
Supplementary Material for Learning Directed Graphical Models with Optimal Transport

Anonymous Authors

Affiliation

Address

email

A All Proofs

We now present the proof of Theorem A.1 which is the key theorem in our paper.

Theorem A.1. *For every ϕ_i as defined above and fixed ψ_θ ,*

$$W_c(P_d(X_{\mathbf{O}}); P_\theta(X_{\mathbf{O}})) = \inf_{[\phi_i \in \mathfrak{C}(X_i)]_{i \in \mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}), \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))], \quad (1)$$

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

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 $P_\theta(\text{PA}_{X_{\mathbf{O}}}) = P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ over $B = \prod_{i \in \mathbf{O}} \prod_{k \in \text{PA}_{X_i}} \mathcal{X}_k$. Here we note that the last distribution $P_\theta(\text{PA}_{X_{\mathbf{O}}}) = P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ 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 $\beta = (id, \psi_\theta) \# P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ be a deterministic coupling or joint distribution over $P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ and $P_\theta(X_{\mathbf{O}})$. Using the gluing lemma (see Lemma 5.5 in [7]), 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 $P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$.

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

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

Here we note that we have $\stackrel{(1)}{=}$ because α_{BC} is a deterministic coupling and we have $\stackrel{(2)}{=}$ 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 (4). We define the joint distribution γ over $P_d(X_{\mathbf{O}})$ and $P_\theta(\text{PA}_{X_{\mathbf{O}}}) = P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ as follows. We first sample $X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}})$ and for each $i \in \mathbf{O}$, we sample $\text{PA}_{X_i} \sim \phi_i(X_i)$, and finally gather $(X_{\mathbf{O}}, \text{PA}_{X_{\mathbf{O}}}) \sim \gamma$ where $\text{PA}_{X_{\mathbf{O}}} = [\text{PA}_{X_i}]_{i \in \mathbf{O}}$. Consider the joint distribution γ over $P_d(X_{\mathbf{O}})$, $P_\theta(\text{PA}_{X_{\mathbf{O}}}) = P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ and the deterministic coupling or joint distribution $\beta = (id, \psi_\theta) \# P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ over $P_\theta([\text{PA}_{X_i}]_{i \in \mathbf{O}})$ 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

$$\begin{aligned}
& \inf_{[\phi_i \in \mathfrak{C}(X_i)]_{i \in \mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}} \sim P_d, \text{PA}_{X_{\mathbf{O}}} = \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))] \\
&= \mathbb{E}_{X_{\mathbf{O}} \sim P_d, \text{PA}_{X_{\mathbf{O}}} = \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))] \\
&\stackrel{(1)}{=} \mathbb{E}_{X_{\mathbf{O}} \sim P_d, \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}}), \tilde{X}_{\mathbf{O}} = \psi_\theta(\text{PA}_{X_{\mathbf{O}}})} [c(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}})] \\
&= \mathbb{E}_{X_{\mathbf{O}} \sim P_d, \text{PA}_{X_{\mathbf{O}}} \sim \gamma(\cdot | X_{\mathbf{O}}), \tilde{X}_{\mathbf{O}} \sim \alpha_{BC}(\cdot | \text{PA}_{X_{\mathbf{O}}})} [c(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}})] \\
&= \mathbb{E}_{(X_{\mathbf{O}}, \text{PA}_{X_{\mathbf{O}}}, \tilde{X}_{\mathbf{O}}) \sim \alpha} [c(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}})] \\
&= \mathbb{E}_{(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}}) \sim \Gamma} [c(X_{\mathbf{O}}, \tilde{X}_{\mathbf{O}})] \geq W_c(P_d(X_{\mathbf{O}}), P_\theta(X_{\mathbf{O}})). \tag{3}
\end{aligned}$$

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

Finally, combining (2) and (3), we reach the conclusion. \square

It is worth noting that according to Theorem A.1, we need to solve the following OP:

$$\inf_{[\phi_i \in \mathfrak{C}(X_i)]_{i \in \mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}), \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))], \tag{4}$$

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

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

$$\min_{[\phi_i \in \mathfrak{C}(X_i)]_{i \in \mathbf{O}}} \mathbb{E}_{X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}), \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))]. \tag{5}$$

To make the OP in (5) tractable for training, we do relaxation as

$$\min_{\phi} \left\{ \mathbb{E}_{X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}), \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))] + \eta D(P_\phi, P_\theta(\text{PA}_{X_{\mathbf{O}}})) \right\}, \tag{6}$$

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

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

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

$$\min_{\phi} \left\{ \mathbb{E}_{X_{\mathbf{O}} \sim P_d(X_{\mathbf{O}}), \text{PA}_{X_{\mathbf{O}}} \sim \phi(X_{\mathbf{O}})} [c(X_{\mathbf{O}}, \psi_\theta(\text{PA}_{X_{\mathbf{O}}}))] + \eta \sum_{i \in \mathbf{O}} W_{c_i}(P_{\phi_i}, P_\theta(\text{PA}_{X_i})) \right\}. \tag{7}$$

The following theorem characterizes the ability to search the optimal solutions for the OPs in (5), (6), and (7).

Theorem A.2. 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 $P_{\theta^*}(X_O) = P_d(X_O)$ and the family of backward maps $\phi_i, i \in O$ has infinite capacity (i.e., they include all measure functions). The OPs in (5), (6), and (7) are equivalent and can obtain the common optimal solution.

Proof. Let $\theta^* \in \Theta$ be the optimal solution such that $P_{\theta^*}(X_O) = P_d(X_O)$ and $W_c(P_d(X_O), P_{\theta^*}(X_O)) = 0$. Let $\Gamma^* \in \mathcal{P}(P_d(X_O), P_{\theta^*}(X_O))$ be the optimal joint distribution over $P_d(X_O)$ and $P_{\theta^*}(X_O)$ of the corresponding Wasserstein distance, meaning that if $(X_O, \tilde{X}_O) \sim \Gamma^*$ then $X_O = \tilde{X}_O$. 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 $\beta^* = (id, \psi_\theta) \# P_\theta^*([PA_{X_i}]_{i \in O})$ is a deterministic coupling or joint distribution over $P_\theta([PA_{X_i}]_{i \in O})$ and $P_\theta^*(X_O)$. This follows that α^* consists of the sample (X_O, PA_{X_O}, X_O) where $\psi_{\theta^*}(PA_{X_O}) = X_O$ with $X_O \sim P_d(X_O) = P_{\theta^*}(X_O)$.

Let us denote $\gamma^* = (\pi_A, \pi_B) \# \alpha^*$ as a joint distribution over $P_d(X_O)$ and $P_\theta^*([PA_{X_i}]_{i \in O})$. Let $\gamma_i^*, i \in O$ as the restriction of γ^* over $P_d(X_i)$ and $P_\theta^*(PA_{X_i})$. Let $\phi_i^*, i \in O$ be the functions in the family of the backward functions that can well-approximate $\gamma_i^*, i \in O$ (i.e., $\phi_i^* = \gamma_i^*, i \in O$). For any $X_O \sim P_d(X_O)$, we have for all $i \in O$, $PA_{X_i} = \phi_i^*(X_i)$ and $\psi_{\theta^*}(PA_{X_i}) = X_i$. These imply that (i) $\mathbb{E}_{X_O \sim P_d(X_O), PA_{X_O} \sim \phi^*(X_O)} [c(X_O, \psi_{\theta^*}(PA_{X_O}))] = 0$ and (ii) $P_{\phi_i^*} = P_{\theta^*}(PA_{X_i}), \forall i \in O$, which further indicate that the OPs in (5), (6), and (7) are minimized at 0 with the common optimal solution ϕ^* and θ^* . \square

B Training algorithms

Algorithm 1 provides the pseudo-code for OTP-DAG learning procedure. The simplicity of the learning process is evident. Figure 1a visualizes our backward-forward algorithm in the empirical setting, where learning the backward functions for the endogenous variables only is sufficient for estimation. Regardless of the complexity of the graphical structure, a single learning procedure is applied. The first step is to identify the observed nodes and their parent nodes; then, for each parent-child pair, define the appropriate backward map and reparameterize the model distribution into a set of deterministic forward maps parameterized by θ (i.e., model parameters to be learned). Finally, one only needs to plug in the suitable cost function and divergence measure, and follow the backward-forward procedure to learn θ via stochastic gradient descent.

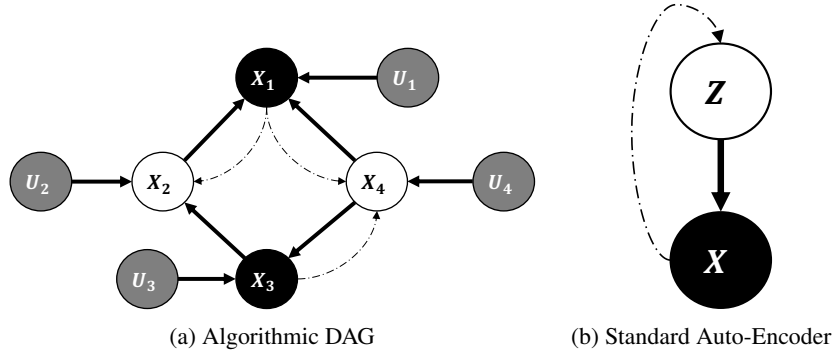


Figure 1

The application of discrete representation learning sheds light on an interesting connection of our method with auto-encoding models, particularly with Wasserstein auto-encoder [WAE, 9]. Indeed, WAE can be viewed as an application of OTP-DAG on a simple graphical model with only 2 nodes: the observed node X and latent variables Z . Likewise, both functions are jointly learned by minimizing the reconstruction loss. In this case, the backward mapping ϕ and forward mapping ψ respectively play the role of the encoder and decoder (See Figure 1b). Regardless, when there are more parameters and hidden variables interplaying in a more complex structure, the learning procedure of OTP-DAG still applies.

Algorithm 1 : OTP-DAG Algorithm

Input: Directed graph \mathbf{G} with observed nodes \mathbf{O} , noise distribution $P(U)$, stochastic backward maps $\phi = \{\phi_i(X_i)\}_{i \in \mathbf{O}}$, regularization coefficient η , reconstruction cost function c , and push-forward divergence measure D .

Output: Point estimate θ .

Re-parameterize P_θ into a set of deterministic mappings $\psi_\theta = \{\psi_{\theta_i}\}_{i \in \mathbf{O}}$ where $X_i = \psi_{\theta_i}(\text{PA}_{X_i}, U_i)$ and $U_i \sim P(U)$.

Initialize the parameters of the forward ψ_θ and backward ϕ mapping functions.

while not converged do

for $i \in \mathbf{O}$ **do**

 Sample batch $X_i^B = \{x_i^1, \dots, x_i^B\}$;

 Sample $\text{PA}_{X_i^B}$ from $\phi_i(X_i^B)$;

 Sampling U_i from the prior $P(U)$;

 Evaluate $\tilde{X}_i^B = \psi_{\theta_i}(\text{PA}_{X_i^B}, U_i)$.

end

 Update θ by descending

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

end

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 anonymously published at <https://anonymous.4open.science/r/OTP-7944/>.

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 $W_{1:N} = (W_1, W_2, \dots, W_N)$. 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 $\{1, \dots, V\}$. Latent Dirichlet Allocation (LDA) [1] dictates the following generative process for every document in the corpus:

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

where $\text{Dir}(\cdot)$ 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

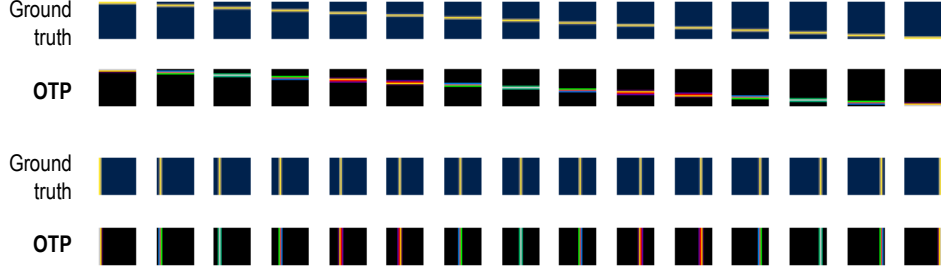


Figure 2: 30 topic-word distributions inferred by OTP-DAG from the third set of synthetic data. OTP-DAG recovers the ground-truth successfully.

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

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 [4, 6], we re-parameterize the Categorical distribution into a function $\text{Cat-Concrete}(\cdot)$ 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 $[p_1, p_2, \dots, p_K]$, for every the $\text{Cat-Concrete}(\cdot)$ function is defined on each p_k as

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

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 as follows

$$\phi(W_{1:N}) = \text{Cat}(Z).$$

Given observations $W_{1:N}$, our learning procedure begins by sampling $\tilde{Z} \sim P_\phi(Z|W_{1:N})$ and pass \tilde{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 [5, 8] to approximate a Dirichlet distribution with a softmax Gaussian distribution. The relation between α and the Gaussian parameters (μ_k, Σ_k) w.r.t a category k where Σ_k is a diagonal matrix is given as

$$\mu_k(\alpha) = \log \alpha_k - \frac{1}{K} \sum_{i=1}^K \log \alpha_i, \quad \Sigma_k(\alpha) = \frac{1}{\alpha_k} \left(1 - \frac{2}{K}\right) + \frac{1}{K^2} \sum_{i=1}^K \frac{1}{\alpha_i}. \quad (8)$$

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

$$\min_{\alpha, \gamma} \mathbb{E}_{W_{1:N}, \tilde{Z}} \left[c(W_{1:N}, \psi(\tilde{Z})) + \eta W_c[P_\phi(Z|W_{1:N}), \theta] \right], \quad (9)$$

where $W_{1:N} \sim \mathcal{D}$, $\tilde{Z} \sim P_\phi(Z|W_{1:N})$, $\theta \sim P_\alpha$, c is cross-entropy loss function and W_c is exact Wasserstein distance¹. The sampling process $\theta \sim P_\alpha$ is also relaxed using standard Gaussian reparameterization trick whereby $\theta = \mu(\alpha) + u\Sigma(\alpha)$ with $u \sim \mathcal{N}(0, 1)$.

¹<https://pythonot.github.io/index.html>

Topic Evaluation. In this experiment, we apply OTP-DAG on real-world topic modeling tasks. 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

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

where $\gamma_k = \mu_k(\exp(\beta_k)) + u_k \Sigma_k(\exp(\beta_k))$ and $u_k \sim \mathcal{N}(0, 1)$ is a Gaussian noise. This is realized by using softmax Gaussian trick as in Eq. (8), then applying standard Gaussian reparameterization trick. The optimization procedure follows one described in the previous application.

Table 1: Topics inferred for 3 real-world datasets.

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

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. For OTP-DAG, we set $\tau = 1.0, 2.0$ and $\eta = 1e - 4, 1e - 1$ respectively. The qualitative examples for both applications are given in Figure 2 and Table 1.

C.2 Hidden Markov Models

Poisson Time-series Data Segmentation. 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

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

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

$$Z_1 \sim \text{Cat}\left(\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right), \quad Z_t|Z_{t-1} \sim \text{Cat}\left(\left\{\begin{array}{ll} p & \text{if } Z_t = Z_{t-1} \\ \frac{1-p}{4-1} & \text{otherwise} \end{array}\right\}\right)$$

Let $P(Z_1)$ and $P(Z_t|Z_{t-1})$ respectively denote these prior transition distributions. 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 $\phi(X_t) := \text{Cat}(Z_t)$.

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. Putting everything together, we minimize the following empirical objective

$$\mathbb{E}_{X_{1:T}, \tilde{Z}_{1:T}} \left[c(X_{1:T}, \psi(\tilde{Z}_{1:T})) + \eta \text{KL}[P_\phi(Z_1|X_1), P(Z_1)] + \eta \sum_{t=2}^T \text{KL}[P_\phi(Z_t|X_t), P(Z_t|Z_{t-1})] \right], \quad (10)$$

where $X_{1:T} \sim \mathcal{D}$, $\tilde{Z}_{1:T} \sim P_\phi(Z_{1:T}|X_{1:T})$ and $\psi = [\psi_t]_{t=1}^T$.

In this case, $\theta := \lambda_{1:K}$, $T = 200$, smooth L_1 loss [2] is chosen as the cost function and KL refers to the Kullback-Leibler divergence. We additionally compute MAP estimates of the Poisson rates using stochastic gradient descent, using a log - Normal(5, 5) prior for $p(\lambda)$.

Polyphonic Music Modeling. In this section, we consider another application of HMM to model sequences of polyphonic music. The training set consists of $N = 229$ sequences, each of which has a maximum length of $T = 129$ and $D = 51$ notes. The data matrix is a Boolean tensor of size $N \times T \times D$. The observation at each time step is modeled using a factored observation distribution of the form

$$P(X_t|Z_t = k) = \prod_{d=1}^D \text{Ber}(X_{td}|B_d(k)),$$

where $B_d(k) = P(X_{td} = 1|Z_t = k)$ and $k = 1, \dots, K$.

The transition probabilities are sampled from a Dirichlet distribution with concentration parameters $\alpha_{1:K}$, where $\alpha_k = 1$ if the state remains and 0.1 otherwise

$$Z_1 \sim \text{Cat}(\{1/K\}), \quad Z_t|Z_{t-1} \sim \text{Cat}(p), \quad p \sim \text{Dir}\left(\left\{\begin{array}{ll} 1.0 & \text{if } Z_t = Z_{t-1} \\ 0.1 & \text{otherwise} \end{array}\right\}\right).$$

The parameter set θ is a matrix size $D \times K$ where each element $\theta_{ij} \in [0, 1]$ parameterize $B_d k(\cdot)$. If we view the Bernoulli distribution as a Categorical distribution of 2 categories, one can apply the Gumbel-Softmax trick [4, 6] to relax it into the following forward mapping

$$X_t = \psi_t(Z_t) = \text{Bin-Concrete}(Z_t' \theta),$$

where $Z_t \in \{0, 1\}^K$ is in the one-hot representation, Z_t' is its transpose and the Bin-Concrete function is defined over a binary vector s as follows: with temperature τ , random noises G_{i0} and $G_{i1} \sim G_t = -\log(-\log u_t)$, $u_t \sim \text{Uniform}(0, 1)$,

$$\text{Bin-Concrete}(s) = \frac{\exp\{(\log s + G_{i1})/\tau\}}{\exp\{(\log(1-s) + G_{i0})/\tau\} + \exp\{(\log s + G_{i1})/\tau\}}.$$

A global backward map ϕ is defined as $\phi(X_t) := \text{Cat}(Z_t)$ as in the Poisson HMM, and we learn θ by optimizing Eq. (10) using cross-entropy loss function for c .

Training Configuration. The underlying architecture of the backward maps in both applications is a 3-layer fully connected perceptron. The Poisson HMM is trained for 20,000 epochs with $\eta = 1e-1$ and the Bernoulli HMM is trained for 5,000 epochs on training batches of size 200 at $\eta = 1e-4$. For both applications, we set $\tau = 0.1$.

C.3 Learning Discrete Representations

To understand vector quantized models, let us briefly review Quantization Variational Auto-Encoder (VQ-VAE) [10]. 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 $k = \underset{k}{\operatorname{argmin}} d(Z^m, c_k)$, $Z^m = f_e^m(X)$ 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:

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

where \mathcal{D} is the empirical data, \mathbf{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 [10].

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 k^{th} sub-space (for $k \in \{1, \dots, K\}$), we impose a Gaussian distribution parameterized by μ_k, Σ_k where Σ_k is diagonal. We also endow M discrete distributions over $\mathbf{C}^1, \dots, \mathbf{C}^M$, sharing a common support set as the set of sub-spaces induced by $\{(\mu_k, \Sigma_k)\}_{k=1}^K$:

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

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 $\pi^m = [\pi_1^m, \dots, \pi_K^m]$. In such a practical setting, the generative process is detailed as follows

1. For $m \in \{1, \dots, M\}$,
 - 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 $k = \underset{k}{\operatorname{argmin}} d_z(z^m; \mu_k)$ 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

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

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:

$$\begin{aligned} \min_{\pi, \mu, \Sigma, \theta} \quad & \mathbb{E}_{X \sim \mathcal{D}} \left[c[X, \psi_\theta(Z, \mu_k)] \right] + \frac{\eta}{M} \sum_{m=1}^M [\mathbf{W}_c(P_\phi(Z^m), P(\tilde{Z}^m)) + \mathbf{W}_c(P_\phi(Z^m), \mathbb{P}_{k, \pi^m})] \\ & + \eta_r \sum_{m=1}^M \text{KL}(\pi^m, \mathcal{U}_K), \end{aligned}$$

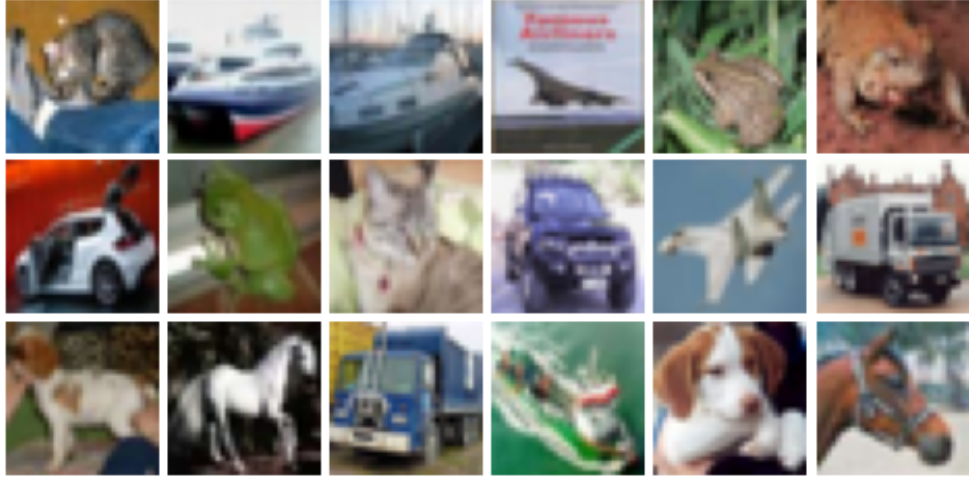
where $P_\phi(Z^m) := f_e^m \# P(X)$ given by the backward ϕ , $P(\tilde{Z}^m) = \sum_{k=1}^K \pi_k^m \mathcal{N}(\tilde{Z}^m | \mu_k, \Sigma_k)$ is the mixture of Gaussian distributions. The copy gradient trick [10] is applied throughout to facilitate backpropagation.

The first term is the conventional reconstruction loss where c is chosen to be mean squared error. Minimizing the second term $\mathbf{W}_c(P_\phi(Z^m), P(\tilde{Z}^m))$ forces the latent representations to follow the Gaussian distribution $\mathcal{N}(\mu_k^m, \Sigma_k^m)$. Minimizing the third term $\mathbf{W}_c(P_\phi(Z^m), \mathbb{P}_{k, \pi^m})$ 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 $\pi_k^m, k = 1, \dots, K$. Therefore, we can use the fourth term $\sum_{m=1}^M \text{KL}(\pi^m, \mathcal{U}_K)$ to guarantee every centroid is utilized.

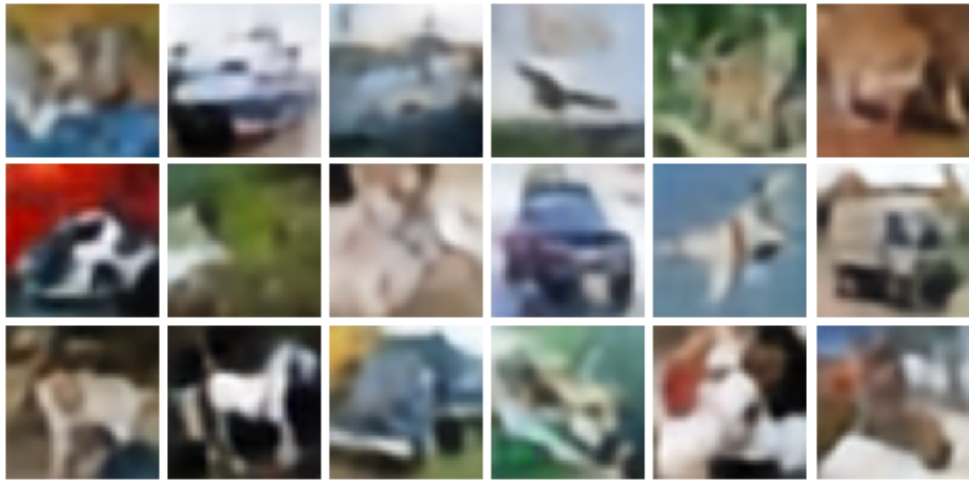
Training Configuration. We use the same experiment setting on all datasets. The models have an encoder with two convolutional layers of stride 2 and filter size of 4×4 with ReLU activation, followed by 2 residual blocks, which contained a 3×3 , stride 1 convolutional layer with ReLU activation followed 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** [11], which calculates the distance between the feature representations of the two images; (4) the dataset-level Fréchet Inception Distance (**FID**) [3], 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.

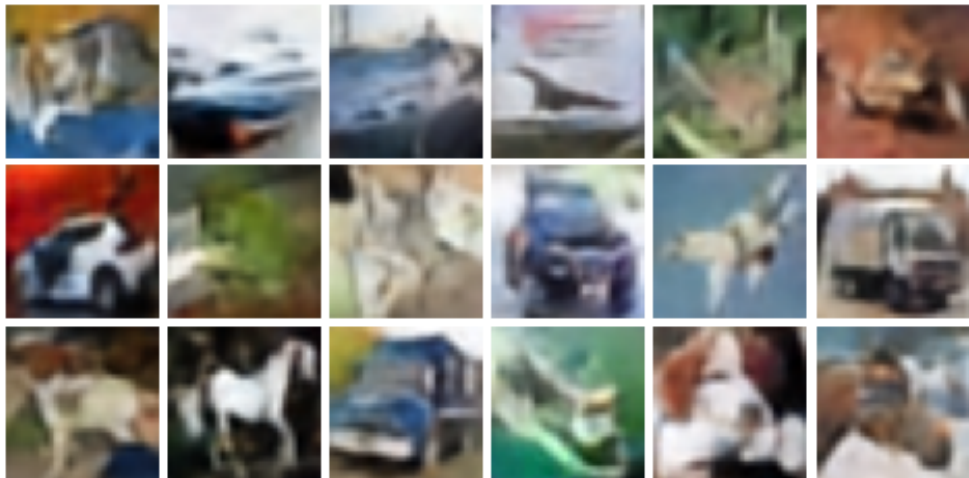
We present the reconstructed samples from CIFAR10 dataset for qualitative evaluation. From Figure 3, it can be seen that the reconstructions from OTP-DAG have higher visual quality than VQ-VAE. The high-level semantic features of the input image and colors are better preserved with OTP-DAG than VQ-VAE from which some reconstructed images are much more blurry.



(a) Original images.



(b) VQ-VAE.



(c) OTP-DAG.

Figure 3: Random reconstructed images from CIFAR10 dataset.

References

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 4
- [2] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 7
- [3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 9
- [4] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 5, 7
- [5] David JC MacKay. Choice of basis for laplace approximation. *Machine learning*, 33:77–86, 1998. 5
- [6] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 5, 7
- [7] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015. 1
- [8] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017. 5
- [9] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. 3
- [10] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 8, 9
- [11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 9