Vector Quantized Wasserstein Auto-Encoder

Tung-Long Vuong [1 2] **Trung Le** [1] **He Zhao** [3] **Chuanxia Zheng** [4] **Mehrtash Harandi** [1] **Jianfei Cai** [1]
Dinh Phung [1 2]

Abstract

Learning deep discrete latent presentations offers a promise of better symbolic and summarized abstractions that are more useful to subsequent downstream tasks. Inspired by the seminal Vector Quantized Variational Auto-Encoder (VQ-VAE), most of work in learning deep discrete representations has mainly focused on improving the original VQ-VAE form and none of them has studied learning deep discrete representations from the generative viewpoint. In this work, we study learning deep discrete representations from the generative viewpoint. Specifically, we endow discrete distributions over sequences of codewords and learn a deterministic decoder that transports the distribution over the sequences of codewords to the data distribution via minimizing a WS distance between them. We develop further theories to connect it with the clustering viewpoint of WS distance, allowing us to have a better and more controllable clustering solution. Finally, we empirically evaluate our method on several well-known benchmarks, where it achieves better qualitative and quantitative performances than the other VQ-VAE variants in terms of the codebook utilization and image reconstruction/generation.

1. Introduction

Learning compact yet expressive representations from large-scale and high-dimensional unlabeled data is an important and long-standing task in machine learning (Kingma & Welling, 2013; Chen et al., 2020; Chen & He, 2021). Among many different kinds of methods, Variational Auto-Encoder (VAE) (Kingma & Welling, 2013) and its variants (Tolstikhin et al., 2017; Alemi et al., 2016; Higgins et al., 2016;

[1]Monash University, Australia [2]Vinai, Vietnam [3]CSIRO's Data61, Australia [4]University of Oxford, United Kingdom. Correspondence to: Tung-Long Vuong <Tung-Long.Vuong@monash.edu>.

Voloshynovskiy et al., 2019) have shown great success in unsupervised representation learning. Although these continuous representation learning methods have been successfully applied to various problems, ranging from images (Pathak et al., 2016; Goodfellow et al., 2014; Kingma et al., 2016), video, and audio (Reed et al., 2017; Oord et al., 2016; Kalchbrenner et al., 2017), in some contexts, input data is more naturally modeled and encoded as discrete symbols rather than continuous ones. For example, discrete representations are a natural fit for complex reasoning, planning, and predictive learning (Van Den Oord et al., 2017). This motivates the need for learning discrete representations while preserving the insightful characteristics of the input data. The Vector Quantization Variational Auto-Encoder (VQ-VAE) (Van Den Oord et al., 2017) is a pioneering generative model that successfully combines the VAE framework with discrete latent representations. In particular, vector quantized models learn a compact discrete representation using a deterministic encoder-decoder architecture in the first stage and subsequently apply this highly compressed representation to various downstream tasks. Examples include image generation (Esser et al., 2021), cross-modal translation (Kim et al., 2022), and image recognition (Yu et al., 2021).

VQ-VAE aims at learning encoder-decoder and a trainable codebook. The *codebook* is formed by set of codewords $C = \{c_k\}_{k=1}^K$ on the latent space $\mathcal{Z} \in \mathbb{R}^{n_z}$ ($C \in \mathbb{R}^{K \times n_z}$). We denote a M-dimensional discrete latent space related to the codebook as the M-ary Cartesian power of C: $C^M \in \mathbb{R}^{M \times n_z}$ with M is the number of components in the latent space. We also denote a latent variable in C^M and its m-th component as $\bar{z}_n \in C^M$ and $\bar{z}_n^m \in C$ respectively. The *encoder* $f_e : \mathbb{R}^{n_x} \to \mathbb{R}^{M \times n_z}$ first map the data examples $x_n \in \mathbb{R}^{n_x}$ to the latent $z_n \in \mathbb{R}^{M \times n_z}$ ($z_n^m = f_e^m(x_n)$ is the m-th component of z_n), followed by a quantization Q_C projecting z_n onto $C^M : \bar{z}_n = Q_C(z_n)$. The quantization process is modelled as a deterministic categorical posterior distribution such that: $\bar{z}_n^m = \mathrm{argmin}_k \rho_z\left(f_e^m(x_n), c_k\right)$ where ρ_z is a metric on the latent space. The *decoder* $f_d : \mathbb{R}^{M \times n_z} \to \mathbb{R}^{n_x}$ reconstructs accurately the data examples from the discrete latent representations.

The objective function of VQ-VAE is as follows:

$$\mathbb{E}_{x \sim \mathbb{P}_x} \left[\begin{array}{c} [d_x\left(f_d\left(Q_C(f_e(x))\right), x\right) \\ +d_z\left(\mathbf{sg}\left(f_e(x)\right), \bar{z}\right) + \beta d_z\left(f_e(x), \mathbf{sg}(\bar{z})\right)] \end{array} \right],$$

1

where $\mathbb{P}_x = \frac{1}{N}\sum_{n=1}^{N}\delta_{x_n}$ is the empirical data distribution, **sg** specifies stop gradient, d_x is a distance on data space, and β is set between 0.1 and 2.0 (Van Den Oord et al., 2017).

While VQ-VAE has been widely applied to representation learning in many areas (Henter et al., 2018; Baevski et al., 2020; Razavi et al., 2019; Kumar et al., 2019; Dieleman et al., 2018; Yan et al., 2021; Hu et al., 2023), it is known to suffer from *codebook collapse*, which has a low codebook usage, *i.e.* most of embedded latent vectors are quantized to just few discrete codewords, while the *other codewords are rarely used, or dead*. This issue arises due to the poor initialization of the codebook, which reduces the information capacity of the bottleneck (Roy et al., 2018; Takida et al., 2022; Yu et al., 2021).

To mitigate this issue, several additional training heuristics were proposed, such as the *exponential moving average* (EMA) update (Van Den Oord et al., 2017; Razavi et al., 2019), soft expectation maximization (EM) update (Roy et al., 2018), codebook reset (Dhariwal et al., 2020; Williams et al., 2020). Notably, the *soft expectation maximization* (EM) update (Roy et al., 2018) connects the EMA update with an EM algorithm and softens the EM algorithm with a stochastic posterior. *Codebook reset* randomly reinitializes unused or low-used codewords to one of the encoder outputs (Dhariwal et al., 2020) or those near codewords of high usage (Williams et al., 2020). Takida et al. (2022) extends the standard VAE by incorporating stochastic quantization and a trainable posterior categorical distribution. Their findings demonstrate that annealing the stochasticity of the quantization process leads to a significant improvement in codebook utilization.

Recently, Wasserstein (WS) distance has been applied successfully to *generative models* and <u>*continuous representation learning*</u> (Arjovsky et al., 2017; Gulrajani et al., 2017; Tolstikhin et al., 2017) owing to its nice properties and theory. It is natural to ask: *"Can we take advantages of intuitive properties of the WS distance and its mature theory for learning compact yet expressive <u>discrete</u> representations?"*

Towards addressing this question, in this paper, we develop solid theories by connecting the theory bodies and viewpoints of the WS distance, generative models, and deep discrete representation learning. In particular, we establish theories for the real and practical setting of learning discrete representation in which a data example $\mathbf{X}$ is mapped to a sequence of M latent codes $\mathbf{Z} = [\mathbf{Z}^1, \ldots, \mathbf{Z}^M]$ corresponding to a sequence of M codewords $\mathbf{C} = [\mathbf{C}^1, \ldots, \mathbf{C}^M]$ via an encoder f_e. Our theory development pathway is as follows. We first endow M discrete distributions over $\mathbf{C}^1, \ldots, \mathbf{C}^M$, sharing a common support set as the set of codewords $C = [c_k]_{k=1}^{K} \in \mathbb{R}^{K \times n_z}$. We then use a joint distribution γ, admitting these discrete distributions over $\mathbf{C}^1, \ldots, \mathbf{C}^M$ as its marginal distributions to sample a se-

quence of M codewords $\mathbf{C} = [\mathbf{C}^1, \ldots, \mathbf{C}^M]$. From the generative viewpoint, we propose learning a decoder f_d to minimize the *codebook-data distortion* as the WS distance: $\mathcal{W}_{d_z}(f_d \# \gamma, \mathbb{P}_x)$ (cf. (1)).

Subsequently, we develop rigorous theories to equivalently turn the formulation in the generative viewpoint to a trainable form in Theorem 2.3, engaging the deterministic encoder f_e to minimize the reconstruction error and a WS distance between the distribution over sequences of latent codes $[\mathbf{Z}^1, \ldots, \mathbf{Z}^M]$ and the optimal γ over $[\mathbf{C}^1, \ldots, \mathbf{C}^M]$. Additionally, this WS distance is further proven to equivalently decompose into the sum of M WS distances between each $\mathbf{Z^m}$ and $\mathbf{C}^m, m = 1, \ldots, M$. Interestingly, in Corollary 2.5, we prove that when minimizing the WS distance between the latent code $\mathbf{Z}^m$ and codeword $\mathbf{C}^m$, the codewords tend to flexibly move to the clustering centroids of the latent representations with a control on the proportion of latent representations associated to a centroid. We argue and empirically demonstrate that using the clustering viewpoint of a WS distance to learn the codewords, we can obtain more *controllable* and *better centroids* than using a simple k-means as in VQ-VAE (cf. Sections 2.1 and 4.2).

Moreover, we leverage the developed theory to propose a practical method called *Vector Quantized Wasserstein Auto-Encoder* (VQ-WAE), which utilizes the WS distance to learn a more controllable codebook, resulting in improved the codebook utilization. We conduct comprehensive experiments to demonstrate our key contributions by comparing with VQ-VAE (Van Den Oord et al., 2017) and SQ-VAE (Takida et al., 2022) (*i.e.*, the recent work that can improve the codebook utilization). The experimental results show that our VQ-WAE can achieve better codebook utilization with higher codebook perplexity, hence leading to lower (compared with VQ-VAE) or comparable (compared with SQ-VAE) reconstruction error, with significantly lower reconstructed Fréchlet Inception Distance (FID) score (Heusel et al., 2017). Generally, a better quantizer in the stage-1 can naturally contribute to stage-2 downstream tasks (Yu et al., 2021; Zheng et al., 2022). To further demonstrate this, we conduct comprehensive experiments on four benchmark datasets. The experimental results indicate that from the codebooks of our VQ-WAE, we can generate better images with lower FID scores.

Our contributions in this paper can be summarized:

- We are the first work that studies learning discrete representations from the generative viewpoint. Subsequently, we develop rigorous and comprehensive theories that equivalently transform the formulation in the generative viewpoint into another trainable form involving a reconstruction term and a WS distance alignment between the latent representations and learnable codewords.

- We harvest our theory development to propose the practical method, namely VQ-WAE, that can learn more controllable codebook for improving the codebook utilization and reconstruct/generate better images with lower FID scores.

2. Vector Quantized Wasserstein Auto-Encoder

We present the theoretical development of our VQ-WAE framework, which connects the viewpoints of the WS distance, generative models, and deep discrete representation learning in Section 2.1. It is important to note that our theories are specifically developed for the real setting of discrete representation learning, where a deterministic decoder maps a data example to a sequence of latent codes corresponding to a sequence of codewords. This poses a significant challenge in theory development. Based on the theoretical development, we devise a practical algorithm for VQ-WAE in Section 2.2. All proofs can be found in Appendix A.

2.1. Theoretical Development

Given a training set $\mathbb{D} = \{x_1, ..., x_N\} \subset \mathbb{R}^{n_x}$, we wish to learn a set of *codewords* $C = \{c_k\}_{k=1}^{K} \subset \mathbb{R}^{K \times n_z}$ on a latent space $\mathcal{Z}$ and *an encoder* to map each data example to a sequence of M codewords, preserving insightful characteristics carried in the data. We now endow M discrete distributions:

$$\mathbb{P}_{c,\pi^m} = \sum_{k=1}^{K} \pi_k^m \delta_{c_k}, m = 1, \dots, M$$

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.

We denote $\Gamma = \Gamma(\mathbb{P}_{c,\pi^1}, ..., \mathbb{P}_{c,\pi^M})$ as the set of all joint distributions over sequences of M codewords, admitting $\mathbb{P}_{c,\pi^1}, \dots, \mathbb{P}_{c,\pi^M}$ as its marginal distributions. Let also define $\pi = [\pi^1, \dots, \pi^M]$ as the set of all weights.

From the generative viewpoint, we propose to learn a decoder function $f_d : \mathcal{Z}^M \to \mathcal{X}$ (i.e., mapping from $\mathcal{Z}^M$ with the latent space $\mathcal{Z} \subset \mathbb{R}^{n_z}$ to the data space $\mathcal{X}$), the codebook C, and the weights π, to minimize:

$$\min_{C,\pi} \min_{\gamma \in \Gamma} \min_{f^d} \mathcal{W}_{d_x} (f_d \# \gamma, \mathbb{P}_x), \quad (1)$$

where $\mathbb{P}_x = \frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}$ is the empirical data distribution and d_x is a cost metric on the data space.

We interpret the optimization problem (OP) in Eq. (1) as follows. Given discrete distributions $\mathbb{P}_{c,\pi^{1:M}}$, we employ a joint distribution $\gamma \in \Gamma$ as a distribution over sequences of M codewords in C^M. We then use the decoder f_d to map the sequences of M codewords in C^M to the data space and

consider $\mathcal{W}_{d_x} (f_d \# \gamma, \mathbb{P}_x)$ as the *codebook-data distortion* w.r.t. f_d and γ. We subsequently learn f_d to minimize the codebook-data distortion given γ and finally adjust the codebook C, π, and γ to minimize the optimal codebook-data distortion. To offer more intuition for the OP in Eq. (1), we introduce the following lemma.

Lemma 2.1. *Let $C^* = \{c_k^*\}_k$, π^*, γ^*, and f_d^* be the optimal solution of the OP in Eq. (1). Assume $K^M < N$, then $C^* = \{c_k^*\}_k$, π^*, and f_d^* are also the optimal solution of the following OP:*

$$\min_{f_d} \min_{\pi} \min_{\sigma_{1:M} \in \Sigma_\pi} \sum_{n=1}^{N} d_x \left(x_n, f_d \left([c_{\sigma_m(n)}]_{m=1}^M \right) \right), \quad (2)$$

where Σ_π is the set of assignment functions σ : $\{1, ..., N\} \to \{1, ..., K\}$ such that for every m the cardinalities $\left| \sigma_m^{-1}(k) \right|$, $k = 1, ..., K$ are proportional to π_k^m, $k = 1, ..., K$. Here we denote $\sigma_m^{-1}(k) = \{n \in [N] : \sigma_m(n) = k\}$ with $[N] = \{1, 2, ..., N\}$.

Lemma 2.1 states that for the optimal solution $C^* = \{c_k^*\}$, π^*, $\sigma_{1:M}^*$, and f_d^* of the OP in (1), each x_n is assigned to the centroid $f_d^*([c_{\sigma_m^*}(n)]_{m=1}^M)$ which forms optimal clustering centroids of the optimal clustering solution minimizing the distortion. We establish the following theorem to engage the OP in (1) with the latent space.

Theorem 2.2. *We can equivalently turn the optimization problem in (1) to*

$$\min_{C,\pi,f_d} \min_{\gamma \in \Gamma} \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x} \left[d_x \left(f_d \left(\bar{f}_e(x) \right), x \right) \right], \quad (3)$$

*where $\bar{f}_e$ is a **deterministic discrete** encoder mapping data example x directly to a sequence of M codewords in C^M.*

Theorem 2.2 can be interpreted as follows. First, we learn both the codebook C and the weights π. Next, we glue the codebook distributions $\mathbb{P}_{c,\pi^m}, m = 1, \dots, M$ using the joint distribution $\gamma \in \Gamma$. Subsequently, we seek a *deterministic discrete* encoder $\bar{f}_e$ mapping data example x to sequence of M codewords drawn from γ, concurring with vector quantization and serving our further derivations. Finally, we minimize the reconstruction error of the sequence of M codewords corresponding to $\bar{f}_e(x)$ and x.

Additionally, $\bar{f}_e$ is a deterministic discrete encoder mapping a data example x directly to a sequence of codewords. To make it trainable, we replace $\bar{f}_e$ by a continuous encoder $f_e : \mathcal{X} \to \mathcal{Z}^M$ with $f_e(x) = [f_e^m(x)]_{m=1}^M$ (i.e., each $f_e^m : \mathcal{X} \to \mathcal{Z}$) in the following theorem.

Theorem 2.3. *If we seek f_d and f_e in a family with infinite capacity (e.g., the family of all measurable functions), the the two OPs of interest in (1) and (3) are equivalent to the following OP*

$$\min_{C,\pi} \min_{\gamma \in \Gamma} \min_{f_d,f_e} \left\{ \begin{array}{c} \mathbb{E}_{x \sim \mathbb{P}_x} [d_x (f_d (Q_C (f_e(x))), x)] \\ + \lambda \mathcal{W}_{d_z} (f_e \# \mathbb{P}_x, \gamma) \end{array} \right\}, \quad (4)$$

where $Q_C\left(f_e\left(x\right)\right) = \left[Q_C(f_e^m\left(x\right))\right]_{m=1}^M$ with $Q_C(f_e^m\left(x\right)) = argmin_{c \in C}\rho_z\left(f_e^m\left(x\right), c\right)$ is a quantization operator which returns the sequence of closest codewords to $f_e^m\left(x\right), m = 1, \ldots, M$ and the parameter $\lambda > 0$. Here we overload the quantization operator for both $f_e(x) \in \mathcal{Z}^M$ and $f_e^m(x) \in \mathcal{Z}$. Additionally, given $z = [z^m]_{m=1}^M \in \mathcal{Z}^M, \bar{z} = [\bar{z}^m]_{m=1}^M \in \mathcal{Z}^M$, the distance between them is defined as

$$d_z\left(z, \bar{z}\right) = \frac{1}{M}\sum_{m=1}^M \rho_z\left(z^m, \bar{z}^m\right),$$

where ρ_z is a distance on $\mathcal{Z}$.

Particularly, we rigorously prove that the OPs of interest in (1), (3), and (4) are equivalent under some mild conditions in Theorem 2.3. This rationally explains why we could solve the OP in (4) for our final tractable solution. Moreover, the OP in (4) conveys important meaningful interpretations. Specifically, by minimizing $\mathcal{W}_{d_z}\left(f_e\#\mathbb{P}_x, \gamma\right)$ w.r.t. C, π where γ admits $\mathbb{P}_{c,\pi^{1:M}}$ as its marginal distributions, we implicitly minimize $\mathcal{W}_{\rho_z}(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}), m = 1, \ldots, M$ due to the fact that the former is an upper-bound of the latter as in Lemma 2.4. Furthermore, in Lemma 2.4, we also develop a close form for the WS distance of interest, hinting us a practical method.

Lemma 2.4. *The Wasserstein distance of interest* $\min_\pi \min_{\gamma \in \Gamma} \mathcal{W}_{d_z}\left(f_e\#\mathbb{P}_x, \gamma\right)$ *is upper-bounded by*

$$\frac{1}{M}\sum_{m=1}^M \mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right). \quad (5)$$

According to Lemma 2.4, the OP of interest in (4) can be replaced by minimizing its upper-bound as follows

$$\min_{C,\pi} \min_{f_d,f_e} \left\{ \begin{array}{l} \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(Q_C\left(f_e\left(x\right)\right)\right), x\right)\right] \\ + \frac{\lambda}{M}\sum_{m=1}^M \mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right) \end{array} \right\}. \quad (6)$$

We now interpret the WS term $\mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right)$ in Corollary 2.5.

Corollary 2.5. *Given* $m \in [M]$, *consider minimizing the term:* $\min_{f_e,C} \mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right)$ *in (4), given* π^m *and assume* $K < N$, *its optimal solution* f_e^{*m} *and* C^* *are also the optimal solution of the OP:*

$$\min_{f_e,C} \min_{\sigma\in\Sigma_\pi} \sum_{n=1}^N \rho_z\left(f_e^m\left(x_n\right), c_{\sigma(n)}\right), \quad (7)$$

where Σ_π *is the set of assignment functions* $\sigma : \{1, ..., N\} \rightarrow \{1, ..., K\}$ *such that the cardinalities* $\left|\sigma^{-1}(k)\right|, k = 1, ..., K$ *are proportional to* $\pi_k^m, k = 1, ..., K$.

Corollary 2.5 indicates the aim of minimizing the second term $\mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right)$. By which, we adjust the encoder f_e and the codebook C such that the codewords of C become the clustering centroids of the latent representations $\{f_e^m\left(x_n\right)\}_n$ to minimize the *codebook-latent distortion*. Additionally, at the optimal solution, the optimal assignment function σ^*, which indicates how latent representations (or data examples) associated with the clustering centroids (i.e., the codewords) has a valuable property, i.e., *the cardinalities* $\left|(\sigma^*)^{-1}(k)\right|, k = 1, ..., K$ *are proportional to* $\pi_k^m, k = 1, ..., K$.

Remark: Recall the codebook collapse issue, i.e. most of embedded latent vectors are quantized to just few discrete codewords while the other codewords are rarely used. Corollary 2.5 give us important properties: **(1)** *we can control the number of latent representations assigned to each codeword by adjust* π^m, *guaranteeing all codewords are utilized*, **(2)** *codewords become the clustering centroids of the associated latent representations to minimize the codebook-latent distortion*, to develop our VQ-WAE framework. Particularly, we propose adding the regularization terms $D_{KL}(\pi^m, \mathcal{U}_K)$ as the Kullback-Leibler divergence between π^m and the uniform distribution $\mathcal{U}_K = [\frac{1}{K}]_K$ to regularize π^m.

2.2. Practical Algorithm for VQ-WAE

We now harvest our theoretical development to propose a practical method named *Vector Quantized Wasserstein Auto-Encoder* (VQ-WAE). Particularly, we combine the objective function in (6) with the regularization terms $D_{KL}(\pi^m, \mathcal{U}_K), m = 1, \ldots, M$ and $\mathcal{U}_K = \left[\frac{1}{K}\right]_K$ inspired by Corollary 2.5 to arrive at the following OP:

$$\min_{C,\pi,f_d,f_e} \left\{ \begin{array}{l} \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(Q_C\left(f_e\left(x\right)\right)\right), x\right)\right] \\ + \frac{\lambda}{M}\times\sum_{m=1}^M \mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right) \\ + \lambda_r\sum_{m=1}^M D_{KL}(\pi^m, \mathcal{U}_K) \end{array} \right\}, \quad (8)$$

where $\lambda, \lambda_r > 0$ are two trade-off parameters.

To learn the weights π^m, we parameterize $\pi^m = \pi^m(\beta^m) = \text{softmax}(\beta^m), m = 1, \ldots, M$ with $\beta^m \in \mathbb{R}^K$. Additionally, in order to optimize (8), we have to deal with M WS distances $\mathcal{W}_{\rho_z}\left(f_e^m\#\mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right)$ with $m = 1, ..., M$. Therefore, we proposed to use entropic dual form of optimal transport (Genevay et al., 2016) which enable us to compute these WS distances in parallel by matrix computation from current deep learning framework.

At each iteration, we sample a mini-batch $x_1, ..., x_B$ and then solve the above OP by updating f_d, f_e and $C, \beta^{1..M}$ based on this mini-batch as follows. Let us denote

$$\mathbb{P}_B = \frac{1}{B}\sum_{i=1}^B \delta_{x_i}$$

as the empirical distribution over the current batch.

For each mini-batch, we replace $\mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c,\pi^m}\right)$ by $\mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_B, \mathbb{P}_{c,\pi^m}\right)$ and approximate it with entropic regularized duality form $\mathcal{R}_{WS}^m$ (see Eq. (27) in Appendix B) as follows:

$$
\begin{aligned}
\mathcal{R}_{WS}^m = \max_{\phi^m} \Bigg\{ \frac{1}{B} \sum_{i=1}^{B} \Bigg[-\epsilon \log \Bigg(\sum_{k=1}^{K} \pi_k^m \Bigg[\exp\Bigg\{ \\
\frac{-\rho_z(f_e^m(x_i), c_k) + \phi^m(c_k)}{\epsilon} \Bigg\} \Bigg] \Bigg) \Bigg] + \sum_{k=1}^{K} \pi_k^m \phi^m(c_k) \Bigg\}
\end{aligned}
\tag{9}
$$

where ϕ^m is the Kantorovich potential network.

Substituting (9) into (8), we reach final OP to update $f_d, f_e, C, \{\beta^m\}_{m=1}^M$ for each mini-batch:

$$
\min_{C, \{\beta^m\}_{m=1}^M} \min_{f_d, f_e} \left\{ \begin{array}{c} \frac{1}{B} \sum_{i=1}^{B} d_x\left(f_d\left(Q\left(f_e(x_i)\right)\right)\right) \\ + \frac{\lambda}{M} \sum_{m=1}^{M} \mathcal{R}_{WS}^m \\ + \lambda_r \sum_{m=1}^{M} D_{KL}\left(\pi^m(\beta^m), \mathcal{U}_K\right) \end{array} \right\}.
\tag{10}
$$

We use the copy gradient trick (Van Den Oord et al., 2017) to deal with the back-propagation from decoder to encoder for reconstruction term. The pseudocode of our VQ-WAE is summarized in Algorithm 1.

Algorithm 1 VQ-WAE

1: **Initialize**: encoder f_e, decoder f_d, codebook C and $\{\pi^m = \text{softmax}(\beta^m), \phi^m\}_{m=1}^M$.
2: **for** iter **in** batch-iterations **do**
3:     Sample a mini-batch of samples $x_1, ..., x_B$ forming the empirical batch distribution $\mathbb{P}_B$.
4:     Encode: $z_{1...B} = f_e(x_{1...B})$
5:     Quantize: $\bar{z}_{1..B} = Q_C(z_{1...B})$
6:     Decode: $\bar{x}_{1...B} = f_d(\bar{z}_{1...B})$
7:     **for** iter **in** ϕ-iterations **do**
8:         Optimize $\{\phi^m\}_{m=1}^M$ by maximizing the objective in (9).
9:     **end for**
10:    Optimize $f_e, f_d, \{\beta^m\}_{m=1}^M$ and C by minimizing the objective in (10).
11: **end for**
12: **Return:** The optimal f_e, f_d and C.

---

3. Related Work

The Variational Auto-Encoder (VAE) was initially introduced by Kingma & Welling (2013) for learning continuous representations. However, learning discrete latent representations has proven to be much more challenging due to the difficulty of accurately evaluating the gradients required for training the models. To make the gradients tractable, one possible solution is to apply the Gumbel Softmax reparameterization trick (Jang et al., 2016) to VAE, which allows us to estimate stochastic gradients for updating the models. Although this technique provides gradients with low variance, it introduces a high-bias gradient estimator. Another possible solution is to employ the REINFORCE algorithm (Williams, 1992), which is unbiased but has a high variance. Furthermore, these two techniques can be combined in a complementary manner (Tucker et al., 2017).

To facilitate the learning of discrete latent codes, VQ-VAE (Van Den Oord et al., 2017) employs a deterministic encoder/decoder architecture and encourages the codebooks to represent the clustering centroids of the latent representations. Additionally, the copy gradient trick is utilized to back-propagate gradients from the decoder to the encoder (Bengio, 2013). Several subsequent works have extended VQ-VAE, notably Roy et al. (2018); Wu & Flierl (2020).Particularly, Roy et al. (2018) uses the Expectation Maximization (EM) algorithm in the bottleneck stage to train the VQ-VAE for improving the quality of the generated images. However, to maintain the stability of this approach, we need to collect a large number of samples on the latent space. Wu & Flierl (2020) imposes noises on the latent codes and uses a Bayesian estimator to optimize the quantizer-based representation. The introduced bottleneck Bayesian estimator outputs the posterior mean of the centroids to the decoder and performs soft quantization of the noisy latent codes which have latent representations preserving the similarity relations of the data space. Recently, Takida et al. (2022) extends the standard VAE with stochastic quantization and trainable posterior categorical distribution, showing that the annealing of the stochasticity of the quantization process significantly improves the codebook utilization.

Wasserstein (WS) distance has been widely used in various problems (Zhao et al., 2021; Nguyen et al., 2021a;b; Le et al., 2021; Bui et al., 2022), especially in generative models (Arjovsky et al., 2017; Gulrajani et al., 2017; Tolstikhin et al., 2017; Dam et al., 2019). In their work, Arjovsky et al. (2017) utilized a dual form of the WS distance to develop the Wasserstein generative adversarial network (WGAN). Subsequently, Gulrajani et al. (2017) introduced the gradient penalty trick to enhance the stability of WGAN. In terms of theory development, mostly related to our work is Wasserstein Auto-Encoder (Tolstikhin et al., 2017), which focuses on learning continuous latent representations while preserving the characteristics of the input data.

4. Experiments

Datasets: We empirically evaluate the proposed VQ-WAE in comparison with VQ-VAE (Van Den Oord et al., 2017) that is the baseline method, VQ-GAN (Esser et al., 2021) and recently proposed SQ-VAE (Takida et al., 2022) which

is the state-of-the-art work of improving the codebook usage, on five different benchmark datasets: CIFAR10 (Van Den Oord et al., 2017), MNIST (Deng, 2012), SVHN (Netzer et al., 2011), CelebA dataset (Liu et al., 2015; Takida et al., 2022) and the high-resolution images dataset FFHQ.

Implementation: For a fair comparison, we utilize the same architectures and hyperparameters for all methods. Additionally, in the primary setting, we use a codeword (discrete latent) dimensionality of 64 and codebook size $|C| = 512$ for all datasets except FFHQ, which has a codeword dimensionality of 256 and codebook size $|C| = 1024$, while the hyper-parameters $\{\beta, \tau, \lambda\}$ are specified as presented in the original papers, *i.e.*, $\beta = 0.25$ for VQ-VAE and VQ-GAN (Esser et al., 2021), $\tau = 1e^{-5}$ for SQ-VAE and $\lambda = 1e^{-3}$, $\lambda_r = 1.0$ for our VQ-WAE. The details of the experimental settings are presented in Appendix D.

4.1. Results on Benchmark Datasets

Quantitative assessment: In order to quantitatively assess the quality of the reconstructed images, we report the results on most common evaluation metrics, including the pixel-level peak signal-to-noise ratio (PSNR), patch-level structure similarity index (SSIM), feature-level LPIPS (Zhang et al., 2018), and dataset-level Fréchlet Inception Distance (FID) (Heusel et al., 2017). We report the test-set reconstruction results on four datasets in Table 1. With regard to the codebook utilization, we employ perplexity score which is defined as $e^{-\sum_{k=1}^{K} p_{c_k} \log p_{c_k}}$ where $p_{c_k} = \frac{N_{c_k}}{\sum_{i=1}^{K} N_{c_i}}$ (i.e., N_{c_i} is the number of latent representations associated with the codeword c_i) is the probability of the ith codeword being used. Note that by formula, $\text{perplexity}_{\max} = |C|$ as $P(c)$ becomes to the uniform distribution, which means that all the codewords are utilized equally by the model.

We compare VQ-WAE with VQ-VAE, SQ-VAE and VQ-GAN for image reconstruction in Table 1. All instantiations of our model significantly outperform the baseline VQ-VAE under the same compression ratio, with the same network architecture. While the latest state-of-the-art SQ-VAE or VQ-GAN holds slightly better scores for traditional pixel- and patch-level metrics, our method achieves much better rFID scores which evaluate the image quality at the dataset level. Note that our VQ-WAE significantly improves the perplexity of the learned codebook. This suggests that the proposed method significantly improves the codebook usage, resulting in better reconstruction quality. which is further demonstrated in the following qualitative assessment.

Qualitative assessment: We present the reconstructed samples from FFHQ (high-resolution images) for qualitative evaluation. It can be clearly seen that the high-level semantic
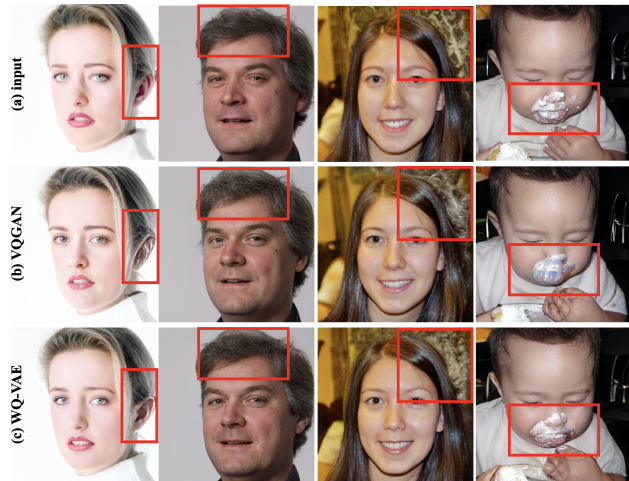

Figure 1: Reconstruction results for the FFHQ dataset.

features of the input image and colors are better preserved with VQ-WAE than the baseline. Particularly, we notice that VQ-GAN often produces repeated artifact patterns in image synthesis (see the hair of man is second column in Figure 1) while VQ-WAE does not. This is because VQ-GAN is lack of diversity in the codebook, which will be further analyzed in Section 4.2.1. Consequently, the quantization operator embeds similar patches into the same quantization index and ignores the variance in these patches (e.g., VQ-GAN reconstructs the background in third column of Figure 1 as hair of woman).

4.2. Detailed Analysis

We run a number of ablations to analyze the properties of VQ-VAE, SQ-VAE and VQ-WAE, in order to assess if our VQ-WAE can simultaneously achieve (i) *efficient codebook usage*, (ii) *reasonable latent representation*.

4.2.1. CODEBOOK USAGE

We observe the codebook utilization of three methods with different codebook sizes $\{64, 128, 256, 512\}$ on MNIST and CIFAR10 datasets. Particularly, we present the reconstruction performance for different settings in Table 2 and the histogram of latent representations over the codebook in Figure 2. As discussed in Section 2.1, the number of used centroids reflects the capability of the latent representations. In other words, it represents the certain amount of information is preserved in the latent space.

It can be seen from Figure 2 that the latent distribution of VQ-WAE over the codebook is nearly uniform and the codebook's perplexity almost reaches the optimal value (i.e., the value of perplexities reach to corresponding codebook sizes) in different settings. *It is also observed that as the size of the codebook increases, the perplexity of codebook*

Table 1: Reconstruction performance (↓: the lower the better and ↑: the higher the better).

Dataset	Model	Latent Size	SSIM ↑	PSNR ↑	LPIPS ↓	rFID ↓	Perplexity ↑
CIFAR10	VQ-VAE	8×8	0.70	23.14	0.35	77.3	69.8
	SQ-VAE	8×8	**0.80**	**26.11**	**0.23**	55.4	434.8
	VQ-WAE	8×8	**0.80**	25.93	**0.23**	**54.3**	**497.3**
MNIST	VQ-VAE	8×8	0.98	33.37	0.02	4.8	47.2
	SQ-VAE	8×8	**0.99**	**36.25**	**0.01**	3.2	301.8
	VQ-WAE	8×8	**0.99**	35.71	**0.01**	**2.33**	**508.4**
SVHN	VQ-VAE	8×8	0.88	26.94	0.17	38.5	114.6
	SQ-VAE	8×8	**0.96**	**35.37**	**0.06**	24.8	389.8
	VQ-WAE	8×8	**0.96**	34.62	0.07	**23.4**	**485.1**
CELEBA	VQ-VAE	16×16	0.82	27.48	0.19	19.4	48.9
	SQ-VAE	16×16	**0.89**	**31.05**	0.12	14.8	427.8
	VQ-WAE	16×16	**0.89**	30.60	**0.11**	**12.2**	**503.0**
FFHQ	VQ-GAN	16×16	0.6641	22.24	**0.1175**	4.42	423
	VQ-WAE	16×16	**0.6648**	**22.45**	0.1245	**4.20**	**1022**

Table 2: Distortion and Perplexity with different codebook sizes.

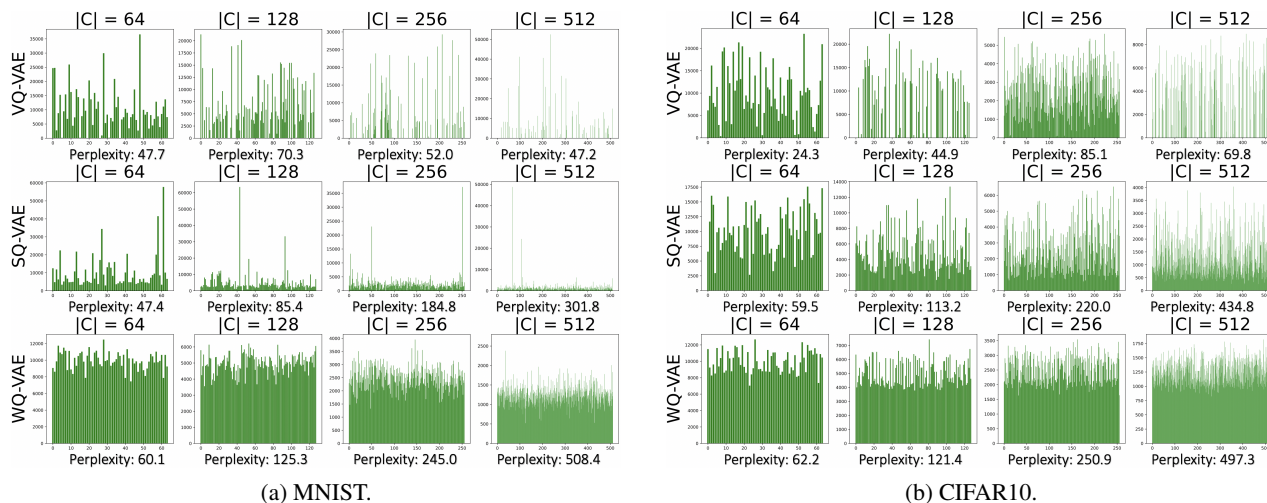| Dataset		MNIST				CIFAR10			
$|C|$		64	128	256	512	64	128	256	512
VQ-VAE	Perplexity	47.8	70.3	52.0	47.2	24.3	44.9	85.1	69.8
	rFID	5.9	6.2	5.2	4.8	86.6	78.9	73.6	69.8
SQ-VAE	Perplexity	47.4	85.4	184.8	301.8	59.5	113.2	220.0	434.8
	rFID	**4.7**	4.3	3.5	3.2	**71.5**	**66.9**	62.6	55.4
VQ-WAE	Perplexity	**60.1**	**125.3**	**245.0**	**508.4**	**62.2**	**121.4**	**250.9**	**497.3**
	rFID	5.6	**3.9**	**2.8**	**2.3**	73.5	68.2	**60.5**	**54.3**

(a) MNIST.

(b) CIFAR10.

Figure 2: Latent distribution over the codebook on test-set.

of VQ-WAE also increases, leading to the better reconstruction performance (Table 2), in line with the analysis in (Wu & Flierl, 2018). SQ-VAE also has good codebook utilization as its perplexity is proportional to the size of the codebook. However, it becomes less efficient when the codebook size becomes large, especially in low texture dataset. (i.e., MNIST). On the contrary, the codebook usage of VQ-VAE is less efficient, i.e., there are many zero entries in its codebook usage histogram, indicating that some codewords have never been used (Figure 2). Furthermore, Table 2 also shows the instability of VQ-VAE's reconstruction performance with different codebook sizes.

4.2.2. CONTROLLABILITY OF CODEBOOK

To further underscore the codebook-controllability of VQ-WAE, we proceed to perform the following ablations. Firstly, additional experiments are conducted involving different initializations of π^m, specifically including Peaked-form (P), Gaussian-form (G), and Uniform-form (U). Our objective is to observe whether the latent distributions over the codebook, obtained after training with a fixed π^m configuration, exhibit proportionality to the initial π^m, thereby effectively demonstrating the controllability. Secondly, we investigate the implications of optimizing π^m as opposed to maintaining a fixed state throughout the training process.
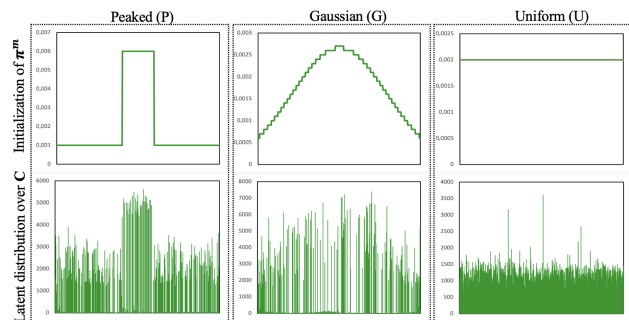

Figure 3: *Top*. Different initialization of Codebook; *Bottom*. Latent distribution over the codebook C with fixed π^m.

Figure 3 provides evidence indicating that the latent distributions over the codebook exhibit proportionality to the initial π^m, thereby serving as a demonstration of the controllability of VQ-WAE's codebook. However, it is important to note that our primary objective is to learn latent representations that accurately approximate the true underlying latent distribution of the data. Consequently, if we have prior knowledge of the true underlying latent distribution of the data, it would be optimal to fix π^m accordingly. Nonetheless, in practical scenarios, the true underlying distribution of the data is typically unknown. If the initial π^m significantly deviates from the true underlying distribution, it can adversely affect the model's performance. Hence, it is imperative to optimize π^m during training process.

Table 3: Reconstruction performance with different codebook initializations (PPL - Perplexity).

π^m	Metric	P	G	U
Fixed	rFID	63.77	68.87	56.06
Fixed	PPL	229.4	165.1	**502.6**
Updated, $\lambda_r = 0.0$	rFID	62.04	62.16	57.49
Updated, $\lambda_r = 0.0$	PPL	292.5	285.6	456.5
Updated, $\lambda_r = 1.0$	rFID	60.60	60.31	**54.30**
Updated, $\lambda_r = 1.0$	PPL	410.0	442.8	497.3

In such cases, π^m will be gradually updated to match the latent distribution. Therefore, our intuition is to initialize π^m with a distribution that can easily adapt to arbitrary distributions. The results presented in Table 3 indicate that a uniform initialization is a suitable choice for π^m.

It is worth noting that the motivation behind employing KL-regularization is to encourage the utilization of every discrete codeword, thus avoiding the occurrence of certain π_k^m values becoming zero (additional discussion regarding the motivation of KL-regularization can be found in Appendix C). This feature of VQ-WAE is unique as it allows for the reflection of the latent distribution and enables control over it. Consequently, the Wasserstein distance with KL-regularization in Objective (8) serves to match the codebook distribution with the latent data distribution, while also ensuring the utilization of all codewords. This guarantees the robustness of the model.
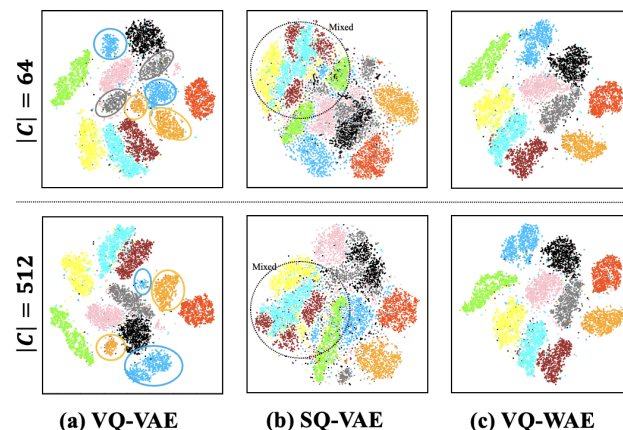
4.2.3. VISUALIZATION OF LATENT REPRESENTATION

(a) VQ-VAE   **(b) SQ-VAE**   **(c) VQ-WAE**

Figure 4: The t-SNE feature visualization on the MNIST dataset (different colors for different digits).

T-SNE visualization. To better understand the codebook's representation power, we employ t-SNE (van der Maaten &

Hinton, 2008) to visualize the latent that have been learned by VQ-VAE, SQ-VAE and VQ-WAE on the MNIST dataset with two codebook sizes of 64 and 512. Figure 4 shows the latent distributions of different classes in the latent space, in which the samples are colored accordingly to their class labels. Figure 4c shows that representations from different classes of VQ-WAE are well clustered (i.e., each class focuses on only one cluster) and clearly separated to other classes. In contrast, the representations of some classes in VQ-VAE and SQ-VAE are distributed to several clusters and or mixed to each other (Figure 4a,b). Moreover, the class-clusters of SQ-VAE are uncondensed and tend to overlap with each other. These results suggest that the representations learned by VQ-WAE can better preserve the similarity relations of the data space better than the other baselines.

Single-layer Classification on latent space. We train a separate single-layer classifier using the latent representation from auto-encoders (VQ-VAE, SQ-VAE and VQ-WAE) as input. We did not optimize autoencoder's parameters with respect to the classifier's loss to measure the unsupervised representation learning performance of auto-encoders.

Table 4: Single-layer classification accuracy on latent space.

Dataset	VQ-VAE	SQ-VAE	VQ-WAE
Cifar10	43.21	46.17	**50.19**
Mnist	95.12	94.48	**95.62**
SVHN	35.10	36.73	**38.38**

It can be seen from Table 4 that VQ-WAE obtained higher performance compared to SQ-VAE, further demonstrating the better quality of a learned representation of VQ-WAE.

4.2.4. IMAGE GENERATION

As discussed in the previous section, VQ-WAE is able to optimally utilize its codebook, leading to meaningful and diverse codewords that naturally improve the image generation. To confirm this ability, we perform the image generation on the benchmark datasets. Since the decoder reconstructs images directly from the discrete embeddings, we only need to model a prior distribution over the discrete latent space (i.e., codebook) to generate images. We employ a conventional autoregressive model, the CNN-based Pixel-CNN (Van den Oord et al., 2016), to estimate a prior distribution over the discrete latent space of VQ-VAE, SQ-VAE and VQ-WAE on CIFAR10, MNIST, SVHN and CelebA. The details of generation settings are presented in Section 3.2 of the supplementary material. The quantitative results in Table 5 indicate that the codebook of VQ-WAE leads to a better generation ability baselines.

Table 5: FID scores of unconditional (U) and class-conditional (C) generated images.

Dataset	Model	Latent size	U	C
CIFAR10	VQ-VAE	8×8	117.49	117.16
	SQ-VAE	8×8	103.78	90.74
	VQ-WAE	8×8	**87.73**	**88.51**
MNIST	VQ-VAE	8×8	27.01	25.56
	SQ-VAE	8×8	8.93	4.94
	VQ-WAE	8×8	**8.21**	**3.88**
SVHN	VQ-VAE	8×8	62.13	64.24
	SQ-VAE	8×8	31.26	36.41
	VQ-WAE	8×8	**30.71**	**34.44**
CELEBA	VQ-VAE	16×16	42.0	-
	SQ-VAE	16×16	29.5	-
	VQ-WAE	16×16	**28.8**	-

5. Conclusion

In this paper, we study discrete deep representation learning from the generative perspective. By leveraging with the nice properties of the WS distance, we develop rigorous and rich theories to turn the generative-inspired formulation to an equivalent trainable form relevant to a reconstruction term and the WS distances between latent representations and the codeword distributions. We harvest our theory development to propose Vector Quantized Wasserstein Auto-Encoder (VQ-WAE). We conduct comprehensive experiments to show that our VQ-WAE utilizes the codebooks more efficiently than the baselines, hence leading to better reconstructed and generated image quality. Additionally, the ablation study shows our proposed framework can optimally utilize the codebook, resulting diverse codewords, allowing VQ-WAE to produce better reconstructions of data examples and more reasonable geometry of the latent manifold.

Moreover, the OP in 3 in Theorem 2.3 hints us a question about learning the joint distribution γ over $\mathbb{P}_{c,\pi^m}, m = 1, \ldots, M$, which if learned appropriately can be served as a distribution over the sequences of codewords in a generative model. Certainly, we can employ a learnable auto-regressive model to characterize γ and train it together with the codewords, encoder, and decoder. Currently, we resort a simple solution by minimizing a relevant upper-bound. We leave the problem of learning γ for our future research.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 2017.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Bengio, Y. Estimating or propagating gradients through stochastic neurons. *arXiv preprint arXiv:1305.2982*, 2013.

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*, 2022. URL https://openreview.net/forum?id=Dzpe9C1mpiv.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Dam, N., Hoang, Q., Le, T., Nguyen, T. D., Bui, H., and Phung, D. Three-player wasserstein gan via amortised duality. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2202–2208, 2019.

Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Dieleman, S., van den Oord, A., and Simonyan, K. The challenge of realistic music generation: modelling raw audio at scale. *Advances in Neural Information Processing Systems*, 31, 2018.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470*, 2018.

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.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Hu, M., Zheng, C., Zheng, H., Cham, T.-J., Wang, C., Yang, Z., Tao, D., and Suganthan, P. N. Unified discrete diffusion for simultaneous vision-language generation. In *International Conference on Learning Representations*, 2023.

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

Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. In *International Conference on Machine Learning*, pp. 1771–1779. PMLR, 2017.

Kim, T., Song, G., Lee, S., Kim, S., Seo, Y., Lee, S., Kim, S. H., Lee, H., and Bae, K. L-verse: Bidirectional generation between image and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16526–16536, 2022.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brébisson, A., Bengio, Y., and Courville, A. C. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32, 2019.

Le, T., Nguyen, T., Ho, N., Bui, H., and Phung, D. Lamda: Label matching deep domain adaptation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6043–6054. PMLR, 18–24 Jul 2021.

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.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nguyen, T., Le, T., Dam, N., Tran, Q. H., Nguyen, T., and Phung, D. Tidot: a teacher imitation learning approach for domain adaptation with optimal transport. In *International Joint Conference on Artificial Intelligence 2021*, pp. 2862–2868. Association for the Advancement of Artificial Intelligence (AAAI), 2021a.

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. PMLR, 2021b.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and Freitas, N. Parallel multiscale autoregressive density estimation. In *International conference on machine learning*, pp. 2912–2921. PMLR, 2017.

Roy, A., Vaswani, A., Neelakantan, A., and Parmar, N. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*, 2018.

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

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 Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20987–21012. PMLR, 17–23 Jul 2022.

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

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

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

van der Maaten, L. and Hinton, G. Visualizing data using t-sne, 2008.

Voloshynovskiy, S., Kondah, M., Rezaeifar, S., Taran, O., Holotyak, T., and Rezende, D. J. Information bottleneck through variational glasses. *arXiv preprint arXiv:1912.00830*, 2019.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Williams, W., Ringer, S., Ash, T., MacLeod, D., Dougherty, J., and Hughes, J. Hierarchical quantized autoencoders. *Advances in Neural Information Processing Systems*, 33: 4524–4535, 2020.

Wu, H. and Flierl, M. Variational information bottleneck on vector quantized autoencoders. *arXiv preprint arXiv:1808.01048*, 2018.

Wu, H. and Flierl, M. Vector quantization-based regularization for autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6380–6387, 2020.

Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

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. In *International Conference on Learning Representations*, 2021.

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*, 2021. URL https://openreview.net/forum?id=Oos98K9Lv-k.

Zheng, C., Vuong, L. T., Cai, J., and Phung, D. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35, 2022.

Appendix

This appendix is organized as follows:

- In Section A, we present all proofs for theory developed in the main paper.

- In Section B, we present the detail of practical algorithm for VQ-WAE.

- In Section C, we delve deeper into the motivation behind KL regularization and conduct an analysis of the parameters λ and λ_r.

- In Section D, we present experimental settings and implementation specification of VQ-WAE.

A. Theoretical Development

Lemma A.1. *(Lemma 2.1 in the main paper) Let $C^* = \{c_k^*\}_k$, π^*, γ^*, and f_d^* be the optimal solution of the OP in Eq. (1). Assume $K^M < N$, then $C^* = \{c_k^*\}_k$, π^*, and f_d^* are also the optimal solution of the following OP:*

$$\min_{f_d} \min_{\pi} \min_{\sigma_{1:M} \in \Sigma_\pi} \sum_{n=1}^{N} d_x\left(x_n, f_d\left([c_{\sigma_m(n)}]_{m=1}^{M}\right)\right), \tag{11}$$

where Σ_π is the set of assignment functions $\sigma : \{1, ..., N\} \to \{1, ..., K\}$ such that for every m the cardinalities $\left|\sigma_m^{-1}(k)\right|, k = 1, ..., K$ are proportional to $\pi_k^m, k = 1, ..., K$. Here we denote $\sigma_m^{-1}(k) = \{n \in [N] : \sigma_m(n) = k\}$ with $[N] = \{1, 2, ..., N\}$.

Proof of Lemma A.1

$\gamma \in \Gamma$ is a distribution over C^M with $\gamma([c_{i_1}, \ldots, c_{i_M}])$ satisfying $\sum_{i_1, ..., i_{m-1}, i_m = k, i_{m+1}, ..., i_M} \gamma([c_{i_1}, .., c_{i_M}]) = \pi_k^m$.

$f_d \# \gamma$ is a distribution over $f_d([c_{i_1}, \ldots, c_{i_M}]$ with the mass $\gamma([c_{i_1}, \ldots, c_{i_M}])$ or in other words, we have

$$f_d \# \gamma = \sum_{i_1, ..., i_M} \gamma([c_{i_1}, \ldots, c_{i_M}]) \delta_{f_d([c_{i_1}, ..., c_{i_M}])}.$$

Therefore, we reach the following OP:

$$\min_{C, \pi} \min_{\gamma} \min_{f_d} \mathcal{W}_{d_x}\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}, \sum_{i_1, ..., i_M} \gamma([c_{i_1}, \ldots, c_{i_M}]) \delta_{f_d([c_{i_1}, ..., c_{i_M}])}\right). \tag{12}$$

By using the Monge definition, we have

$$\mathcal{W}_{d_x}\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}, \sum_{i_1, ..., i_M} \gamma([c_{i_1}, \ldots, c_{i_M}]) \delta_{f_d([c_{i_1}, ..., c_{i_M}])}\right) = \min_{T : T \# \mathbb{P}_x = f_d \# \gamma} \mathbb{E}_{x \sim \mathbb{P}_x}\left[d_x\left(x, T(x)\right)\right]$$

$$= \frac{1}{N} \min_{T : T \# \mathbb{P}_x = f_d \# \gamma} \sum_{n=1}^{N} d_x\left(x_n, T(x_n)\right).$$

Since $T \# \mathbb{P}_x = f_d \# \gamma$, $T(x_n) = f_d([c_{i_1}, \ldots, c_{i_M}])$ for some $i_1, ..., i_M$. Additionally, $\left|T^{-1}\left(f_d([c_{i_1}, \ldots, c_{i_M}])\right)\right|, k = 1, ..., K$ are proportional to $\gamma([c_{i_1}, \ldots, c_{i_M}])$. Denote $\sigma_1, ..., \sigma_M : \{1, ..., N\} \to \{1, ..., K\}$ such that $T(x_n) = f_d([c_{\sigma_1(n)}, \ldots, c_{\sigma_M(n)}]), \forall i = 1, ..., N$, we have $\sigma_1, \ldots, \sigma_M \in \Sigma_\pi$. It follows that

$$\mathcal{W}_{d_x}\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}, \sum_{i_1, ..., i_M} \gamma([c_{i_1}, \ldots, c_{i_M}]) \delta_{f_d([c_{i_1}, ..., c_{i_M}])}\right) = \frac{1}{N} \min_{\sigma_{1:M} \in \Sigma_\pi} \sum_{n=1}^{N} d_x\left(x_n, f_d([c_{i_1}, \ldots, c_{i_M}])\right).$$

Finally, the the optimal solution of the OP in Eq. (12) is equivalent to

$$\min_{f_d} \min_{C,\pi} \min_{\sigma_{1:M} \in \Sigma_\pi} \sum_{n=1}^{N} d_x \left(x_n, f_d \left([c_{i_1}, \ldots, c_{i_M}] \right) \right),$$

which directly implies the conclusion because we have

$$|\sigma_m^{-1}(k)| \propto \sum_{i_1, \ldots, i_{m-1}, i_m = k, i_{m+1}, \ldots, i_M} \gamma \left([c_{i_1}, \ldots, c_{i_M}] \right) = \pi_k^m.$$

Theorem A.2. *(Theorem 2.2 in the main paper) We can equivalently turn the optimization problem in (1) to*

$$\min_{C,\pi,f_d} \min_{\gamma \in \Gamma} \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x} \left[d_x \left(f_d \left(\bar{f}_e(x) \right), x \right) \right], \tag{13}$$

*where $\bar{f}_e$ is a **deterministic discrete** encoder mapping data example x directly to a sequence of M codewords in C^M.*

Proof of Theorem A.2

We first prove that the OP of interest in (1) is equivalent to

$$\min_{C,\pi,f_d} \min_{\gamma \in \Gamma} \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right], \tag{14}$$

where $\bar{f}_e$ is a **stochastic discrete** encoder mapping a data example x directly to sequences of M codewords. To this end, we prove that

$$\mathcal{W}_{d_x} \left(f_d \# \gamma, \mathbb{P}_x \right) = \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right], \tag{15}$$

where $\bar{f}_e$ is a **stochastic discrete** encoder mapping data example x directly to the codebooks.

Let $\bar{f}_e$ be a **stochastic discrete** encoder such that $\bar{f}_e \# \mathbb{P}_x = \gamma$ (i.e., $x \sim \mathbb{P}_x$ and $[c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)$ implies $[c_{i_1}, \ldots, c_{i_M}] \sim \gamma$). We consider $\alpha_{d,c}$ as the joint distribution of $(x, [c_{i_1}, \ldots, c_{i_M}])$ with $x \sim \mathbb{P}_x$ and $[c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)$. We also consider $\alpha_{fc,d}$ as the joint distribution including $(x, x') \sim \alpha_{fc,d}$ where $x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)$, and $x' = f_d \left([c_{i_1}, \ldots, c_{i_M}] \right)$. This follows that $\alpha_{fc,d} \in \Gamma \left(f_d \# \gamma, \mathbb{P}_x \right)$ which admits $f_d \# \gamma$ and $\mathbb{P}_x$ as its marginal distribution have:

$$\mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right] = \mathbb{E}_{(x, [c_{i_1}, \ldots, c_{i_M}]) \sim \alpha_{d,c}} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right]$$

$$\overset{(1)}{=} \mathbb{E}_{(x,x') \sim \alpha_{fc,d}} \left[d_x \left(x, x' \right) \right]$$

$$\geq \min_{\alpha_{fc,d} \in \Gamma(f_d \# \gamma, \mathbb{P}_x)} \mathbb{E}_{(x,x') \sim \alpha_{fc,d}} \left[d_x \left(x, x' \right) \right]$$

$$= \mathcal{W}_{d_x} \left(f_d \# \alpha, \mathbb{P}_x \right).$$

Note that we have the equality in (1) due to $(id, f_d) \# \alpha_{d,c} = \alpha_{fc,d}$.

Therefore, we reach

$$\min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right] \geq \mathcal{W}_{d_x} \left(f_d \# \gamma, \mathbb{P}_x \right).$$

Let $\alpha_{fc,d} \in \Gamma \left(f_d \# \gamma, \mathbb{P}_x \right)$. Let $\alpha_{fc,c} \in \Gamma \left(f_d \# \gamma, \gamma \right)$ be a deterministic coupling such that $[c_{i_1}, \ldots, c_{i_M}] \sim \gamma$ and $x = f_d \left([c_{i_1}, \ldots, c_{i_M}] \right)$ imply $([c_{i_1}, \ldots, c_{i_M}], x) \sim \alpha_{c,fc}$. Using the gluing lemma (see Lemma 5.5 in (Santambrogio, 2015)), there exists a joint distribution $\alpha \in \Gamma \left(\gamma, f_d \# \gamma, \mathbb{P}_x \right)$ which admits $\alpha_{fc,d}$ and $\alpha_{fc,c}$ as the corresponding joint distributions. By denoting $\alpha_{d,c} \in \Gamma \left(\mathbb{P}_x, \gamma \right)$ as the marginal distribution of α over $\mathbb{P}_x, \gamma$, we then have

$$\mathbb{E}_{(x,x') \sim \alpha_{fc,d}} \left[d_x \left(x, x' \right) \right] = \mathbb{E}_{([c_{i_1}, \ldots, c_{i_M}], x', x) \sim \alpha} \left[d_x \left(x, x' \right) \right] = \mathbb{E}_{([c_{i_1}, \ldots, c_{i_M}], x) \sim \alpha_{d,c}, x' = f_d([c_{i_1}, \ldots, c_{i_M}])} \left[d_x \left(x, x' \right) \right]$$

$$= \mathbb{E}_{([c_{i_1}, \ldots, c_{i_M}], x) \sim \alpha_{d,c}} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right]$$

$$= \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right]$$

$$\geq \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} \left[d_x \left(f_d \left([c_{i_1}, \ldots, c_{i_M}] \right), x \right) \right],$$

where $\bar{f}_e(x) = \alpha_{d,c}(\cdot \mid x)$.

This follows that

$$\mathcal{W}_{d_x}(f_d \# \gamma, \mathbb{P}_x) = \min_{\alpha_{fc,d} \in \Gamma(f_d \# \gamma, \mathbb{P}_x)} \mathbb{E}_{(x,x') \sim \alpha_{fc,d}} [d_x(x, x')]$$
$$\geq \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} [d_x(f_d([c_{i_1}, \ldots, c_{i_M}]), x)].$$

This completes the proof for the equality in Eq. (15), which means that the OP of interest in (1) is equivalent to

$$\min_{C, \pi, f_d} \min_{\gamma \in \Gamma} \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e(x)} [d_x(f_d([c_{i_1}, \ldots, c_{i_M}]), x)]. \tag{16}$$

We now further prove the above OP is equivalent to

$$\min_{C, \pi, f_d} \min_{\gamma \in \Gamma} \min_{\bar{f}_e : \bar{f}_e \# \mathbb{P}_x = \gamma} \mathbb{E}_{x \sim \mathbb{P}_x} [d_x(f_d(\bar{f}_e(x)), x)], \tag{17}$$

where $\bar{f}_e$ is a **deterministic discrete** encoder mapping data example x directly to the codebooks.

It is obvious that the OP in (17) is special case of that in (16) when we limit to search for deterministic discrete encoders. Given the optimal solution $C^{*1}, \pi^{*1}, \gamma^{*1}, f_d^{*1}$, and $\bar{f}_e^{*1}$ of the OP in (16), we show how to construct the optimal solution for the OP in (17). Let us construct $C^{*2} = C^{*1}, f_d^{*2} = f_d^{*1}$. Given $x \sim \mathbb{P}_x$, let us denote $\bar{f}_e^{*2}(x) = \operatorname{argmin}_{[c_{i_1}, \ldots, c_{i_M}]} d_x(f_d^{*2}([c_{i_1}, \ldots, c_{i_M}]), x)$. Thus, $\bar{f}_e^{*2}$ is a deterministic discrete encoder mapping data example x directly to a sequence of codewords. We define $\pi_k^{*m2} = Pr(\bar{f}_{e,m}^{*2}(x) = c_k : x \sim \mathbb{P}_x), k = 1, \ldots, K$ where $\bar{f}_e^{*2}(x) = [\bar{f}_{e,m}^{*2}(x)]_{m=1}^M$, meaning that $\bar{f}_e^{*2} \# \mathbb{P}_x = \gamma^{*2}$, admitting $\mathbb{P}_{c^{*2}, \pi^{*m2}}, m = 1, \ldots, M$ as its marginal distributions. From the construction of $\bar{f}_e^{*2}$, we have

$$\mathbb{E}_{x \sim \mathbb{P}_x} [d_x(f_d^{*2}(\bar{f}_e^{*2}(x)), x)] \leq \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e^{*1}(x)} [d_x(f_d^{*1}([c_{i_1}, \ldots, c_{i_M}]), x)].$$

Furthermore, because $C^{*2}, \pi^{*2}, f_d^{*2}$, and $\bar{f}_e^{*2}$ are also a feasible solution of the OP in (17), we have

$$\mathbb{E}_{x \sim \mathbb{P}_x} [d_x(f_d^{*2}(\bar{f}_e^{*2}(x)), x)] \geq \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e^{*1}(x)} [d_x(f_d^{*1}([c_{i_1}, \ldots, c_{i_M}]), x)].$$

This means that

$$\mathbb{E}_{x \sim \mathbb{P}_x} [d_x(f_d^{*2}(\bar{f}_e^{*2}(x)), x)] = \mathbb{E}_{x \sim \mathbb{P}_x, [c_{i_1}, \ldots, c_{i_M}] \sim \bar{f}_e^{*1}(x)} [d_x(f_d^{*1}([c_{i_1}, \ldots, c_{i_M}]), x)],$$

and $C^{*2}, \pi^{*2}, \gamma^{*2}, f_d^{*2}$, and $\bar{f}_e^{*2}$ are also the optimal solution of the OP in (17).

We now propose and prove the following lemma that is necessary for the proof of Theorem A.4.

Lemma A.3. *Consider C, π, f_d, and f_e as a feasible solution of the OP in (4). Let us denote $\bar{f}_e^m(x) = \operatorname{argmin}_c \rho_z(f_e^m(x)), c) = Q_C(x)$, then $\bar{f}_e^m(x)$ is a Borel measurable function and hence also $\bar{f}_e(x) = [\bar{f}_e^m(x)]_{m=1}^M$*

Proof of Lemma A.3.

We denote the set A_k on the latent space as

$$A_k = \{z : \rho_z(z, c_k) < \rho_z(z, c_j), \forall j \neq k\} = \{z : Q_C(z) = c_k\}.$$

A_k is known as a Voronoi cell w.r.t. the metric ρ_z. If we consider a continuous metric ρ_z, A_k is a measurable set. Given a Borel measurable function B, we prove that $(\bar{f}_e^m)^{-1}(B)$ is a Borel measurable set on the data space.

Let $B \cap \{c_1, .., c_K\} = \{c_{i_1}, ..., c_{i_t}\}$, we prove that $(\bar{f}_e^m)^{-1}(B) = \cup_{j=1}^t (\bar{f}_e^m)^{-1}(A_{i_j})$. Indeed, take $x \in (\bar{f}_e^m)^{-1}(B)$, then $(\bar{f}_e^m)^{-1}(x) \in B$, implying that $(\bar{f}_e^m)^{-1}(x) = Q_C(x) = c_{i_j}$ for some $j = 1, ..., t$. This means that $f_e^m(x) \in A_{i_j}$ for some $j = 1, ..., t$. Therefore, we reach $(\bar{f}_e^m)^{-1}(B) \subset \cup_{j=1}^t (f_e^m)^{-1}(A_{i_j})$.

We now take $x \in \cup_{j=1}^t (f_e^m)^{-1}(A_{i_j})$. Then $f_e^m(x) \in A_{i_j}$ for $j = 1, ..., t$, hence $\bar{f}_e^m(x) = Q_C(x) = c_{i_j}$ for some $j = 1, ..., t$. Thus, $\bar{f}_e^m(x) \subset B$ or equivalently $x \in (\bar{f}_e^m)^{-1}(B)$, implying $(\bar{f}_e^m)^{-1}(B) \supset \cup_{j=1}^t (f_e^m)^{-1}(A_{i_j})$.

Finally, we reach $(\bar{f}_e^m)^{-1}(B) = \cup_{j=1}^t (f_e^m)^{-1}(A_{i_j})$, which concludes our proof because f_e^m is a measurable function and A_{i_j} are measurable sets.

Theorem A.4. *(**Theorem 2.3 in the main paper**) If we seek f_d and f_e in a family with infinite capacity (e.g., the family of all measurable functions), the the two OPs of interest in (1) and (3) are equivalent to the following OP*

$$\min_{C,\pi} \min_{\gamma\in\Gamma} \min_{f_d,f_e} \left\{ \begin{array}{c} \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(Q_C\left(f_e\left(x\right)\right)\right),x\right)\right] \\ +\lambda\mathcal{W}_{d_z}\left(f_e\#\mathbb{P}_x,\gamma\right), \end{array} \right\} \tag{18}$$

where $Q_C\left(f_e\left(x\right)\right) = [Q_C(f_e^m\left(x\right))]_{m=1}^M$ with $Q_C(f_e^m\left(x\right)) = argmin_{c\in C}\rho_z\left(f_e^m\left(x\right),c\right)$ is a quantization operator *which returns the sequence of closest codewords to $f_e^m\left(x\right), m = 1,\ldots,M$ and the parameter $\lambda > 0$. Here we overload the quantization operator for both $f_e(x) \in \mathcal{Z}^M$ and $f_e^m(x) \in \mathcal{Z}$. Additionally, given $z = [z^m]_{m=1}^M \in \mathcal{Z}^M, \bar{z} = [\bar{z}^m]_{m=1}^M \in \mathcal{Z}^M$, the distance between them is defined as $d_z\left(z,\bar{z}\right) = \frac{1}{M}\sum_{m=1}^M \rho_z\left(z^m,\bar{z}^m\right)$ where ρ_z is a distance on $\mathcal{Z}$.*

Proof of Theorem A.4.

Given the optimal solution $C^{*1}, \pi^{*1}, f_d^{*1}, \gamma^{*1}$, and f_e^{*1} of the OP in (4), we conduct the optimal solution for the OP in (3). Let us conduct $C^{*2} = C^{*1}, f_d^{*2} = f_d^{*1}$. We next define $\bar{f}_e^{*2}\left(x\right) = Q_{C^{*1}}\left(f_e^{*1}\left(x\right)\right) = Q_{C^{*2}}\left(f_e^{*1}\left(x\right)\right)$. We prove that $C^{*2}, \pi^{*2}, f_d^{*2}$, and $\bar{f}_e^{*2}$ are optimal solution of the OP in (3). Define $\gamma^{*2} = Q_{C^{*2}}\#(f_e^{*1}\#\mathbb{P}_x)$. By this definition, we yield $\bar{f}_e^{*2}\#\mathbb{P}_x = \gamma^{*2}$ and hence $\mathcal{W}_{d_z}\left(\bar{f}_e^{*2}\#\mathbb{P}_x,\gamma^{*2}\right) = 0$. Therefore, we need to verify the following:

(i) $\bar{f}_e^{*2}$ is a Borel-measurable function.

(ii) Given a feasible solution C, π, f_d, γ, and $\bar{f}_e$ of (3), we have

$$\mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*2}\left(\bar{f}_e^{*2}\left(x\right)\right),x\right)\right] \le \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(\bar{f}_e\left(x\right)\right),x\right)\right]. \tag{19}$$

We first prove (i). It is a direct conclusion because the application of Lemma A.3 to $C^{*1}, \pi^{*1}, f_d^{*1}$, and f_e^{*1}.

We next prove (ii). We further derive as

$$\begin{aligned} &\mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*2}\left(\bar{f}_e^{*2}\left(x\right)\right),x\right)\right] + \lambda\mathcal{W}_{d_z}\left(\bar{f}_e^{*2}\#\mathbb{P}_x,\gamma^{*2}\right) \\ &= \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*2}\left(\bar{f}_e^{*2}\left(x\right)\right),x\right)\right] \\ &= \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*1}\left(Q_{C^{*2}}\left(f_e^{*1}\left(x\right)\right)\right),x\right)\right] \\ &= \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*1}\left(Q_{C^{*1}}\left(f_e^{*1}\left(x\right)\right)\right),x\right)\right] \\ &\le \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*1}\left(Q_{C^{*1}}\left(f_e^{*1}\left(x\right)\right)\right),x\right)\right] + \lambda\mathcal{W}_{d_z}\left(f_e^{*1}\#\mathbb{P}_x,\gamma^{*1}\right). \end{aligned} \tag{20}$$

Moreover, because $\bar{f}_e\#\mathbb{P}_x = \gamma$ which is a discrete distribution over C^M, we obtain $Q_C(\bar{f}_e(x)) = \bar{f}_e(x)$. Note that C, π, f_d, and $\bar{f}_e$ is also a feasible solution of (4) because $\bar{f}_e$ is also a specific encoder mapping from the data space to the latent space, we achieve

$$\begin{aligned} &\mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(Q_C\left(\bar{f}_e\left(x\right)\right)\right),x\right)\right] + \lambda\mathcal{W}_{d_z}\left(\bar{f}_e\#\mathbb{P}_x,\gamma\right) \\ &\ge \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*1}\left(Q_{C^{*1}}\left(\bar{f}_e^{*1}\left(x\right)\right)\right),x\right)\right] + \lambda\mathcal{W}_{d_z}\left(\bar{f}_e^{*1}\#\mathbb{P}_x,\gamma^{*1}\right). \end{aligned}$$

Noting that $\bar{f}_e\#\mathbb{P}_x = \gamma$ and $Q_C(\bar{f}_e(x)) = \bar{f}_e(x)$, we arrive at

$$\begin{aligned} &\mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(\bar{f}_e\left(x\right)\right),x\right)\right] \\ &\ge \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*1}\left(Q_{C^{*1}}\left(\bar{f}_e^{*1}\left(x\right)\right)\right),x\right)\right] + \lambda\mathcal{W}_{d_z}\left(\bar{f}_e^{*1}\#\mathbb{P}_x,\gamma^{*1}\right). \end{aligned} \tag{21}$$

Combining the inequalities in (20) and (21), we obtain Inequality (19) as

$$\mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d^{*2}\left(\bar{f}_e^{*2}\left(x\right)\right),x\right)\right] \le \mathbb{E}_{x\sim\mathbb{P}_x}\left[d_x\left(f_d\left(\bar{f}_e\left(x\right)\right),x\right)\right]. \tag{22}$$

This concludes our proof.

Lemma A.5. *The WS of interest* $\min_\pi \min_{\gamma \in \Gamma} \mathcal{W}_{d_z}\left(f_e \# \mathbb{P}_x, \gamma\right)$ *is upper-bounded by*

$$\frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right). \tag{23}$$

Proof of Lemma A.5

Let $\alpha^{*m} \in \Gamma\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right)$ be the optimal coupling for the WS distance $\mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right)$. We construct a coupling $\alpha \in \Gamma(f_e \# \mathbb{P}_x, \gamma)$ as follows. We first sample $X \sim \mathbb{P}_x$. We then simultaneously sample $C_m \sim \alpha^{*m}(\cdot \mid f_e^m(X)), m = 1, \ldots, M$. Let γ^* be the law of $[C_1, \ldots, C_M]$ and α^* be the law of $(f_e(X), [C_1, \ldots, C_M])$. Let define π^{*m} such that $\mathbb{P}_{c, \pi^{*m}}$ is the marginal distribution of γ^* over C_m. We then have $\gamma^* \in \Gamma(\mathbb{P}_{c, \pi^1}, \ldots, \mathbb{P}_{c, \pi^M})$ and $\alpha^* \in \Gamma(f_e \# \mathbb{P}_x, \gamma^*)$. It follows that

$$\begin{aligned}
\mathcal{W}_{d_z}\left(f_e \# \mathbb{P}_x, \gamma^*\right) &= \mathbb{E}_{(Z, [C_1, \ldots, C_M]) \sim \alpha^*}\left[d_z\left(Z, [C_1, \ldots, C_M]\right)\right] \\
&= \mathbb{E}_{(f_e(X), [C_1, \ldots, C_M]) \sim \alpha^*}\left[d_z\left([f_e^1(X), \ldots, f_e^M(X)], [C_1, \ldots, C_M]\right)\right] \\
&= \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{(f_e^m(X), C_m) \sim \alpha^{*m}}\left[\rho_z\left(f_e^m(X), C_m\right)\right] \\
&= \frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right).
\end{aligned}$$

$$\min_\pi \min_{\gamma \in \Gamma} \mathcal{W}_{d_z}\left(f_e \# \mathbb{P}_x, \gamma\right) \le \mathcal{W}_{d_z}\left(f_e \# \mathbb{P}_x, \gamma^*\right) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right). \tag{24}$$

Corollary A.6. (Corollary 2.5 in the main paper) *Given* $m \in [M]$, *consider minimizing the term:* $\min_{f_e, C} \mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right)$ *in (4), given* π^m *and assume* $K < N$, *its optimal solution* f_e^{*m} *and* C^* *are also the optimal solution of the OP:*

$$\min_{f_e, C} \min_{\sigma \in \Sigma_\pi} \sum_{n=1}^{N} \rho_z\left(f_e^m(x_n), c_{\sigma(n)}\right), \tag{25}$$

where Σ_π *is the set of assignment functions* $\sigma : \{1, \ldots, N\} \to \{1, \ldots, K\}$ *such that the cardinalities* $\left|\sigma^{-1}(k)\right|, k = 1, \ldots, K$ *are proportional to* $\pi_k^m, k = 1, \ldots, K$.

Proof of Corollary A.6.

By the Monge definition, we have

$$\begin{aligned}
\mathcal{W}_{\rho_z}\left(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}\right) &= \mathcal{W}_{\rho_z}\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{f_e^m(x_n)}, \sum_{k=1}^{K} \pi_k^m \delta_{c_k}\right) = \min_{T : T \#(f_e^m \# \mathbb{P}_x) = \mathbb{P}_{c, \pi^m}} \mathbb{E}_{z \sim f_e^m \# \mathbb{P}_x}\left[\rho_z\left(z, T(z)\right)\right] \\
&= \frac{1}{N} \min_{T : T \#(f_e^m \# \mathbb{P}_x) = \mathbb{P}_{c, \pi^m}} \sum_{n=1}^{N} \rho_z\left(f_e^m(x_n), T(f_e^m(x_n))\right).
\end{aligned}$$

Since $T \#\left(f_e^m \# \mathbb{P}_x\right) = \mathbb{P}_{c, \pi^m}$, $T\left(f_e(x_n)\right) = c_k$ for some k. Additionally, $\left|T^{-1}(c_k)\right|, k = 1, \ldots, K$ are proportional to $\pi_k^m, k = 1, \ldots, K$. Denote $\sigma : \{1, \ldots, N\} \to \{1, \ldots, K\}$ such that $T\left(f_e^m(x_n)\right) = c_{\sigma(n)}, \forall i = 1, \ldots, N$, we have $\sigma \in \Sigma_\pi$. It also follows that

$$\mathcal{W}_{\rho_z}\left(\frac{1}{N} \sum_{n=1}^{N} \delta_{f_e^m(x_n)}, \sum_{k=1}^{K} \pi_k^m \delta_{c_k}\right) = \frac{1}{N} \min_{\sigma \in \Sigma_\pi} \sum_{n=1}^{N} \rho_z\left(f_e^m(x_n), c_{\sigma(n)}\right).$$

B. Practical Algorithm for VQ-WAE

We first re-introduce the entropic regularized dual form of optimal transport by (Genevay et al., 2016) which enables the application of optimal transport in machine learning and deep learning:

$$\mathcal{W}_d^{\epsilon}(\mathbb{Q}, \mathbb{P}) := \min_{\gamma \in \Gamma(\mathbb{Q}, \mathbb{P})} \left\{ \mathbb{E}_{(x,y) \sim \gamma}[d(x,y)] + \epsilon D_{KL}(\gamma \parallel \mathbb{Q} \otimes \mathbb{P}) \right\} \tag{26}$$

where ϵ is the regularization rate, $D_{KL}(\cdot \parallel \cdot)$ is the Kullback-Leibler (KL) divergence, an $\mathbb{Q} \otimes \mathbb{P}$ represents the specific coupling in which $\mathbb{Q}$ and $\mathbb{P}$ are independent.

Second, using the Fenchel-Rockafellar theorem, they obtained the following dual form w.r.t. the potential ϕ:

$$\mathcal{W}_d^{\epsilon}(\mathbb{Q}, \mathbb{P}) = \max_{\phi} \left\{ \mathbb{E}_{\mathbb{Q}}[\phi_{\epsilon}^c(x)] + \mathbb{E}_{\mathbb{P}}[\phi(y)] \right\} \tag{27}$$

where $\phi_{\epsilon}^c(x) = -\epsilon \log \left(\mathbb{E}_{\mathbb{P}} \left[\exp \left\{ \frac{-d(x,y) + \phi(y))}{\epsilon} \right\} \right] \right)$.

We now present how to develop a practical method for our VQ-WAE by entropic regularized dual form (27). We rewrite our objective function:

$$\min_{C, \pi, f_d, f_e} \left\{ \begin{array}{c} \mathbb{E}_{x \sim \mathbb{P}_x}[d_x(f_d(Q_C(f_e(x))), x)] \\ + \frac{\lambda}{M} \times \sum_{m=1}^{M} \mathcal{W}_{\rho_z}(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m}) \\ + \lambda_r \sum_{m=1}^{M} D_{KL}(\pi^m, \mathcal{U}_K) \end{array} \right\} \tag{28}$$

where $\lambda, \lambda_r > 0$ are two trade-off parameters and and $\mathcal{U}_K = \left[\frac{1}{K} \right]_K$.

To learn the weights π, we parameterize $\pi^m = \pi^m(\beta^m) = \text{softmax}(\beta^m), m = 1, \ldots, M$ with $\beta^m \in \mathbb{R}^K$. At each iteration, we sample a mini-batch $x_1, \ldots, x_B$ and then solve the above OP by updating f_d, f_e and $C, \beta^{1..M}$ based on this mini-batch as follows. Let us denote

$$\mathbb{P}_B = \frac{1}{B} \sum_{i=1}^{B} \delta_{x_i}$$

as the empirical distribution over the current batch.

For each mini-batch, we replace $\mathcal{W}_{\rho_z}(f_e^m \# \mathbb{P}_x, \mathbb{P}_{c, \pi^m})$ by $\mathcal{W}_{\rho_z}(f_e^m \# \mathbb{P}_B, \mathbb{P}_{c, \pi^m})$ and approximate it with entropic regularized duality form $\mathcal{R}_{WS}^m$ (see Eq. (27)) as follows:

$$\mathcal{R}_{WS}^m = \max_{\phi^m} \left\{ \frac{1}{B} \sum_{i=1}^{B} \left[-\epsilon \log \left(\sum_{k=1}^{K} \pi_k^m \left[\exp \left\{ \frac{-\rho_z(f_e^m(x_i), c_k) + \phi^m(c_k)}{\epsilon} \right\} \right] \right) \right] + \sum_{k=1}^{K} \pi_k^m \phi^m(c_k) \right\} \tag{29}$$

where ϕ^m is a neural net named Kantorovich potential network.

Finally, we update $f_d, f_e, C, \beta^{1:M}$ by solving for each mini-batch:

$$\min_{C, \beta^{1...M}} \min_{f_d, f_e} \max_{\phi^{1...M}} \left\{ \frac{1}{B} \sum_{i=1}^{B} d_x(f_d(Q(f_e(x_i)))) + \sum_{m=1}^{M} \left(\frac{\lambda}{M} \mathcal{R}_{WS}^m + \lambda_r D_{KL}(\pi^m(\beta^m), \mathcal{U}_K) \right) \right\}. \tag{30}$$
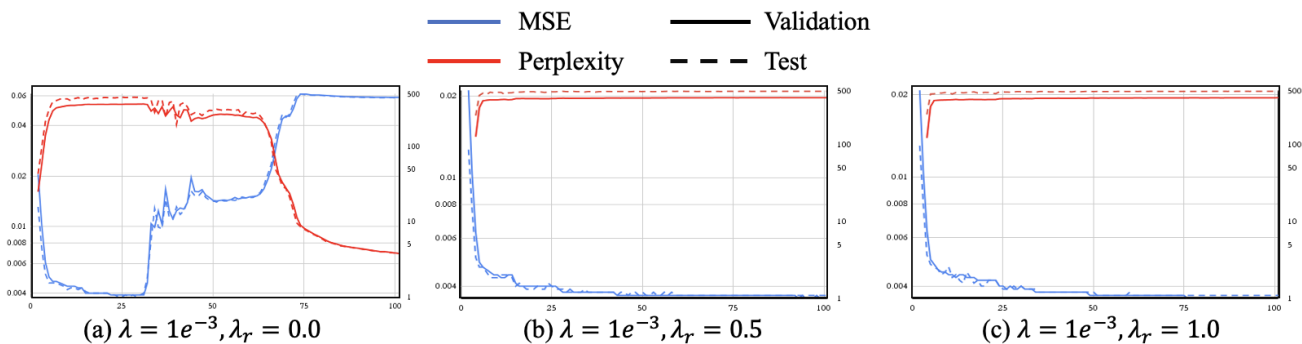
Note that we can optimize M WS distances $\mathcal{W}_{d_z}^{\epsilon}(f_e^m \# \mathbb{P}_N, \mathbb{P}_{c, \pi^m})$ in parallel by matrix computation from current deep learning framework.

C. Analysis of λ and λ_r

In this section, we provide further elaboration on the rationale behind employing regularization on π^m to enforce a uniform distribution, as denoted by the third term in objective 8. The first motivation stems from the desire to ensure the utilization of every discrete codeword. Specifically, we have observed that in the absence of KL regularization (i.e., $\lambda_r = 0.0$), the complexity can be reduced. This reduction occurs because during the optimization of $\{\pi^m\}_{m=1}^{M}$, certain π_k^m values can significantly decrease and converge to zero, resulting in low usage of certain codewords.

Table 6: Reconstruction performance of VQ-WAE with different λ values on CIFAR10 dataset.

Model	λ_r	λ	rFID $\downarrow$	Perplexity $\uparrow$
VQ-WAE	1.0	$1e^{-2}$	55.82	504.5
		$1e^{-3}$	**54.30**	497.3
		$1e^{-4}$	58.96	**507.9**
	0.0	$1e^{-2}$	68.99	445.8
		$1e^{-3}$	57.49	456.5
		$1e^{-4}$	58.17	467.8
VQ-VAE			77.3	69.8
SQ-VAE			55.4	434.8

Figure 5: Training and Validation curve of CIFAR10 with different λ_r.

Secondly, we have observed that training VQ-WAE without KL-regularization leads to divergence after convergence (Figure 5.a). However, the addition of a small KL-regularization term not only enhances model performance but also stabilizes the training process (Figure 5.b and Figure 5.c). Furthermore, the results presented in Table 6 demonstrate that in the absence of KL-regularization ($\lambda_r = 0.0$), performance exhibits significant variability when the value of λ changes. This finding suggests that incorporating the KL-regularization term reduces the model's sensitivity to variations in λ. Additionally, we report the performance of VQ-WAE on CIFAR10 with a fixed π assumed to be a uniform distribution (Table 6). The findings indicate that extremely high perplexity can have a detrimental impact on performance.

D. Experimental Settings

D.1. VQ-model

Implementation: For fair comparison, we utilize the same framework architecture and hyper-parameters for both VQ-VAE and VQ-WAE. Specifically, we construct the VQ-VAE and VQ-WAE models as follows:

- For CIFAR10, MNIST and SVHN 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 deconvolutional layers.

- For CelebA dataset, the models have an encoder with two convolutional layers of stride 2 and filter size of 4×4 with ReLU activation, followed by 6 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 deconvolutional layers.

- For high-quality image dataset FFHQ, we utilize the well-known VQGAN framework (Esser et al., 2021) as the baseline.

Hyper-parameters: Following ([Takida et al., 2022](#)), we adopt the *adam optimizer* for training with: *learning-rate* is e^{-3}, *batch-size* of 32, *embedding dimension* of 64 and *codebook size* $|C| = 512$ for all datasets except FFHQ with *embedding dimension* of 256 and $|C| = 1024$. Finally, we train model for CIFAR10, MNIST, SVHN, FFHQ in 100 epoches and for CelebA in 70 epoches respectively.

Time Complexity: We report extra computation required by VQ-WAE on CIFAR dataset. Note that we need to trains a kantorovich network to estimate the empirical Wasserstein distance which take extra computation for training. In our experiments, the kantorovich network is designed with a hidden layer of $M \times 64$ nodes where M is the number of components of a latent while 64 is the embedding dimension. The training steps ϕ-*iteration is set to 5* which is chosen for fast computation and sufficient optimization. Precisely on the system of a GPU NVIDIA Tesla V100 with dual CPUs Intel Xeon E5-2698 v4, training VQ-WAE takes about 64 *seconds* for one epoch on CIFAR10 dataset, while training a standard VQ-VAE only takes approximately 40 *seconds* for one epoch. For inference, both methods take the same time.

D.2. Generation model

Implementation: It is worth to noting that we employ the codebooks learned from reported VQ-models to extract codeword indices and we use the same model for generation for both VQ-VAE and WQ-VAE.

- CIFAR10, MNIST and SVHN contain the images of shape $(32, 32, 3)$ and latent of shape $(8, 8, 1)$, we feed PixelCNN over the "pixel" values of the 8×8 1-channel latent space.

- CelebA contains the images of shape $(64, 64, 3)$ and latent of shape $(16, 16, 1)$, we feed PixelCNN over the "pixel" values of the 16×16 1-channel latent space.

Hyper-parameters: we adopt the *adam optimizer* for training with: *learning-rate* is $3e^{-4}$, *batch-size* of 32.