the sole responsibility of the authors.

Impact Statement

This work introduces an application of machine learning to effectively address a class of statistical estimation problems in a scalable manner. While we are currently unaware of any potential negative societal impacts of our work, machine learning frequently yields unintended consequences in various domains, necessitating thorough consideration of societal advantages and drawbacks when implementing the proposed method in real-world scenarios.

References

Adler, J. and Lunz, S. Banach wasserstein gan. *Advances in neural information processing systems*, 31, 2018.

Aletras, N. and Stevenson, M. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on computational semantics (IWCS 2013)–Long Papers*, pp. 13–22, 2013.

Ambrogioni, L., Güçlü, U., Güçlütürk, Y., Hinne, M., Van Gerven, M. A., and Maris, E. Wasserstein variational inference. *Advances in Neural Information Processing Systems*, 2018-December(NeurIPS):2473–2482, 2018. ISSN 10495258.

Ambrogioni, L., Lin, K., Fertig, E., Vikram, S., Hinne, M., Moore, D., and van Gerven, M. Automatic structured variational inference. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 676–684. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/ambrogioni21a.html.

Amos, B. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.

Anandkumar, A., Hsu, D., and Kakade, S. M. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pp. 33–1. JMLR Workshop and Conference Proceedings, 2012.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Bassetti, F., Bodini, A., and Regazzini, E. On minimum kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302, 2006.

Beal, M. J. and Ghahramani, Z. Variational bayesian learning of directed graphical models with hidden variables. 2006.

Bernton, E., Jacob, P. E., Gerber, M., and Robert, C. P. On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4): 657–676, 2019.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

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

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

Bui, A. T., Le, T., Tran, Q. H., Zhao, H., and Phung, D. A unified wasserstein distributional robustness framework for adversarial training. In *International Conference on Learning Representations*, 2021.

Cappé, O. and Moulines, E. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

Cuturi, M. Lightspeed computation of optimal transportation distances. *Advances in Neural Information Processing Systems*, 26(2):2292–2300, 2013.

Dai, B., Wang, Z., and Wipf, D. The usual suspects? reassessing blame for vae posterior collapse. In *International conference on machine learning*, pp. 2313–2322. PMLR, 2020.

Delyon, B., Lavielle, M., and Moulines, E. Convergence of a stochastic approximation version of the em algorithm. *Annals of statistics*, pp. 94–128, 1999.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum Likelihood from Incomplete Data Via the EM Algorithm . *Journal of the Royal Statistical Society:*

Series B (Methodological), 39(1):1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Foti, N., Xu, J., Laird, D., and Fox, E. Stochastic variational inference for hidden markov models. *Advances in neural information processing systems*, 27, 2014.

Gao, J., Zhao, H., Guo, D., and Zha, H. Distribution alignment optimization through neural collapse for long-tailed classification. In *International Conference on Machine Learning*, 2024.

Geffner, T., Antoran, J., Foster, A., Gong, W., Ma, C., Kiciman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Gelfand, A. E. and Smith, A. F. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. *Markov chain Monte Carlo in practice*. CRC press, 1995.

Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Greene, D. and Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pp. 377–384. ACM Press, 2006.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Griffiths, T. L. and Steyvers, M. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101 (suppl_1):5228–5235, 2004.

Hammersley, J. *Monte carlo methods*. Springer Science & Business Media, 2013.

Hellinger, E. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909.

Hensman, J., Rattray, M., and Lawrence, N. Fast variational inference in the conjugate exponential family. *Advances in neural information processing systems*, 25, 2012.

Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T., Hernández-Lobato, D., and Turner, R. Black-box alpha divergence minimization. In *International conference on machine learning*, pp. 1511–1520. PMLR, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, N., Nguyen, X., Yurochkin, M., Bui, H. H., Huynh, V., and Phung, D. Multilevel clustering via wasserstein means. In *International conference on machine learning*, pp. 1501–1509. PMLR, 2017.

Hoffman, M. D. and Blei, D. M. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pp. 361–369, 2015.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Huynh, V., Zhao, H., and Phung, D. OTLDA: A geometry-aware optimal transport approach for topic modeling. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18573–18582, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Johnson, M. and Willsky, A. Stochastic variational inference for bayesian time series models. In *International Conference on Machine Learning*, pp. 1854–1862. PMLR, 2014.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

Kantorovich, L. V. On a problem of monge. *Journal of Mathematical Sciences*, 133(4):1383–1383, 2006.

King, N. J. and Lawrence, N. D. Fast variational inference for gaussian process models through kl-correction. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pp. 270–281. Springer, 2006.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Kolouri, S., Rohde, G. K., and Hoffmann, H. Sliced wasserstein distance for learning gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.

Lázaro-Gredilla, M., Van Vaerenbergh, S., and Lawrence, N. D. Overlapping mixtures of gaussian processes for the data association problem. *Pattern recognition*, 45(4): 1386–1395, 2012.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, Y. and Turner, R. E. Rényi divergence variational inference. *Advances in neural information processing systems*, 29, 2016.

Liang, P. and Klein, D. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pp. 611–619, 2009.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Luong, M., Nguyen, K., Ho, N., Haf, R., Phung, D., and Qu, L. Revisiting deep audio-text retrieval through the lens of transportation. *arXiv preprint arXiv:2405.10084*, 2024.

MacKay, D. J. Choice of basis for laplace approximation. *Machine learning*, 33:77–86, 1998.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Mena, G., Nejatbakhsh, A., Varol, E., and Niles-Weed, J. Sinkhorn em: an expectation-maximization algorithm based on entropic optimal transport. *arXiv preprint arXiv:2006.16548*, 2020.

Molchanov, D., Kharitonov, V., Sobolev, A., and Vetrov, D. Doubly semi-implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2593–2602. PMLR, 2019.

Neal, R. M. and Hinton, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pp. 355–368, 1998.

Neath, R. C. et al. On convergence properties of the monte carlo em algorithm. *Advances in modern statistical theory and applications: a Festschrift in Honor of Morris L. Eaton*, pp. 43–62, 2013.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, Spain, 2011.

Nguyen, T., Le, T., Zhao, H., Tran, Q. H., Nguyen, T., and Phung, D. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *Uncertainty in Artificial Intelligence*, pp. 225–235, 2021.

Nguyen, T., Nguyen, V., Le, T., Zhao, H., Tran, Q. H., and Phung, D. Cycle class consistency with distributional optimal transport and knowledge distillation for unsupervised domain adaptation. In *Uncertainty in Artificial Intelligence*, pp. 1519–1529. PMLR, 2022.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22:1–64, 2021. ISSN 15337928.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. Image transformer. In *International conference on machine learning*, pp. 4055–4064. PMLR, 2018.

Peyré, G., Cuturi, M., et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, (2017-86), 2017.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International conference on machine learning*, pp. 324–333. PMLR, 2016.

Ruiz, F. R., AUEB, T. R., Blei, D., et al. The generalized reparameterization gradient. *Advances in neural information processing systems*, 29, 2016.

Santambrogio, F. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

Saul, L. and Jordan, M. Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8, 1995.

Srivastava, A. and Sutton, C. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

Takida, Y., Shibuya, T., Liao, W., Lai, C.-H., Ohmura, J., Uesaka, T., Murata, N., Takahashi, S., Kumakura, T., and Mitsufuji, Y. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. In *International Conference on Machine Learning*, pp. 20987–21012. PMLR, 2022.

Teh, Y., Newman, D., and Welling, M. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. *Advances in neural information processing systems*, 19, 2006.

Teicher, H. On the mixture of distributions. *The Annals of Mathematical Statistics*, 31(1):55–73, 1960.

Titsias, M. K. and Ruiz, F. Unbiased implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 167–176. PMLR, 2019.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Tran, D., Blei, D., and Airoldi, E. M. Copula variational inference. *Advances in neural information processing systems*, 28, 2015.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Villani, C. et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Vo, V., Zhao, H., Le, T., Bonilla, E. V., and Phung, D. Optimal transport for structure learning under missing data. In *International Conference on Machine Learning*, 2024.

Vuong, T.-L., Le, T., Zhao, H., Zheng, C., Harandi, M., Cai, J., and Phung, D. Vector quantized wasserstein auto-encoder. *arXiv preprint arXiv:2302.05917*, 2023.

Wan, N., Li, D., and Hovakimyan, N. F-divergence variational inference. *Advances in neural information processing systems*, 33:17370–17379, 2020.

Wang, D., Guo, D., Zhao, H., Zheng, H., Tanwisuth, K., Chen, B., Zhou, M., et al. Representing mixtures of word embeddings with mixtures of topic embeddings. In *International Conference on Learning Representations*, 2022.

Wang, Y. Convergence rates of latent topic models under relaxed identifiability conditions. 2019.

Wei, G. C. and Tanner, M. A. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

Xu, M., Quiroz, M., Kohn, R., and Sisson, S. A. Variance reduction properties of the reparameterization trick. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2711–2720. PMLR, 2019.

Ye, H., Fan, W., Song, X., Zheng, S., Zhao, H., dan Guo, D., and Chang, Y. Ptarl: Prototype-based tabular representation learning via space calibration. In *International Conference on Learning Representations*, 2024.

Yin, M. and Zhou, M. Semi-implicit variational inference. In *International Conference on Machine Learning*, pp. 5660–5669. PMLR, 2018.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Yu, Y., Chen, J., Gao, T., and Yu, M. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pp. 7154–7163. PMLR, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhao, H., Phung, D., Huynh, V., Le, T., and Buntine, W. Neural topic model via optimal transport. In *International Conference on Learning Representations*, 2020.

Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., and Buntine, W. Topic modelling meets deep neural networks: a survey. In *International Joint Conference on Artificial Intelligence 2021*, pp. 4713–4720, 2021.

A. Proof

We now present the proof of Theorem 4.1 which is the key theorem in our paper.

Theorem 4.1. *For every ϕ_i as defined above and fixed ψ_θ,*

$$W_c\big(P_d(X_{\mathbf{O}}); P_\theta(X_{\mathbf{O}})\big) = \inf_{\left[\phi_i \in \mathfrak{C}(X_i)\right]_{i\in\mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}})}\left[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right],$$

where $\mathrm{PA}_{X_{\mathbf{O}}} := \left[[X_{ij}]_{j\in\mathrm{PA}_{X_i}}\right]_{i\in\mathbf{O}}$.

Proof. Let $\Gamma \in \mathcal{P}(P_d(X_{\mathbf{O}}), P_\theta(X_{\mathbf{O}}))$ be the optimal joint distribution over $P_d(X_{\mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$ of the corresponding Wasserstein distance. We consider three distributions: $P_d(X_{\mathbf{O}})$ over $A = \prod_{i\in\mathbf{O}} \mathcal{X}_i$, $P_\theta(X_{\mathbf{O}}))$ over $C = \prod_{i\in\mathbf{O}} \mathcal{X}_i$, and $P_\theta(\mathrm{PA}_{X_{\mathbf{O}}}) = P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ over $B = \prod_{i\in\mathbf{O}}\prod_{k\in\mathrm{PA}_{X_i}} \mathcal{X}_k$. Here we note that the last distribution $P_\theta(\mathrm{PA}_{X_{\mathbf{O}}}) = P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ is the model distribution over the parent nodes of the observed nodes.

It is evident that $\Gamma \in \mathcal{P}(P_d(X_{\mathbf{O}}), P_\theta(X_{\mathbf{O}}))$ is a joint distribution over $P_d(X_{\mathbf{O}})$ and$P_\theta(X_{\mathbf{O}})$; let $\beta = (id, \psi_\theta)\#P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ be a deterministic coupling or joint distribution over $P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$. Using the gluing lemma (see Lemma 5.5 in (Santambrogio, 2015)), there exists a joint distribution α over $A \times B \times C$ such that $\alpha_{AC} = (\pi_A, \pi_C)\#\alpha = \Gamma$ and $\alpha_{BC} = (\pi_B, \pi_C)\#\alpha = \beta$ where π is the projection operation. Let us denote $\gamma = (\pi_A, \pi_B)\#\alpha$ as a joint distribution over $P_d(X_{\mathbf{O}})$ and $P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$.

Given $i \in \mathbf{O}$, we denote γ_i as the projection of γ over $\mathcal{X}_i$ and $\prod_{k\in\mathrm{PA}_{X_i}} \mathcal{X}_k$. We further denote $\phi_i(X_i) = \gamma_i(\cdot \mid X_i)$ as a stochastic map from $\mathcal{X}_i$ to $\prod_{k\in\mathrm{PA}_{X_i}} \mathcal{X}_k$. It is worth noting that because γ_i is a joint distribution over $P_d(X_i)$ and $P_\theta(\mathrm{PA}_{X_i})$, $\phi_i \in \mathfrak{C}(X_i)$.

$$\begin{aligned}
W_c\left(P_d\left(X_{\mathbf{O}}\right), P_\theta\left(X_{\mathbf{O}}\right)\right) &= \mathbb{E}_{(X_{\mathbf{O}},\tilde{X}_{\mathbf{O}})\sim\Gamma}\left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] = \mathbb{E}_{(X_{\mathbf{O}},\mathrm{PA}_{X_{\mathbf{O}}},\tilde{X}_{\mathbf{O}})\sim\alpha}\left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
&= \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \left[\mathrm{PA}_{X_i}\sim\gamma_i(\cdot|X_i)\right]_{i\in\mathbf{O}}, \tilde{X}_{\mathbf{O}}\sim\alpha_{BC}(\cdot|\mathrm{PA}_{X_{\mathbf{O}}})}\left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
&\overset{(1)}{=} \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \left[\mathrm{PA}_{X_i}=\phi_i(X_i)\right]_{i\in\mathbf{O}}, \tilde{X}_{\mathbf{O}}=\psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})}\left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
&= \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}=\phi(X_{\mathbf{O}}), \tilde{X}_{\mathbf{O}}=\psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})}\left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
&\overset{(2)}{=} \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}=\phi(X_{\mathbf{O}})}\left[c\left(X_{\mathbf{O}}, \psi_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)\right] \\
&\geq \inf_{[\phi_i\in\mathfrak{C}(X_i)]_{i\in\mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}=\phi(X_{\mathbf{O}})}\left[c\left(X_{\mathbf{O}}, \psi_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)\right].
\end{aligned} \tag{6}$$

Here we note that we have $\overset{(1)}{=}$ because α_{BC} is a deterministic coupling and we have $\overset{(2)}{=}$ because the expectation is preserved through a deterministic push-forward map.

Let $[\phi_i \in \mathfrak{C}(X_i)]_{i\in\mathbf{O}}$ be the optimal backward maps of the optimization problem (OP) in (A). We define the joint distribution γ over $P_d\left(X_{\mathbf{O}}\right)$ and $P_\theta(\mathrm{PA}_{X_{\mathbf{O}}}) = P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ as follows. We first sample $X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}})$ and for each $i \in \mathbf{O}$, we sample $\mathrm{PA}_{X_i} \sim \phi_i(X_i)$, and finally gather $(X_{\mathbf{O}}, \mathrm{PA}_{X_{\mathbf{O}}}) \sim \gamma$ where $\mathrm{PA}_{X_{\mathbf{O}}} = [\mathrm{PA}_{X_i}]_{i\in\mathbf{O}}$. Consider the joint distribution γ over $P_d\left(X_{\mathbf{O}}\right)$, $P_\theta(\mathrm{PA}_{X_{\mathbf{O}}}) = P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ and the deterministic coupling or joint distribution $\beta = (id, \psi_\theta)\#P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ over $P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$, the gluing lemma indicates the existence of the joint distribution α over $A \times C \times B$ such that $\alpha_{AB} = (\pi_A, \pi_B)\#\alpha = \gamma$ and $\alpha_{BC} = (\pi_B, \pi_C)\#\alpha = \beta$. We further denote

$\Gamma = \alpha_{AC} = (\pi_A, \pi_C)\#\alpha$ which is a joint distribution over $P_d(X_{\mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$. It follows that

$$\begin{aligned}
&\inf_{[\phi_i \in \mathfrak{C}(X_i)]_{i\in\mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}=\phi(X_{\mathbf{O}})} \left[c\left(X_{\mathbf{O}}, \psi_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)\right] \\
=&\mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}=\phi(X_{\mathbf{O}})} \left[c\left(X_{\mathbf{O}}, \psi_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)\right] \\
\overset{(1)}{=}&\mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}}), \tilde{X}_{\mathbf{O}}=\psi_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)} \left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
=&\mathbb{E}_{X_{\mathbf{O}}\sim P_d, \mathrm{PA}_{X_{\mathbf{O}}}\sim\gamma(\cdot|X_{\mathbf{O}}), \tilde{X}_{\mathbf{O}}\sim\alpha_{BC}(\cdot|\mathrm{PA}_{X_{\mathbf{O}}})} \left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
=&\mathbb{E}_{(X_{\mathbf{O}}, \mathrm{PA}_{X_{\mathbf{O}}}, \tilde{X}_{\mathbf{O}})\sim\alpha} \left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \\
=&\mathbb{E}_{(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}})\sim\Gamma} \left[c\left(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}\right)\right] \geq W_c\left(P_d\left(X_{\mathbf{O}}\right), P_\theta\left(X_{\mathbf{O}}\right)\right).
\end{aligned} \tag{7}$$

Here we note that we have $\overset{(1)}{=}$ because the expectation is preserved through a deterministic push-forward map.

Finally, combining (6) and (7), we reach the conclusion. □

It is worth noting that according to Theorem 4.1, we need to solve the following OP:

$$\inf_{\left[\phi_i \in \mathfrak{C}(X_i)\right]_{i\in\mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}})} \left[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right], \tag{8}$$

where $\mathfrak{C}\left(X_i\right) = \left\{\phi_i : \phi_i\#P_d\left(X_i\right) = P_\theta\left(\mathrm{PA}_{X_i}\right)\right\}, \forall i \in \mathbf{O}$.

If we make some further assumptions including: (i) the family model distributions $P_\theta, \theta \in \Theta$ induced by the graphical model is sufficiently rich to contain the data distribution, meaning that there exist $\theta^* \in \Theta$ such that $P_{\theta^*}(X_{\mathbf{O}}) = P_d(X_{\mathbf{O}})$ and (ii) the family of backward maps $\phi_i, i \in \mathbf{O}$ has infinite capacity (i.e., they include all measure functions), the infimum really peaks 0 at an optimal backward maps $\phi_i^*, i \in \mathbf{O}$. We thus can replace the infimum by a minimization as

$$\min_{\left[\phi_i \in \mathfrak{C}(X_i)\right]_{i\in\mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}})} \left[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right]. \tag{9}$$

To make the OP in (9) tractable for training, we do relaxation as

$$\min_{\phi} \left\{\mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}})} \left[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right] + \eta D\left(P_\phi, P_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)\right\}, \tag{10}$$

where $\eta > 0$, P_ϕ is the distribution induced by the backward maps, and D represents a general divergence. Here we note that $D\left(P_\phi, P_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right)$ can be decomposed into

$$D\left(P_\phi, P_\theta\left(\mathrm{PA}_{X_{\mathbf{O}}}\right)\right) = \sum_{i\in\mathbf{O}} D_i\left(P_{\phi_i}, P_\theta\left(\mathrm{PA}_{X_{\mathbf{i}}}\right)\right),$$

which is the sum of the divergences between the specific backward map distributions and their corresponding model distributions on the parent nodes (i.e., $P_{\phi_i} = \phi_i\#P_d\left(X_i\right)$). Additionally, in practice, using the WS distance for D_i leads to the following OP

$$\min_{\phi} \left\{\mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi(X_{\mathbf{O}})} \left[c\big(X_{\mathbf{O}}, \psi_\theta(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right] + \eta \sum_{i\in\mathbf{O}} \mathrm{W}_{c_i}\left(P_{\phi_i}, P_\theta\left(\mathrm{PA}_{X_i}\right)\right)\right\}. \tag{11}$$

The following theorem characterizes the ability to search the optimal solutions for the OPs in (9), (10), and (11).

Theorem A.1. *Assume that the family model distributions $P_\theta, \theta \in \Theta$ induced by the graphical model is sufficiently rich to contain the data distribution, meaning that there exist $\theta^* \in \Theta$ such that $P_{\theta^*}(X_{\mathbf{O}}) = P_d(X_{\mathbf{O}})$ and the family of backward maps $\phi_i, i \in \mathbf{O}$ has infinite capacity (i.e., they include all measure functions). The OPs in (9), (10), and (11) are equivalent and can obtain the common optimal solution.*

Proof. Let $\theta^* \in \Theta$ be the optimal solution such that $P_{\theta^*}(X_{\mathbf{O}}) = P_d(X_{\mathbf{O}})$ and $W_c\left(P_d\left(X_{\mathbf{O}}\right), P_{\theta^*}\left(X_{\mathbf{O}}\right)\right) = 0$. Let $\Gamma^* \in \mathcal{P}(P_d(X_{\mathbf{O}}), P_\theta(X_{\mathbf{O}}))$ be the optimal joint distribution over $P_d(X_{\mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$ of the corresponding Wasserstein distance, meaning that if $(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}) \sim \Gamma^*$ then $X_{\mathbf{O}} = \tilde{X}_{\mathbf{O}}$. Using the gluing lemma as in the previous theorem, there exists a joint distribution α^* over $A \times B \times C$ such that $\alpha^*_{AC} = (\pi_A, \pi_C)\#\alpha^* = \Gamma^*$ and $\alpha^*_{BC} = (\pi_B, \pi_C)\#\alpha^* = \beta^*$ where $\beta^* = (id, \psi_\theta)\#P^*_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ is a deterministic coupling or joint distribution over $P_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$ and $P^*_\theta(X_{\mathbf{O}})$. This follows that α^* consists of the sample $(X_{\mathbf{O}}, \mathrm{PA}_{X_{\mathbf{O}}}, X_{\mathbf{O}})$ where $\psi_{\theta^*}(\mathrm{PA}_{X_{\mathbf{O}}}) = X_{\mathbf{O}}$ with $X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}) = P^*_\theta(X_{\mathbf{O}})$.

Let us denote $\gamma^* = (\pi_A, \pi_B)\#\alpha^*$ as a joint distribution over $P_d(X_{\mathbf{O}})$ and $P^*_\theta([\mathrm{PA}_{X_i}]_{i\in\mathbf{O}})$. Let $\gamma^*_i, i \in \mathbf{O}$ as the restriction of γ^* over $P_d(X_i)$ and $P^*_\theta(\mathrm{PA}_{X_i})$. Let $\phi^*_i, i \in \mathbf{O}$ be the functions in the family of the backward functions that can well-approximate $\gamma^*_i, i \in \mathbf{O}$ (i.e., $\phi^*_i = \gamma^*_i, i \in \mathbf{O}$). For any $X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}})$, we have for all $i \in \mathbf{O}$, $\mathrm{PA}_{X_i} = \phi^*_i(X_i)$ and $\psi_{\theta^*}(\mathrm{PA}_{X_i}) = X_i$. These imply that (i) $\mathbb{E}_{X_{\mathbf{O}}\sim P_d(X_{\mathbf{O}}), \mathrm{PA}_{X_{\mathbf{O}}}\sim\phi^*(X_{\mathbf{O}})}\left[c\big(X_{\mathbf{O}}, \psi_{\theta^*}(\mathrm{PA}_{X_{\mathbf{O}}})\big)\right] = 0$ and (ii) $P_{\phi^*_i} = P_{\theta^*}\left(\mathrm{PA}_{X_i}\right), \forall i \in \mathbf{O}$, which further indicate that the OPs in (9), (10), and (11) are minimized at 0 with the common optimal solution ϕ^* and θ^*. □

B. OTP-DAG as a generalization of WAE

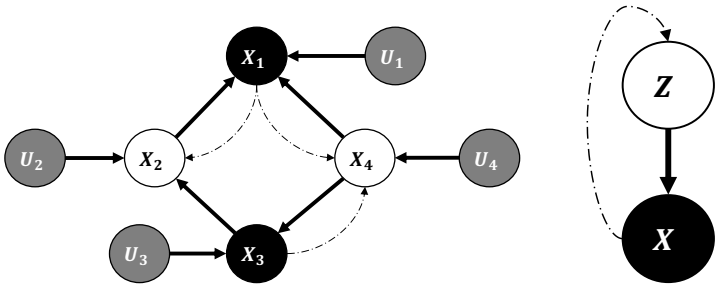

Figure 5. **(Left)** Algorithmic DAG. **(Right)** Standard Auto-encoder.

Figure 5 sheds light on an interesting connection of our method with auto-encoding models. Considering a graphical model of only two nodes: the observed node X and its latent parent Z, we define a backward map ϕ over X such that $\phi\#P_d(X) = P_\theta(Z)$ where $P_\theta(Z)$ is the prior over Z. The backward map can be viewed as a (stochastic) encoder approximating the prior $P_\theta(Z)$ with $P_\phi(Z) := \mathbb{E}_X[\phi(Z|X)]$. OTP-DAG now reduces to Wasserstein auto-encoder WAE (Tolstikhin et al., 2017), where the forward mapping ψ plays the role of the decoder. OTP-DAG therefore serves as a generalization of WAE for learning a more complex structure where there is the interplay of more parameters and hidden variables.

In this simplistic case, our training procedure is precisely as follows:

1. Draw $x \sim P_d(X)$.
2. Draw $z \sim \phi(Z|X)$.
3. Draw $\widetilde{x} \sim P_\theta(X|Z)$.
4. Update ϕ and θ alternately by descending objective (5).

Our cost function explicitly minimizes two terms: (1) the push-forward divergence $D[P_\phi(Z), P_\theta(Z)]$ where D is an arbitrary divergence, and (2) the reconstruction loss between x and $\widetilde{x}$. At a high level, our learning dynamic ensures $\phi_\# P_d(X) = P_\phi(Z) = P_\theta(Z)$ so that $z \sim P_\phi(Z)$ follows the prior distribution $P_\theta(Z)$. However, such samples cannot ignore information in the input X because we need $\widetilde{x} \sim P_\theta(X \mid Z = z)$ to effectively reconstruct the observed samples.

There might also be a concern that relaxing the push-forward constraint into the divergence term means the backward ϕ is forced to mimic the prior, which may lead to a situation similar to **posterior collapse** notoriously occurring to VAE. We here detail why VAE is prone to this issue and how the OT-based objective mitigates it.

1. The push-forward divergence: While the objectives of OTP-DAG/WAE and VAE entail the prior matching term. the two formulations are different in nature.

Let Q denote the set of variational distributions. If we consider ϕ as an encoder, the VAE objective can be written as

$$\inf_{\phi(Z|X)\in Q} \mathbb{E}_{X\sim P(X)}[D_{\mathrm{KL}}(\phi(Z|X), P_\theta(Z))] - \mathbb{E}_{Z\sim\phi(Z|X)}[\log P_\theta(X|Z)]. \tag{12}$$

By minimizing the above KL divergence term, VAE basically tries to match the prior $P(Z)$ for all different examples drawn from $P_d(X)$. Under the VAE objective, it is thus easier for ϕ to collapse into a distribution independent of $P_d(X)$, where specifically latent codes are close to each other and reconstructed samples are concentrated around only few values.

For OTP-DAG/WAE, the regularizer in fact penalizes the discrepancy between $P_\theta(Z)$ and $P_\phi(Z) := \mathbb{E}_X[\phi(Z|X)]$, which can be optimized using GAN-based, MMD-based or Wasserstein distance. The latent codes of different examples $x \sim P_d(X)$ can lie far away from each other, which allows the model to maintain the dependency between the latent codes and the input. Therefore, it is more difficult for ϕ to mimic the prior and trivially satisfy the push-forward constraint. We refer readers to Tolstikhin et al. (2017) for extensive empirical evidence.

2. The reconstruction loss: At some point of training, there is still a possibility to land at ϕ that yields samples Z independent of input X. If this occurs, $\phi\#\delta x_c^{(1)} = \phi\#\delta x_c^{(2)} = P(Z)$ for any points $x_c^{(1)}, x_c^{(2)} \sim P_d(X)$. This means $\mathrm{supp}(\phi(X^1)) = \mathrm{supp}(\phi(X^2)) = \mathrm{supp}(P_\theta(Z))$, so it would result in a very large reconstruction loss because it requires to map $\mathrm{supp}(P_\theta(Z))$ to various X^1 and X^2. Thus our reconstruction term would heavily penalizes this. In other words, this term explicitly encourages the model to search for θ that helps reconstruction, thus preventing the model from converging to the backward ϕ that produces sub-optimal ancestral samples.

Meanwhile, for VAE, if the family Q contains all possible conditional distribution $\phi(Z|X)$, its objective is essentially to maximize the marginal log-likelihood $\mathbb{E}_{P(X)}[\log P_\theta(X)]$, or minimize the KL divergence $\mathrm{KL}(P_d, P_\theta)$. It is shown in Dai et al. (2020) that under posterior collapse, VAE produces poor reconstructions yet the loss can still decrease i.e achieve low negative log-likelihood scores and still able to assign high-probability to the training data.

In summary, it is such construction and optimization of the backward that prevents OTP-DAG from posterior collapse situation. We here search for ϕ within a family of measurable functions and in practice approximate it with deep neural networks. It comes down to empirical decisions to select the architecture sufficiently expressive in each application.

C. Experimental setup

In the following, we explain how OTP-DAG algorithm is implemented in practical applications, including how to reparameterize the model distribution, to design the backward mapping and to define the optimization objective. We also here provide the training configurations for our method and the baselines. All models are run on 4 RTX 6000 GPU cores using Adam optimizer with a fixed learning rate of $1e-3$. Our code is published at `https://github.com/isVy08/OTP`.

C.1. Latent Dirichlet Allocation

For completeness, let us recap the model generative process. We consider a corpus $\mathcal{D}$ of M independent documents where each document is a sequence of N words denoted by $W_{1:N} = (W_1, W_2, \cdots, W_N)$. Documents are represented as random mixtures over K latent topics, each of which is characterized by a distribution over words. Let V be the size of a vocabulary indexed by $\{1, \cdots, V\}$. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) dictates the following generative process for every document in the corpus:

1. Choose $\theta \sim \mathrm{Dir}(\alpha)$,
2. Choose $\gamma_k \sim \mathrm{Dir}(\beta)$ where $k \in \{1, \cdots, K\}$,
3. For each of the word positions $n \in \{1, \cdots, N\}$,
   - Choose a topic $z_n \sim$ Multi-Nominal(θ),
   - Choose a word $w_n \sim$ Multi-Nominal$(\gamma_k n)$,

where Dir(.) is a Dirichlet distribution, $\alpha < 1$ and β is typically sparse. θ is a $K-$dimensional vector that lies in the $(K-1)-$simplex and γ_k is a $V-$dimensional vector represents the word distribution corresponding to topic k. Throughout the experiments, K is fixed at 10.

Parameter estimation. We consider the topic-word distribution γ as a fixed quantity to be estimated. γ is a $K \times V$ matrix where $\gamma_{kn} := P(W_n = 1|Z_n = 1)$. The learnable parameters therefore consist of γ and α. An input document is represented with a $N \times V$ matrix where a word W_i is represented with a one-hot $V-$vector such that the value at the index i in the vocabulary is 1 and 0 otherwise. Given $\gamma \in [0,1]^{K\times V}$ and a selected topic k, the deterministic forward mapping to generate a document W is defined as

$$W_{1:N} = \psi_\gamma(Z) = \text{Cat-Concrete}\big(\text{softmax}(Z'\gamma)\big),$$

where $Z \in \{0,1\}^K$ is in the one-hot representation (i.e., $Z^k = 1$ if state k is the selected and 0 otherwise) and Z' is its transpose. By applying the Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2016), we re-parameterize the Categorical distribution into a function Cat-Concrete(.) that takes the categorical probability vector (i.e., sum of all elements equals 1) and output a relaxed probability vector. To be more specific, given a categorical variable of K categories with probabilities $\big[p_1, p_2, ..., p_K\big]$, for every the Cat-Concrete(.) function is defined on each p_k as

$$\text{Cat-Concrete}(p_k) = \frac{\exp\big\{(\log p_k + G_k)/\tau\big\}}{\sum_{k=1}^{K}\exp\big\{(\log p_k + G_k)/\tau\big\}},$$

with temperature τ, random noises G_k independently drawn from Gumbel distribution $G_t = -\log(-\log u_t),\ u_t \sim \text{Uniform}(0,1)$.

We next define a backward map that outputs for a document a distribution over K topics by $\phi(Z \mid W_{1:N}) = \text{Cat}(Z)$. Given observations $W_{1:N}$, our learning procedure begins by sampling $\widetilde{Z} \sim \phi(Z|W_{1:N})$ and pass $\widetilde{Z}$ through the generative process given by ψ to obtain the reconstruction. Notice here that we have a prior constraint over the distribution of θ i.e., θ follows a Dirichlet distribution parameterized by α. This translates to a push forward constraint in order to optimize for α. To facilitate differentiable training, we use softmax Laplace approximation (MacKay, 1998; Srivastava & Sutton, 2017) to approximate a Dirichlet distribution with a softmax Gaussian distribution. The relation between α and the Gaussian parameters $\big(\mu_k, \Sigma_k\big)$ w.r.t a category k where Σ_k is a diagonal matrix is given as

$$\mu_k(\alpha) = \log\alpha_k - \frac{1}{K}\sum_{i=1}^{K}\log\alpha_i, \quad \Sigma_k(\alpha) = \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_{i=1}^{K}\frac{1}{\alpha_i}. \tag{13}$$

Let us denote $P_\alpha := \mathcal{N}\big(\mu(\alpha), \Sigma(\alpha)\big) \approx \text{Dir}(\alpha)$ with $\mu = [\mu_k]_{k=1}^K$ and $\Sigma = [\Sigma_k]_{k=1}^K$ defined as above. The optimization objective is given as

$$\min_{\alpha,\gamma} \quad \mathbb{E}_{W_{1:N}\sim\mathcal{D},\widetilde{Z}\sim\phi(Z|W_{1:N})} c\big[W_{1:N}, \psi_\gamma(\widetilde{Z})\big] + \eta\, D_{\text{WS}}\big(P_\phi(Z), P_\alpha(Z)\big),$$

where c is cross-entropy loss function and D_{WS} is exact Wasserstein distance[6]. The sampling process $\theta \sim P_\alpha$ is also relaxed using standard Gaussian reparameterization trick whereby $\theta = \mu(\alpha) + u\Sigma(\alpha)$ with $u \sim \mathcal{N}(0,1)$.

Remark. Our framework in fact learns both α and γ at the same time. Our estimates for α (averaged over K) are nearly 100% faithful at $0.10, 0.049, 0.033$ (recall that the ground-truth α is uniform over K where $K = 10, 20, 30$ respectively). Figure 6 shows the convergence behavior of OTP-DAG during training where our model converges to the ground-truth patterns relatively quickly.

Figures 7 and 8 additionally present the topic distributions of each method for the second and third synthetic sets. We use horizontal and vertical patterns in different colors to distinguish topics from one another. Red circles indicate erroneous patterns. Note that these configurations are increasingly more complex, so it requires more training time for all methods to achieve better performance. Although our method may exhibit some inconsistencies in recovering accurate word distributions for each topic, these discrepancies are comparatively less pronounced when compared to the baseline methods. This observation indicates a certain level of robustness in our approach.

[6] https://pythonot.github.io/index.html

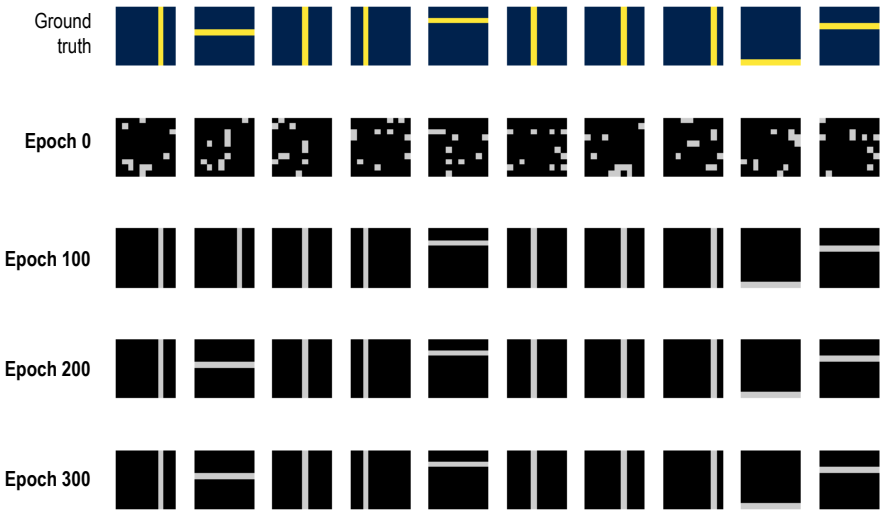

Figure 6. Converging patterns of 10 random topics from our OTP-DAG after $100, 200, 300$ iterations.

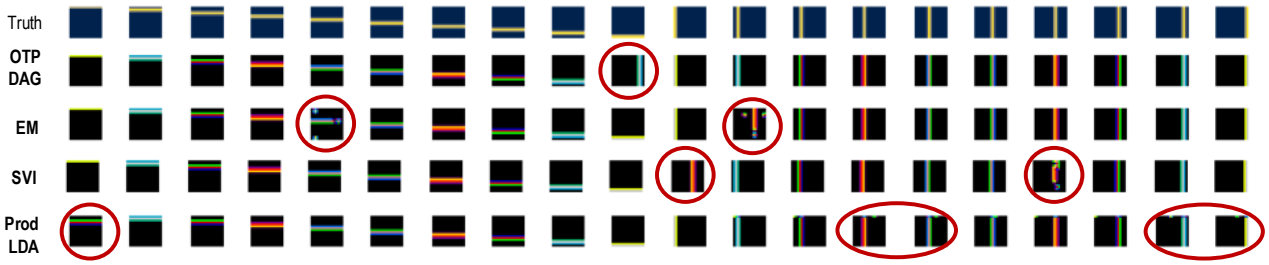

Figure 7. Topic-word distributions inferred by each method from the second set of synthetic data after 300 training epochs.

Figure 8. Topic-word distributions inferred by OTP-DAG from the third set of synthetic data after 300 training epochs.

Topic inference. In this experiment, we apply OTP-DAG on real-world datasets. We here revert to the original generative process where the topic-word distribution follows a Dirichlet distribution parameterized by the concentration parameters β, instead of having γ as a fixed quantity. In this case, β is initialized as a matrix of real values i.e., $\beta \in \mathbb{R}^{K \times V}$ representing the log concentration values. The forward process is given as

$$W_{1:N} = \psi_\gamma(Z) = \text{Cat-Concrete}\big(\text{softmax}(Z'\gamma)\big),$$

where $\gamma_k = \mu_k\big(\exp(\beta_k)\big) + u_k \Sigma_k\big(\exp(\beta_k)\big)$ and $u_k \sim \mathcal{N}(0, 1)$ is a Gaussian noise. This is realized by using softmax

Gaussian trick as in Eq. (13), then applying standard Gaussian reparameterization trick. The optimization procedure follows the previous application.

Table 5. Qualitive evaluation of the topics inferred for 3 real-world datasets.

20 News Group	
Topic 1	*car, bike, front, engine, mile, ride, drive, owner, road, buy*
Topic 2	*game, play, team, player, season, fan, win, hit, year, score*
Topic 3	*government, public, key, clipper, security, encryption, law, agency, private, technology*
Topic 4	*religion, christian, belief, church, argument, faith, truth, evidence, human, life*
Topic 5	*window, file, program, software, application, graphic, display, user, screen, format*
Topic 6	*mail, sell, price, email, interested, sale, offer, reply, info, send*
Topic 7	*card, drive, disk, monitor, chip, video, speed, memory, system, board*
Topic 8	*kill, gun, government, war, child, law, country, crime, weapon, death*
Topic 9	*make, time, good, people, find, thing, give, work, problem, call*
Topic 10	*fire, day, hour, night, burn, doctor, woman, water, food, body*
BBC News	
Topic 1	*rise, growth, market, fall, month, high, economy, expect, economic, price*
Topic 2	*win, play, game, player, good, back, match, team, final, side*
Topic 3	*user, firm, website, computer, net, information, software, internet, system, technology*
Topic 4	*technology, market, digital, high, video, player, company, launch, mobile, phone*
Topic 5	*election, government, party, labour, leader, plan, story, general, public, minister*
Topic 6	*film, include, star, award, good, win, show, top, play, actor*
Topic 7	*charge, case, face, claim, court, ban, lawyer, guilty, drug, trial*
Topic 8	*thing, work, part, life, find, idea, give, world, real, good*
Topic 9	*company, firm, deal, share, buy, business, market, executive, pay, group*
Topic 10	*government, law, issue, spokesman, call, minister, public, give, rule, plan*
DBLP	
Topic 1	*learning, algorithm, time, rule, temporal, logic, framework, real, performance, function*
Topic 2	*efficient, classification, semantic, multiple, constraint, optimization, probabilistic, domain, process, inference*
Topic 3	*search, structure, pattern, large, language, web, problem, representation, support, machine*
Topic 4	*object, detection, application, information, method, estimation, multi, dynamic, tree, motion*
Topic 5	*system, database, query, knowledge, processing, management, orient, relational, expert, transaction*
Topic 6	*model, markov, mixture, variable, gaussian, topic, hide, latent, graphical, appearance*
Topic 7	*network, approach, recognition, neural, face, bayesian, belief, speech, sensor, artificial*
Topic 8	*base, video, content, code, coding, scalable, rate, streaming, frame, distortion*
Topic 9	*datum, analysis, feature, mining, cluster, selection, high, stream, dimensional, component*
Topic 10	*image, learn, segmentation, retrieval, color, wavelet, region, texture, transform, compression*

Training configuration. The underlying architecture of the backward maps consists of an LSTM and one or more linear layers. We train all models for 300 and $1,000$ epochs with batch size of 50 respectively for the 2 applications. We also set $\tau = 1.0, 2.0$ and $\eta = 1e-4, 1e-1$ respectively.

C.2. Hidden Markov Models

We here attempt to learn a Poisson hidden Markov model underlying a data stream. Given a time series $\mathcal{D}$ of T steps, the task is to segment the data stream into K different states, each of which is associated with a Poisson observation model with rate λ_k. The observation at each step t is given as

$$P(X_t|Z_t = k) = \text{Poi}(X_t|\lambda_k), \quad \text{for } k = 1, \cdots, K.$$

The Markov chain stays in the current state with probability p and otherwise transitions to one of the other $K-1$ states

uniformly at random. The transition distribution is given as

$$z_1 \sim \mathrm{Cat}\left(\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right), \quad z_t | z_{t-1} \sim \mathrm{Cat}\left(\left\{ \begin{array}{cc} \pi & \text{if } Z_t = Z_{t-1} \\ \frac{1-\pi}{4-1} & \text{otherwise} \end{array} \right\}\right)$$

We first apply Gaussian reparameterization on each Poisson distribution, giving rise to a deterministic forward mapping

$$X_t = \psi_t(Z_t) = Z_t' \exp(\lambda) + u_t \sqrt{Z_t \exp(\lambda)},$$

where $\lambda \in \mathbb{R}^K$ is the learnable parameter vector representing log rates, $u_k \sim \mathcal{N}(0, 1)$ is a Gaussian noise, $Z_t \in \{0, 1\}^K$ is in the one-hot representation and Z_t' is its transpose. We define a global backward map ϕ that outputs the distributions for individual Z_t as $\phi(Z_t|X_t) := \mathrm{Cat}(Z_t|X_t)$.

The first term in the optimization object is the reconstruction error given by a cost function c. The push forward constraint ensures the backward probabilities for the state variables align with the prior transition distributions. Denoting $\psi = [\psi_t]_{t=1}^T$, we learn $\lambda_{1:K}$ by optimizing the following objective

$$\min_{\lambda} \quad \mathbb{E}_{X_{1:T} \sim \mathcal{D}, \widetilde{Z}_{1:T} \sim \phi(Z_{1:T}|X_{1:T})} c\big[X_{1:T}, \psi(\widetilde{Z}_{1:T})\big] + \eta \, D_{\mathrm{WS}}\big[P_\phi(Z_{1:T}), P_\pi(Z_{1:T})\big].$$

In the experiment, we choose $T = 200$ and smooth L_1 loss (Girshick, 2015) is chosen as the cost function. D_{WS} is exact Wasserstein distance with KL divergence as the ground cost.

Training configuration. The underlying architecture of the backward map is a $3-$ layer fully connected perceptron. The Poisson HMM is trained for 50 epochs with $\eta = 0.1$ and $\tau = 0.1$.

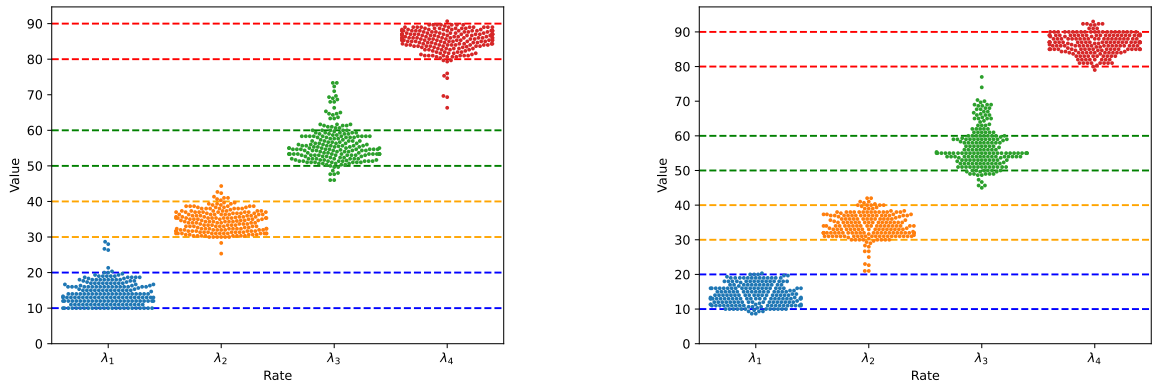

Figure 9. Distribution of the estimates of the concentration parameters $\lambda_{1:4}$ from OTP-DAG **(Left)** and EM **(Right)**. Our estimated distribution aligns more uniformly with the true generative process.

C.3. Learning Discrete Representations

To understand vector quantized models, let us briefly review Quantization Variational Auto-Encoder (VQ-VAE) (Van Den Oord et al., 2017). The practical setting of VQ-VAE in fact considers a $M-$dimensional discrete latent space $\mathcal{C}^M \in \mathbb{R}^{M \times D}$ that is the $M-$ary Cartesian power of $\mathcal{C}$ with $\mathcal{C} = \{c_k\}_{k=1}^K \in \mathbb{R}^{K \times D}$ i.e., $\mathcal{C}$ here is the set of learnable latent embedding vectors c_k. The latent variable $Z = [Z^m]_{m=1}^M$ is an $M-$component vector where each component $Z^m \in \mathcal{C}$. VQ-VAE is an encoder-decoder, in which the encoder $f_e : \mathcal{X} \mapsto \mathbb{R}^{M \times D}$ maps the input data X to the latent representation Z and the decoder $f_d : \mathbb{R}^{M \times D} \mapsto \mathcal{X}$ reconstructs the input from the latent representation. However, different from standard VAE, the latent representation used for reconstruction is discrete, which is the projection of Z onto $\mathcal{C}^M$ via the quantization

process Q. Let $\bar{Z}$ denote the discrete representation. The quantization process is modeled as a deterministic categorical posterior distribution such that

$$\bar{Z}^m = Q(Z^m) = c_k,$$

where $k = \underset{k}{\operatorname{argmin}}\, d\big(Z^m, c_k\big)$, $Z^m = f_e^m(X)$ and d is a metric on the latent space.

In our language, each vector c_k can be viewed as the centroid representing each latent sub-space (or cluster). The quantization operation essentially searches for the closet cluster for every component latent representation z^m. VQ-VAE minimizes the following objective function:

$$\mathbb{E}_{x\sim\mathcal{D}}\Big[d_x\big[f_d\big(Q(f_e(x))\big), x\big] + d_z\big[\mathbf{sg}\big(f_e(x)\big), \bar{z}\big] + \beta d_z\big[f_e(x), \mathbf{sg}\big(\bar{z}\big)\big]\Big],$$

where $\mathcal{D}$ is the empirical data, **sg** is the stop gradient operation for continuous training, d_x, d_z are respectively the distances on the data and latent space and β is set between 0.1 and 2.0 in the original proposal (Van Den Oord et al., 2017).

In our work, we explore a different model to learning discrete representations. Following VQ-VAE, we also consider Z as a $M-$component latent embedding. On a kth sub-space (for $k \in \{1, \cdots, K\}$), we impose a Gaussian distribution parameterized by μ_k, Σ_k where Σ_k is diagonal. We also endow M discrete distributions over $\mathbf{C}^1, \ldots, \mathbf{C}^M$, sharing a common support set as the set of sub-spaces induced by $\{(\mu_k, \Sigma_k)\}_{k=1}^K$:

$$\mathbb{P}_{k,\pi^m} = \sum_{k=1}^{K} \pi_k^m \delta_{\mu_k}, \text{ for } m = 1, \ldots, M.$$

with the Dirac delta function δ and the weights $\pi^m \in \Delta_{K-1} = \{\alpha \geq \mathbf{0} : \|\alpha\|_1 = 1\}$ in the $(K-1)$-simplex. The probability a data point z^m belongs to a discrete k^{th} sub-space follows a $K-$way categorical distribution $\pi^m = [\pi_1^m, \cdots, \pi_K^m]$. In such a practical setting, the generative process is detailed as follows

1. For $m \in \{1, \cdots, M\}$,
   - Sample $k \sim \text{Cat}(\pi^m)$,
   - Sample $z^m \sim \mathcal{N}(\mu_k, \Sigma_k)$,
   - Quantize $\mu_k^m = Q(z^m)$,
2. $x = \psi_\theta([z^m]_{m=1}^M, [\mu_k^m]_{m=1}^M)$.

where ψ is a highly non-convex function with unknown parameters θ. Q refers to the quantization of $[z^m]_{m=1}^M$ to $[\mu_k^m]_{m=1}^M$ defined as $\mu_k^m = Q(z^m)$ where $k = \operatorname{argmin}_k d_z\big(z^m; \mu_k\big)$ and $d_z = \sqrt{(z^m - \mu_k)^T \Sigma_k^{-1} (z^m - \mu_k)}$ is the Mahalanobis distance.

The backward map is defined via an encoder function f_e and quantization process Q as

$$\phi(x) = \big[f_e(x), Q(f_e(x))\big], \quad z = [z^m]_{m=1}^M = f_e(x), \quad [\mu_k^m]_{m=1}^M = Q(z).$$

The learnable parameters are $\{\pi, \mu, \Sigma, \theta\}$ with $\pi = [[\pi_k^m]_{m=1}^M]_{k=1}^K, \mu = [\mu_k]_{k=1}^K, \Sigma = [\Sigma_k]_{k=1}^K$.

Applying OTP-DAG to the above generative model yields the following optimization objective:

$$\min_{\pi,\mu,\Sigma,\theta} \quad \mathbb{E}_{X\sim\mathcal{D}}\Big[c\big[X, \psi_\theta(Z, \mu_k)\big]\Big] + \frac{\eta}{M} \sum_{m=1}^{M} \big[D_{\text{WS}}\big(P_\phi(Z^m), P(\widetilde{Z}^m)\big) + D_{\text{WS}}\big(P_\phi(Z^m), \mathbb{P}_{k,\pi^m}\big)\big]$$
$$+ \eta_r \sum_{m=1}^{M} D_{\text{KL}}\big(\pi^m, \mathcal{U}_K\big),$$

where $P_\phi(Z^m) := f_e^m \# P(X)$ given by the backward ϕ, $P(\widetilde{Z}^m) = \sum_{k=1}^K \pi_k^m \mathcal{N}(\widetilde{Z}^m | \mu_k, \Sigma_k)$ is the mixture of Gaussian distributions. The copy gradient trick (Van Den Oord et al., 2017) is applied throughout to facilitate backpropagation.

The first term is the conventional reconstruction loss where c is chosen to be mean squared error. Minimizing the second term $D_{\text{WS}}\big(P_\phi(Z^m), P(\widetilde{Z}^m)\big)$ forces the latent representations to follow the Gaussian distribution $\mathcal{N}(\mu_k^m, \Sigma_k^m)$. Minimizing the third term $D_{\text{WS}}\big(P_\phi(Z^m), \mathbb{P}_{k,\pi^m}\big)$ encourages every μ_k to become the clustering centroid of the set of latent representations Z^m associated with it. Additionally, the number of latent representations associated with the clustering centroids are proportional to $\pi_k^m, k = 1, ..., K$. Therefore, we use the fourth term $\sum_{m=1}^{M} D_{\text{KL}}\big(\pi^m, \mathcal{U}_K\big)$ to guarantee every centroid is utilized.

Training configuration. We use the same experiment setting on all datasets. The models have an encoder with two convolutional layers of stride 2 and filter size of 4×4 with ReLU activation, followed by 2 residual blocks, which contained a 3×3, stride 1 convolutional layer with ReLU activation followed by a 1×1 convolution. The decoder was similar, with two of these residual blocks followed by two de-convolutional layers. The hyperparameters are: $D = M = 64, K = 512, \eta = 1e-3, \eta_r = 1.0$, batch size of 32 and 100 training epochs.

Evaluation metrics. The evaluation metrics used include (1) **SSIM:** the patch-level structure similarity index, which evaluates the similarity between patches of the two images; (2) **PSNR:** the pixel-level peak signal-to-noise ratio, which measures the similarity between the original and generated image at the pixel level; (3) feature-level **LPIPS** (Zhang et al., 2018), which calculates the distance between the feature representations of the two images; (4) the dataset-level Fr'echlet Inception Distance **(FID)** (Heusel et al., 2017), which measures the difference between the distributions of real and generated images in a high-dimensional feature space; and (5) **Perplexity:** the degree to which the latent representations Z spread uniformly over K sub-spaces i.e., all K regions are occupied. Perplexity score is defined as $\exp\big(-\sum_{k=1}^{K} p_{c_k} \log p_{c_k}\big)$ where $p_{c_k} = N_{c_k} / \sum_{i=1}^{K} N_{c_i}$ is the probability of the ith codeword being used and N_{c_i} is the number of latent representations associated with the codeword c_i. Perplexity is maximized when the distribution over the codebooks is uniform, indicating that all codebooks are utilized equally by the model there is no posterior collapse. Thus, higher perplexity is preferred.

Additional experiment. We additionally investigate a recent model called SQ-VAE (Takida et al., 2022) proposed to tackle the issue of codebook utilization. Table 6 reports the performance of SQ-VAE in comparison with our OTP-DAG. We significantly outperform SQ-VAE on Perplexity, showing that our model mitigates codebook collapse issue more effectively, while compete on par with this SOTA model across the other metrics. It is worth noting that our goal here is not to propose any SOTA model to discrete representation learning, but rather to demonstrate the applicability of OTP-DAG on various tasks, particular problems where traditional methods such as EM or mean-field VI cannot simply tackle.

Qualitative examples. We first present the generated samples from the CelebA dataset using Image transformer (Parmar et al., 2018) as the generative model. From Figure 10, it can be seen that the discrete representation from the our method can be effectively utilized for image generation with acceptable quality.

We additionally show the reconstructed samples from CIFAR10 dataset for qualitative evaluation. Figures 11-13 illustrate that the reconstructions from OTP-DAG have higher visual quality than VQ-VAE. The high-level semantic features of the input image and colors are better preserved with OTP-DAG than VQ-VAE from which some reconstructed images are much more blurry.

Table 6. Quality of image reconstructions

Dataset	Method	Latent Size	SSIM ↑	PSNR ↑	LPIPS ↓	rFID ↓	Perplexity ↑
CIFAR10	**SQ-VAE**	8×8	0.80	**26.11**	0.23	**55.4**	434.8
	OTP-DAG (Ours)	8×8	0.80	25.40	0.23	56.5	**498.6**
MNIST	**SQ-VAE**	8×8	**0.99**	**36.25**	0.01	**3.2**	301.8
	OTP-DAG (Ours)	8×8	0.98	33.62	0.01	3.3	**474.6**
SVHN	**SQ-VAE**	8×8	**0.96**	**35.35**	**0.06**	**24.8**	389.8
	OTP-DAG (Ours)	8×8	0.94	32.56	0.08	25.2	**462.8**
CELEBA	**SQ-VAE**	16×16	0.88	**31.05**	0.12	14.8	427.8
	OTP-DAG (Ours)	16×16	0.88	29.77	**0.11**	**13.1**	**487.5**

Figure 10. Generated images from the discrete representations of OTP-DAG on CelebA dataset.

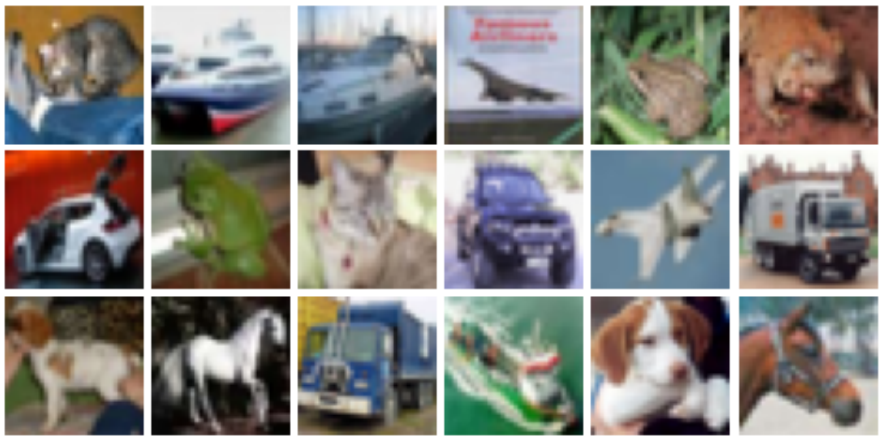

Figure 11. Original CIFAR10 images.

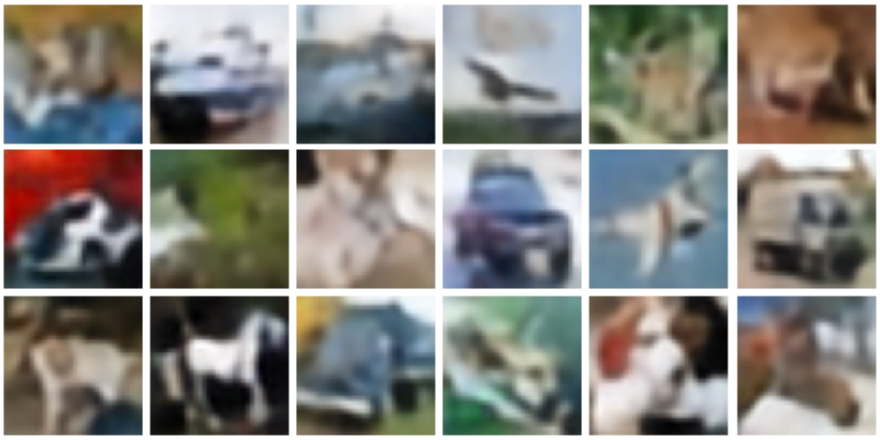

Figure 12. Random reconstructed images by VQ-VAE from CIFAR10 dataset.

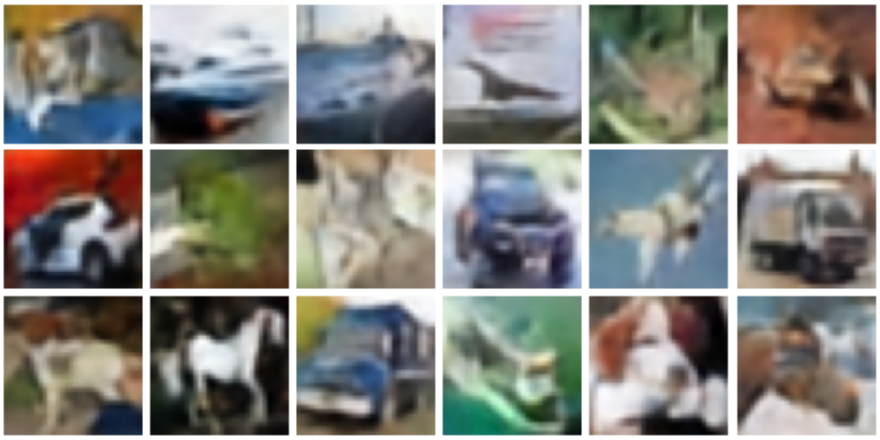

Figure 13. Random reconstructed images by OTP-DAG from CIFAR10 dataset.