Adversarially Regularized Autoencoders

Jake (Junbo) Zhao [*1,2] **Yoon Kim** [*3] **Kelly Zhang** [1] **Alexander M. Rush** [3] **Yann LeCun** [1,2]

Abstract

Deep latent variable models, trained using variational autoencoders or generative adversarial networks, are now a key technique for representation learning of continuous structures. However, applying similar methods to discrete structures, such as text sequences or discretized images, has proven to be more challenging. In this work, we propose a flexible method for training deep latent variable models of discrete structures. Our approach is based on the recently-proposed Wasserstein autoencoder (WAE) which formalizes the adversarial autoencoder (AAE) as an optimal transport problem. We first extend this framework to model discrete sequences, and then further explore different learned priors targeting a controllable representation. This adversarially regularized autoencoder (ARAE) allows us to generate natural textual outputs as well as perform manipulations in the latent space to induce change in the output space. Finally we show that the latent representation can be trained to perform unaligned textual style transfer, giving improvements both in automatic/human evaluation compared to existing methods.

1. Introduction

Recent work on deep latent variable models, such as variational autoencoders (Kingma & Welling, 2014) and generative adversarial networks (Goodfellow et al., 2014), has shown significant progress in learning smooth representations of complex, high-dimensional continuous data such as images. These latent variable representations facilitate the ability to apply smooth transformations in latent space in order to produce complex modifications of generated outputs, while still remaining on the data manifold.

Unfortunately, learning similar latent variable models of discrete structures, such as text sequences or discretized images, remains a challenging problem. Initial work on VAEs for text has shown that optimization is difficult, as the generative model can easily degenerate into a unconditional language model (Bowman et al., 2016). Recent work on generative adversarial networks (GANs) for text has mostly focused on dealing with the non-differentiable objective either through policy gradient methods (Che et al., 2017; Hjelm et al., 2018; Yu et al., 2017) or with the Gumbel-Softmax distribution (Kusner & Hernandez-Lobato, 2016). However, neither approach can yet produce robust representations directly.

In this work, we extend the adversarial autoencoder (AAE) (Makhzani et al., 2015) to discrete sequences/structures. Similar to the AAE, our model learns an encoder from an input space to an adversarially regularized continuous latent space. However unlike the AAE which utilizes a fixed prior, we instead learn a parameterized prior as a GAN. Like sequence VAEs, the model does not require using policy gradients or continuous relaxations. Like GANs, the model provides flexibility in learning a prior through a parameterized generator.

This adversarially regularized autoencoder (ARAE) can further be formalized under the recently-introduced Wasserstein autoencoder (WAE) framework (Tolstikhin et al., 2018), which also generalizes the adversarial autoencoder. This framework connects regularized autoencoders to an optimal transport objective for an implicit generative model. We extend this class of latent variable models to the case of discrete output, specifically showing that the autoencoder cross-entropy loss upper-bounds the total variational distance between the model/data distributions. Under this setup, commonly-used discrete decoders such as RNNs, can be incorporated into the model. Finally to handle non-trivial sequence examples, we consider several different (fixed and learned) prior distributions. These include a standard Gaussian prior used in image models and in the AAE/WAE models, a learned parametric generator acting as a GAN in latent variable space, and a transfer-based parametric generator that is trained to ignore targeted attributes of the input. The last prior can be directly used for unaligned transfer tasks such as sentiment or style transfer.

Experiments apply ARAE to discretized images and text

[*]Equal contribution [1]Department of Computer Science, New York University [2]Facebook AI Research [3]School of Engineering and Applied Sciences, Harvard University. Correspondence to: Jake Zhao <jakezhao@cs.nyu.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

sequences. The latent variable model is able to generate varied samples that can be quantitatively shown to cover the input spaces and to generate consistent image and sentence manipulations by moving around in the latent space via interpolation and offset vector arithmetic. When the ARAE model is trained with task-specific adversarial regularization, the model improves upon strong results on sentiment transfer reported in Shen et al. (2017) and produces compelling outputs on a topic transfer task using only a single shared space. Code is available at https://github.com/jakezhaojb/ARAE.

2. Background and Notation

Discrete Autoencoder Define $\mathcal{X} = \mathcal{V}^n$ to be a set of discrete sequences where $\mathcal{V}$ is a vocabulary of symbols. Our discrete autoencoder will consist of two parameterized functions: a deterministic encoder function $\text{enc}_\phi : \mathcal{X} \mapsto \mathcal{Z}$ with parameters ϕ that maps from input space to code space, and a conditional decoder $p_\psi(\mathbf{x} \mid \mathbf{z})$ over structures $\mathcal{X}$ with parameters ψ. The parameters are trained based on the cross-entropy reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\phi, \psi) = -\log p_\psi(\mathbf{x} \mid \text{enc}_\phi(\mathbf{x}))$$

The choice of the encoder and decoder parameterization is problem-specific, for example we use RNNs for sequences. We use the notation, $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p_\psi(\mathbf{x} \mid \text{enc}_\phi(\mathbf{x}))$ for the decoder mode, and call the model distribution $\mathbb{P}_\psi$.

Generative Adversarial Networks GANs are a class of parameterized implicit generative models (Goodfellow et al., 2014). The method approximates drawing samples from a true distribution $\mathbf{z} \sim \mathbb{P}_*$ by instead employing a noise sample $\mathbf{s}$ and a parameterized generator function $\tilde{\mathbf{z}} = g_\theta(\mathbf{s})$ to produce $\tilde{\mathbf{z}} \sim \mathbb{P}_\mathbf{z}$. Initial work on GANs implicitly minimized the Jensen-Shannon divergence between the distributions. Recent work on Wasserstein GAN (WGAN) (Arjovsky et al., 2017), replaces this with the *Earth-Mover* (Wasserstein-1) distance.

GAN training utilizes two separate models: a *generator* $g_\theta(\mathbf{s})$ maps a latent vector from some easy-to-sample noise distribution to a sample from a more complex distribution, and a critic/discriminator $f_w(\mathbf{z})$ aims to distinguish *real* data and *generated* samples from g_θ. Informally, the generator is trained to fool the critic, and the critic to tell real from generated. WGAN training uses the following min-max optimization over generator θ and critic w,

$$\min_\theta \max_{w \in \mathcal{W}} \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_*}[f_w(\mathbf{z})] - \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_\mathbf{z}}[f_w(\tilde{\mathbf{z}})],$$

where $f_w : \mathcal{Z} \mapsto \mathbb{R}$ denotes the critic function, $\tilde{\mathbf{z}}$ is obtained from the generator, $\tilde{\mathbf{z}} = g_\theta(\mathbf{s})$, and $\mathbb{P}_*$ and $\mathbb{P}_\mathbf{z}$ are real and generated distributions. If the critic parameters w are restricted to an 1-Lipschitz function set $\mathcal{W}$, this term correspond to minimizing Wasserstein-1 distance $W(\mathbb{P}_*, \mathbb{P}_\mathbf{z})$. We use a naive approximation to enforce this property by weight-clipping, i.e. $w = [-\epsilon, \epsilon]^d$ (Arjovsky et al., 2017).[1]

3. Adversarially Regularized Autoencoder

ARAE combines a discrete autoencoder with a GAN-regularized latent representation. The full model is shown in Figure 1, which produces a learned distribution over the discrete space $\mathbb{P}_\psi$. Intuitively, this method aims to provide smoother hidden encoding for discrete sequences with a flexible prior. In the next section we show how this simple network can be formally interpreted as a latent variable model under the Wasserstein autoencoder framework.

The model consists of a discrete autoencoder regularized with a prior distribution,

$$\min_{\phi,\psi} \quad \mathcal{L}_{\text{rec}}(\phi, \psi) + \lambda^{(1)} W(\mathbb{P}_Q, \mathbb{P}_\mathbf{z})$$

Here W is the Wasserstein distance between $\mathbb{P}_Q$, the distribution from a discrete encoder model (i.e. $\text{enc}_\phi(\mathbf{x})$ where $\mathbf{x} \sim \mathbb{P}_\star$), and $\mathbb{P}_\mathbf{z}$, a prior distribution. As above, the W function is computed with an embedded critic function which is optimized adversarially to the generator and encoder.[2]

The model is trained with coordinate descent across: (1) the encoder and decoder to minimize reconstruction, (2) the critic function to approximate the W term, (3) the encoder adversarially to the critic to minimize W:

$$\begin{aligned}
&1)\ \min_{\phi,\psi} && \mathcal{L}_{\text{rec}}(\phi, \psi) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_\star}\left[-\log p_\psi(\mathbf{x} \mid \text{enc}_\phi(\mathbf{x}))\right] \\
&2)\ \max_{w \in \mathcal{W}} && \mathcal{L}_{\text{cri}}(w) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_\star}\left[f_w(\text{enc}_\phi(\mathbf{x}))\right] - \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_\mathbf{z}}\left[f_w(\tilde{\mathbf{z}})\right] \\
&3)\ \min_{\phi} && \mathcal{L}_{\text{enc}}(\phi) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_\star}\left[f_w(\text{enc}_\phi(\mathbf{x}))\right] - \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_\mathbf{z}}\left[f_w(\tilde{\mathbf{z}})\right]
\end{aligned}$$

The full training algorithm is shown in Algorithm 1. Empirically we found that the choice of the prior distribution $\mathbb{P}_\mathbf{z}$ strongly impacted the performance of the model. The simplest choice is to use a fixed distribution such as a Gaussian $\mathcal{N}(0, I)$, which yields a discrete version of the adversarial autoencoder (AAE). However in practice this choice is seemingly too constrained and suffers from mode-collapse.[3]

Instead we exploit the adversarial setup and use learned prior parameterized through a generator model. This is analogous to the use of learned priors in VAEs (Chen et al., 2017; Tomczak & Welling, 2018). Specifically we introduce a generator model, $g_\theta(\mathbf{s})$ over noise $\mathbf{s} \sim \mathcal{N}(0, I)$ to act as an

[1] While we did not experiment with enforcing the Lipschitz constraint via gradient penalty (Gulrajani et al., 2017) or spectral normalization (Miyato et al., 2018), other researchers have found slight improvements by training ARAE with the gradient-penalty version of WGAN (private correspondence).

[2] Other GANs could be used for this optimization. Experimentally we found that WGANs to be more stable than other models.

[3] We note that recent work has successfully utilized AAE for text by instead employing a spherical prior (Cífka et al., 2018).

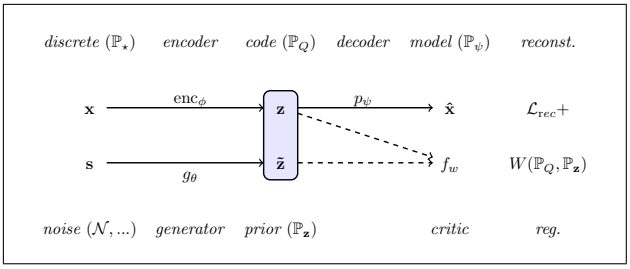

Figure 1: ARAE architecture. A discrete sequence $\mathbf{x}$ is encoded and decoded to produce $\hat{\mathbf{x}}$. A noise sample $\mathbf{s}$ is passed though a generator g_θ (possibly the identity) to produce a prior. The critic function f_w is only used at training to enforce regularization W. The model produce discrete samples $\mathbf{x}$ from noise $\mathbf{s}$. Section 5 relates these samples $\mathbf{x} \sim \mathbb{P}_\psi$ to $\mathbf{x} \sim \mathbb{P}_\star$.

Algorithm 1 ARAE Training

for each training iteration **do**

(1) Train the encoder/decoder for reconstruction (ϕ, ψ)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$

Backprop loss, $\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{i=1}^m \log p_\psi(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i)})$

(2) Train the critic (w)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and $\{\mathbf{s}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$

Compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{z}}^{(i)} = g_\theta(\mathbf{z}^{(i)})$

Backprop loss $-\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{z}^{(i)}) + \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{z}}^{(i)})$

Clip critic w to $[-\epsilon, \epsilon]^d$.

(3) Train the encoder/generator adversarially (ϕ, θ)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and $\{\mathbf{s}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$

Compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{z}}^{(i)} = g_\theta(\mathbf{s}^{(i)})$.

Backprop loss $\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{z}^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{z}}^{(i)})$

end for

implicit prior distribution $\mathbb{P}_\mathbf{z}$.[4] We optimize its parameters θ as part of training in Step 3.

Algorithm 2 ARAE Transfer Extension

Each loop additionally:

(2b) Train attribute classifier (u)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$, lookup $y^{(i)}$, and compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$

Backprop loss $-\frac{1}{m}\sum_{i=1}^m \log p_u(y^{(i)}|\mathbf{z}^{(i)})$

(3b) Train the encoder adversarially (ϕ)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$, lookup $y^{(i)}$, and compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$

Backprop loss $-\frac{1}{m}\sum_{i=1}^m \log p_u(1 - y^{(i)} \mid \mathbf{z}^{(i)})$

Extension: Unaligned Transfer Regularization of the latent space makes it more adaptable for direct continuous optimization that would be difficult over discrete sequences. For example, consider the problem of unaligned transfer, where we want to change an attribute of a discrete input without aligned examples, e.g. to change the topic or sentiment of a sentence. Define this attribute as y and redefine the decoder to be conditional $p_\psi(\mathbf{x} \mid \mathbf{z}, y)$.

To adapt ARAE to this setup, we modify the objective to learn to remove attribute distinctions from the prior (i.e. we want the prior to encode all the relevant information *except* about y). Following similar techniques from other domains, notably in images (Lample et al., 2017) and video modeling (Denton & Birodkar, 2017), we introduce a latent space attribute classifier:

$$\min_{\phi,\psi,\theta} \quad \mathcal{L}_{\text{rec}}(\phi, \psi) + \lambda^{(1)} W(\mathbb{P}_Q, \mathbb{P}_\mathbf{z}) - \lambda^{(2)} \mathcal{L}_{\text{class}}(\phi, u)$$

where $\mathcal{L}_{\text{class}}(\phi, u)$ is the loss of a classifier $p_u(y \mid \mathbf{z})$ from latent variable to labels (in our experiments we always set $\lambda^{(2)} = 1$). This requires two more update steps: (2b) training the classifier, and (3b) adversarially training the encoder to this classifier. This algorithm is shown in Algorithm 2.

4. Theoretical Properties

Standard GANs implicitly minimize a divergence measure (e.g. f-divergence or Wasserstein distance) between the true/model distributions. In our case however, we implicitly minimize the divergence between *learned* code distributions, and it is not clear if this training objective is matching the distributions in the original discrete space. Tolstikhin et al. (2018) recently showed that this style of training is minimizing the Wasserstein distance between the data distribution $\mathbb{P}_\star$ and the model distribution $\mathbb{P}_\psi$ with latent variables (with density $p_\psi(\mathbf{x}) = \int_\mathbf{z} p_\psi(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})\, d\mathbf{z}$).

In this section we apply the above result to the discrete case and show that the ARAE loss minimizes an upper bound on the *total variation distance* between $\mathbb{P}_\star$ and $\mathbb{P}_\psi$.

Definition 1 (Kantorovich's formulation of optimal transport)**.** *Let $\mathbb{P}_\star, \mathbb{P}_\psi$ be distributions over $\mathcal{X}$, and further let $c(\mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ be a cost function. Then the* optimal transport (OT) *problem is given by*

$$W_c(\mathbb{P}_\star, \mathbb{P}_\psi) = \inf_{\Gamma \in \mathcal{P}(\mathbf{x} \sim \mathbb{P}_\star, \mathbf{y} \sim \mathbb{P}_\psi)} \mathbb{E}_{\mathbf{x},\mathbf{y} \sim \Gamma}[c(\mathbf{x}, \mathbf{y})]$$

where $\mathcal{P}(\mathbf{x} \sim \mathbb{P}_\star, \mathbf{y} \sim \mathbb{P}_\psi)$ is the set of all joint distributions of $(\mathbf{x}, \mathbf{y})$ with marginals $\mathbb{P}_\star$ and $\mathbb{P}_\psi$.

In particular, if $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p^p$ then $W_c(\mathbb{P}_\star, \mathbb{P}_\psi)^{\frac{1}{p}}$ is the Wasserstein-p distance between $\mathbb{P}_\star$ and $\mathbb{P}_\psi$. Now suppose we utilize a latent variable model to fit the data, i.e. $\mathbf{z} \sim \mathbb{P}_\mathbf{z}, \mathbf{x} \sim \mathbb{P}_\psi(\mathbf{x} \mid \mathbf{z})$. Then Tolstikhin et al. (2018) prove the following theorem:

Theorem 1. *Let $G_\psi : \mathcal{Z} \to \mathcal{X}$ be a deterministic function (parameterized by ψ) from the latent space $\mathcal{Z}$ to data space $\mathcal{X}$ that induces a dirac distribution $\mathbb{P}_\psi(\mathbf{x} \mid \mathbf{z})$ on $\mathcal{X}$, i.e. $p_\psi(\mathbf{x} \mid \mathbf{z}) = \mathbb{1}\{\mathbf{x} = G_\psi(\mathbf{z})\}$. Let $Q(\mathbf{z} \mid \mathbf{x})$ be*

[4] The downside of this approach is that the latent variable $\mathbf{z}$ is now much less constrained. However we find experimentally that using a a simple MLP for g_θ significantly regularizes the encoder RNN.

any conditional distribution on $\mathcal{Z}$ with density $p_Q(\mathbf{z} \mid \mathbf{x})$. Define its marginal to be $\mathbb{P}_Q$, which has density $p_Q(\mathbf{x}) = \int_{\mathbf{x}} p_Q(\mathbf{z} \mid \mathbf{x})\, p_\star(\mathbf{x}) d\mathbf{x}$. Then,

$$W_c(\mathbb{P}_\star, \mathbb{P}_\psi) = \inf_{Q(\mathbf{z} \mid \mathbf{x}):\mathbb{P}_Q=\mathbb{P}_\mathbf{z}} \mathbb{E}_{\mathbb{P}_\star} \mathbb{E}_{Q(\mathbf{z} \mid \mathbf{x})}[c(\mathbf{x}, G_\psi(\mathbf{z}))]$$

Theorem 1 essentially says that learning an autoencoder can be interpreted as learning a generative model with latent variables, as long as we ensure that the marginalized encoded space is the same as the prior. This provides theoretical justification for adversarial autoencoders (Makhzani et al., 2015), and Tolstikhin et al. (2018) used the above to train deep generative models of images by minimizing the Wasserstein-2 distance (i.e. squared loss between real/generated images). We now apply Theorem 1 to discrete autoencoders trained with cross-entropy loss.

Corollary 1 (Discrete case)**.** *Suppose $\mathbf{x} \in \mathcal{X}$ where $\mathcal{X}$ is the set of all one-hot vectors of length n, and let $f_\psi : \mathcal{Z} \to \Delta^{n-1}$ be a deterministic function that goes from the latent space $\mathcal{Z}$ to the $n-1$ dimensional simplex Δ^{n-1}. Further let $G_\psi : \mathcal{Z} \to \mathcal{X}$ be a deterministic function such that $G_\psi(\mathbf{z}) = \arg\max_{\mathbf{w} \in \mathcal{X}} \mathbf{w}^\top f_\psi(\mathbf{z})$, and as above let $\mathbb{P}_\psi(\mathbf{x} \mid \mathbf{z})$ be the dirac distribution derived from G_ψ such that $p_\psi(\mathbf{x} \mid \mathbf{z}) = \mathbb{1}\{\mathbf{x} = G_\psi(\mathbf{z})\}$. Then the following is an upper bound on $\|\mathbb{P}_\psi - \mathbb{P}_\star\|_{\text{TV}}$, the total variation distance between $\mathbb{P}_\star$ and $\mathbb{P}_\psi$:*

$$\inf_{Q(\mathbf{z} \mid \mathbf{x}):\mathbb{P}_Q=\mathbb{P}_\mathbf{z}} \mathbb{E}_{\mathbb{P}_\star} \mathbb{E}_{Q(\mathbf{z} \mid \mathbf{x})} \Big[-\frac{2}{\log 2} \log \mathbf{x}^\top f_\psi(\mathbf{z}) \Big]$$

The proof is in Appendix A. For natural language we have $n = |\mathcal{V}|^m$ and therefore $\mathcal{X}$ is the set of sentences of length m, where m is the maximum sentence length (shorter sentences are padded if necessary). Then the total variational (TV) distance is given by

$$\|\mathbb{P}_\psi - \mathbb{P}_\star\|_{\text{TV}} = \frac{1}{2} \sum_{\mathbf{x} \in \mathcal{V}^m} |p_\psi(\mathbf{x}) - p_\star(\mathbf{x})|$$

This is an interesting alternative to the usual maximum likelihood approach which instead minimizes $\text{KL}(\mathbb{P}_\star, \mathbb{P}_\psi)$.[5] It is also clear that $-\log \mathbf{x}^\top f_\psi(\mathbf{z}) = -\log p_\psi(\mathbf{x} \mid \mathbf{z})$, the standard autoencoder cross-entropy loss at the sentence level with f_ψ as the decoder. As the above objective is hard to minimize directly, we follow Tolstikhin et al. (2018) and consider an easier objective by (i) restricting $Q(\mathbf{z} \mid \mathbf{x})$ to a family of distributions induced by a deterministic encoder parameterized by ϕ, and (ii) using a Langrangian relaxation of the constraint $\mathbb{P}_Q = \mathbb{P}_\mathbf{z}$. In particular, letting $Q(\mathbf{z} \mid \mathbf{x}) = \mathbb{1}\{\mathbf{z} = \text{enc}_\phi(\mathbf{x})\}$ be the dirac distribution induced by a deterministic encoder (with associated marginal $\mathbb{P}_\phi$), the objective is given by

$$\min_{\phi,\psi} \mathbb{E}_{\mathbb{P}_\star}[-\log p_\psi(\mathbf{x} \mid \text{enc}_\phi(\mathbf{z}))] + \lambda W(\mathbb{P}_\phi, \mathbb{P}_\mathbf{z})$$

Note that our minimizing the Wasserstein distance in the *latent* space $W(\mathbb{P}_\phi, \mathbb{P}_\mathbf{z})$ is independent from the Wassertein distance minimization in the *output* space in WAEs. Finally, instead of using a fixed prior (which led to mode-collapse in our experiments) we parameterize $\mathbb{P}_\mathbf{z}$ implicitly by transforming a simple random variable with a generator (i.e. $\mathbf{s} \sim \mathcal{N}(0, I), \mathbf{z} = g_\theta(\mathbf{s})$). This recovers the ARAE objective from the previous section.

We conclude this section by noting that while the theoretical formalization of the AAE as a latent variable model was an important step, in practice there are many approximations made to the actual optimal transport objective. Meaningfully quantifying (and reducing) such approximation gaps remains an avenue for future work.

5. Methods and Architectures

We experiment with ARAE on three setups: (1) a small model using discretized images trained on the binarized version of MNIST, (2) a model for text sequences trained on the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), and (3) a model trained for text transfer trained on the Yelp/Yahoo datasets for unaligned sentiment/topic transfer. For experiments using a learned prior, the generator architecture uses a low dimensional $\mathbf{s}$ with a Gaussian prior $\mathbf{s} \sim \mathcal{N}(0, \mathbf{I})$, and maps it to $\mathbf{z}$ using an MLP g_θ. The critic f_w is also parameterized as an MLP.

The *image* model encodes/decodes binarized images. Here $\mathcal{X} = \{0,1\}^n$ where n is the image size. The encoder used is an MLP mapping from $\{0,1\}^n \mapsto \mathbb{R}^m$, $\text{enc}_\phi(\mathbf{x}) = \text{MLP}(\mathbf{x}; \phi) = \mathbf{z}$. The decoder predicts each pixel in $\mathbf{x}$ with as a parameterized logistic regression, $p_\psi(\mathbf{x} \mid \mathbf{z}) = \prod_{j=1}^n \sigma(\mathbf{h})^{x_j} (1 - \sigma(\mathbf{h}))^{1-x_j}$ where $\mathbf{h} = \text{MLP}(\mathbf{z}; \psi)$.

The *text* model uses a recurrent neural network (RNN) for both the encoder and decoder. Here $\mathcal{X} = \mathcal{V}^n$ where n is the sentence length and $\mathcal{V}$ is the vocabulary of the underlying language. We define $\text{enc}_\phi(\mathbf{x}) = \mathbf{z}$ to be the last hidden state of an encoder RNN. For decoding we feed $\mathbf{z}$ as an additional input to the decoder RNN at each time step, and calculate the distribution over $\mathcal{V}$ at each time step via softmax, $p_\psi(\mathbf{x} \mid \mathbf{z}) = \prod_{j=1}^n \text{softmax}(\mathbf{W} h_j + \mathbf{b})_{x_j}$ where $\mathbf{W}$ and $\mathbf{b}$ are parameters (part of ψ) and h_j is the decoder RNN hidden state. To be consistent with Corollary 1 we need to find the highest-scoring sequence $\hat{\mathbf{x}}$ under this distribution during decoding, which is intractable in general. Instead we approximate this with greedy search. The *text transfer* model uses the same architecture as the text model but extends it with a classifier $p_u(y \mid \mathbf{z})$ which is modeled using an MLP and trained to minimize cross-entropy.

We further compare our approach with a standard autoencoder (AE) and the cross-aligned autoencoder (Shen et al.,

[5] The relationship between KL-divergence and total variation distance is also given by Pinsker's inquality, which states that $2\|\mathbb{P}_\psi - \mathbb{P}_\star\|_{\text{TV}}^2 \leq \text{KL}(\mathbb{P}_\star, \mathbb{P}_\psi)$.

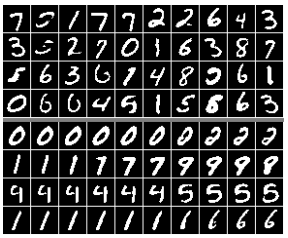

Figure 2: Image samples. The top block shows output generation of the decoder for random noise samples; the bottom block shows sample interpolation results.

Data	Reverse PPL	Forward PPL
Real data	27.4	-
LM samples	90.6	18.8
AE samples	97.3	87.8
ARAE samples	82.2	44.3

Table 1: Reverse PPL: Perplexity of language models trained on the synthetic samples from a ARAE/AE/LM, and evaluated on real data. Forward PPL: Perplexity of a language model trained on real data and evaluated on synthetic samples.

Positive	great indoor mall .
⇒ ARAE	no smoking mall .
⇒ Cross-AE	terrible outdoor urine .
Positive	it has a great atmosphere , with wonderful service .
⇒ ARAE	it has no taste , with a complete jerk .
⇒ Cross-AE	it has a great horrible food and run out service .
Positive	we came on the recommendation of a bell boy and the food was amazing .
⇒ ARAE	we came on the recommendation and the food was a joke .
⇒ Cross-AE	we went on the car of the time and the chicken was awful .
Negative	hell no !
⇒ ARAE	hell great !
⇒ Cross-AE	incredible pork !
Negative	small , smokey , dark and rude management .
⇒ ARAE	small , intimate , and cozy friendly staff .
⇒ Cross-AE	great , , , chips and wine .
Negative	the people who ordered off the menu did n't seem to do much better .
⇒ ARAE	the people who work there are super friendly and the menu is good .
⇒ Cross-AE	the place , one of the office is always worth you do a business .

Table 2: Sentiment transfer results, where we transfer from positive to negative sentiment (Top) and negative to positive sentiment (Bottom). Original sentence and transferred output (from ARAE and the Cross-Aligned AE (from Shen et al. (2017)) of 6 randomly-drawn examples.

2017) for transfer. In both our ARAE and standard AE experiments, the encoder output is normalized to lie on the unit sphere, and the generator output is bounded to lie in $(-1, 1)^n$ by the $\tanh$ function at output layer.

Note, learning deep latent variable models for text sequences has been a significantly more challenging empirical problem than for images. Standard models such as VAEs suffer from optimization issues that have been widely documented. We performed experiments with recurrent VAE, introduced by (Bowman et al., 2016), as well as the adversarial autoencoder (AAE) (Makhzani et al., 2015), both with Gaussian priors. We found that neither model was able to learn meaningful latent representations—the VAE simply ignored the latent code and the AAE experienced mode-collapse and repeatedly generated the same samples.[6] Appendix F includes detailed descriptions of the hyperparameters, model architecture, and training regimes.

6. Experiments

6.1. Distributional Coverage

Section 4 argues that $\mathbb{P}_\psi$ is trained to approximate the true data distribution over discrete sequences $\mathbb{P}_\star$. While it is difficult to test for this property directly (as is the case with most GAN models), we can take samples from model to test the fidelity and coverage of the data space. Figure 2 shows a set of samples from discretized MNIST and Appendix C shows a set of generations from the text ARAE.

A common quantitative measure of sample quality for generative models is to evaluate a strong surrogate model trained on its generated samples. While there are pitfalls of this style of evaluation methods (Theis et al., 2016), it has provided a starting point for image generation models. Here we use a similar method for text generation, which we call *reverse perplexity*. We generate 100k samples from each of the models, train an RNN language model on generated samples and evaluate perplexity on held-out data.[7] While similar metrics for images (e.g. Parzen windows) have been shown to be problematic, we argue that this is less of an issue for text as RNN language models achieve state-of-the-art perplexities on text datasets. We also calculate the usual "forward" perplexity by training an RNN language model on real data and testing on generated data. This measures the fluency of the generated samples, but cannot detect mode-collapse, a common issue in training GANs (Arjovsky & Bottou, 2017; Hu et al., 2018).

Table 1 shows these metrics for (i) ARAE, (ii) an autoencoder (AE),[8] (iii) an RNN language model (LM), and (iv) the real training set. We further find that with a fixed prior, the reverse perplexity of an AAE-style text model (Makhzani et al., 2015) was quite high (980) due to mode-collapse. All models are of the same size to allow for fair comparison. Training directly on real data (understandably) outperforms training on generated data by a large margin. Surprisingly however, training on ARAE samples outperforms training on LM/AE samples in terms of reverse perplexity.

6.2. Unaligned Text Style Transfer

Next we evaluate the model in the context of a learned adversarial prior, as described in Section 3. We experiment with two unaligned text transfer tasks: (i) transfer of sentiment on the Yelp corpus, and (ii) topic on the Yahoo corpus (Zhang

[6] However there have been some recent successes training such models, as noted in the related works section

[7] We also found this metric to be helpful for early-stopping.

[8] To "sample" from an AE we fit a multivariate Gaussian to the code space after training and generate code vectors from this Gaussian to decode back into sentence space.

Model	Automatic Evaluation Transfer	BLEU	Forward	Reverse
Cross-Aligned AE	77.1%	17.75	65.9	124.2
AE	59.3%	37.28	31.9	68.9
ARAE, $\lambda_a^{(1)}$	73.4%	31.15	29.7	70.1
ARAE, $\lambda_b^{(1)}$	81.8%	20.18	27.7	77.0

Model	Human Evaluation Transfer	Similarity	Naturalness
Cross-Aligned AE	57%	3.8	2.7
ARAE, $\lambda_b^{(1)}$	74%	3.7	3.8

Table 3: Sentiment transfer. (Top) Automatic metrics (Transfer/BLEU/Forward PPL/Reverse PPL), (Bottom) Human evaluation metrics (Transfer/Similarity/Naturalness). Cross-Aligned AE is from Shen et al. (2017)

et al., 2015). For sentiment we follow the setup of Shen et al. (2017) and split the Yelp corpus into two sets of unaligned positive and negative reviews. We train ARAE with two separate decoder RNNs, one for positive, $p(\mathbf{x} \mid \mathbf{z}, y = 1)$, and one for negative sentiment $p(\mathbf{x} \mid \mathbf{z}, y = 0)$, and incorporate adversarial training of the encoder to remove sentiment information from the prior. Transfer corresponds to encoding sentences of one class and decoding, greedily, with the opposite decoder. Experiments compare against the cross-aligned AE of Shen et al. (2017) and also an AE trained without the adversarial regularization. For ARAE, we experimented with different $\lambda^{(1)}$ weighting on the adversarial loss (see section 4) with $\lambda_a^{(1)} = 1, \lambda_b^{(1)} = 10$. Both use $\lambda^{(2)} = 1$. Empirically the adversarial regularization enhances transfer and perplexity, but tends to make the transferred text less similar to the original, compared to the AE. Randomly selected example sentences are shown in Table 2 and additional outputs are available in Appendix G.

Table 3 (top) shows quantitative evaluation. We use four automatic metrics: (i) Transfer: how successful the model is at altering sentiment based on an automatic classifier (we use the `fastText` library (Joulin et al., 2017)); (ii) BLEU: the consistency between the transferred text and the original; (iii) Forward PPL: the fluency of the generated text; (iv) Reverse PPL: measuring the extent to which the generations are representative of the underlying data distribution. Both perplexity numbers are obtained by training an RNN language model. Table 3 (bottom) shows human evaluations on the cross-aligned AE and our best ARAE model. We randomly select 1000 sentences (500/500 positive/negative), obtain the corresponding transfers from both models, and ask crowdworkers to evaluate the sentiment (Positive/Neutral/Negative) and naturalness (1-5, 5 being most natural) of the transferred sentences. We create a separate task in which we show the original and the transferred sentences, and ask them to evaluate the similarity based on sentence structure (1-5, 5 being most similar). We explicitly requested that the reader disregard sentiment in similarity assessment.

Science	what is an event horizon with regards to black holes ?
⇒ Music	what is your favorite sitcom with adam sandler ?
⇒ Politics	what is an event with black people ?
Science	take 1ml of hcl (concentrated) and dilute it to 50ml .
⇒ Music	take em to you and shout it to me
⇒ Politics	take bribes to islam and it will be punished .
Science	just multiply the numerator of one fraction by that of the other .
⇒ Music	just multiply the fraction of the other one that 's just like it .
⇒ Politics	just multiply the same fraction of other countries .
Music	do you know a website that you can find people who want to join bands ?
⇒ Science	do you know a website that can help me with science ?
⇒ Politics	do you think that you can find a person who is in prison ?
Music	all three are fabulous artists , with just incredible talent ! !
⇒ Science	all three are genetically bonded with water , but just as many substances , are capable of producing a special case .
⇒ Politics	all three are competing with the government , just as far as i can .
Music	but there are so many more i can 't think of !
⇒ Science	but there are so many more of the number of questions .
⇒ Politics	but there are so many more of the can i think of today .
Politics	republicans : would you vote for a cheney / satan ticket in 2008 ?
⇒ Science	guys : how would you solve this question ?
⇒ Music	guys : would you rather be a good movie ?
Politics	4 years of an idiot in office + electing the idiot again = ?
⇒ Science	4 years of an idiot in the office of science ?
⇒ Music	4) <unk> in an idiot , the idiot is the best of the two points ever !
Politics	anyone who doesnt have a billion dollars for all the publicity cant win .
⇒ Science	anyone who doesnt have a decent chance is the same for all the other .
⇒ Music	anyone who doesnt have a lot of the show for the publicity .

Table 4: Topic Transfer. Random samples from the Yahoo dataset. Note the first row is from ARAE trained on titles while the following ones are from replies.

Model	Medium	Small	Tiny
Supervised Encoder	65.9%	62.5%	57.9%
Semi-Supervised AE	68.5%	64.6%	59.9%
Semi-Supervised ARAE	70.9%	66.8%	62.5%

Table 5: Semi-Supervised accuracy on the natural language inference (SNLI) test set, respectively using 22.2% (medium), 10.8% (small), 5.25% (tiny) of the supervised labels of the full SNLI training set (rest used for unlabeled AE training).

The same method can be applied to other style transfer tasks, for instance the more challenging Yahoo QA data (Zhang et al., 2015). For Yahoo we chose 3 relatively distinct topic classes for transfer: SCIENCE & MATH, ENTERTAINMENT & MUSIC, and POLITICS & GOVERNMENT. As the dataset contains both questions and answers, we separated our experiments into titles (questions) and replies (answers). Randomly-selected generations are shown in Table 4. See Appendix G for additional generation examples.

6.3. Semi-Supervised Training

Latent variable models can also provide an easy method for semi-supervised training. We use a natural language inference task to compare semi-supervised ARAE with other training methods. Results are shown in Table 5. The full SNLI training set contains 543k sentence pairs, and we use supervised sets of 120k (Medium), 59k (Small), and 28k (Tiny) and use the rest of the training set for unlabeled training. As a baseline we use an AE trained on the additional

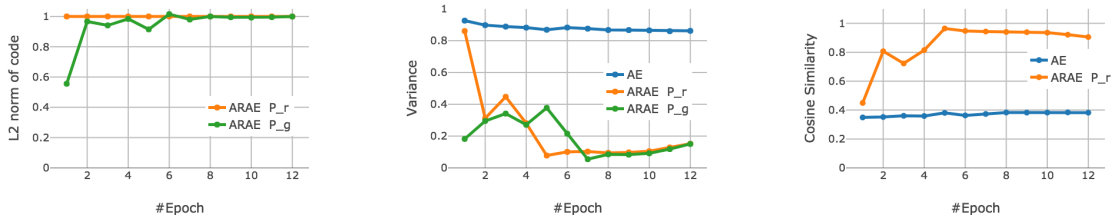

Figure 3: Left: ℓ_2 norm of encoder output $\mathbf{z}$ and generator output $\tilde{\mathbf{z}}$ during ARAE training. ($\mathbf{z}$ is normalized, whereas the generator learns to match). Middle: Sum of the dimension-wise variances of $\mathbf{z}$ and generator codes $\tilde{\mathbf{z}}$ as well as reference AE. Right: Average cosine similarity of nearby sentences (by word edit-distance) for the ARAE and AE during training.

k	AE	ARAE
0	1.06	2.19
1	4.51	4.07
2	6.61	5.39
3	9.14	6.86
4	9.97	7.47

Model	Samples
Original	A woman wearing sunglasses
Noised	A woman sunglasses wearing
AE	A woman sunglasses wearing sunglasses
ARAE	A woman wearing sunglasses
Original	Pets galloping down the street
Noised	Pets down the galloping street
AE	Pets riding the down galloping
ARAE	Pets congregate down the street near a ravine

Figure 4: Reconstruction error (negative log-likelihood averaged over sentences) of the original sentence from a corrupted sentence. Here k is the number of swaps performed on the original sentence.

data, similar to the setting explored in Dai & Le (2015). For ARAE we use the subset of unsupervised data of length < 15 (i.e. ARAE is trained on less data than AE for unsupervised training). The results are shown in Table 5. Training on unlabeled data with an AE objective improves upon a model just trained on labeled data. Training with adversarial regularization provides further gains.

7. Discussion

Impact of Regularization on Discrete Encoding We further examine the impact of adversarial regularization on the encoded representation produced by the model as it is trained. Figure 3 (left), shows a sanity check that the ℓ_2 norm of encoder output $\mathbf{z}$ and prior samples $\tilde{\mathbf{z}}$ converge quickly in ARAE training. The middle plot compares the trace of the covariance matrix between these terms as training progresses. It shows that variance of the encoder and the prior match after several epochs.

Smoothness and Reconstruction We can also assess the "smoothness" of the encoder model learned ARAE (Rifai et al., 2011). We start with a simple proxy that a smooth encoder model should map similar sentences to similar $\mathbf{z}$ values. For 250 sentences, we calculate the average cosine similarity of 100 randomly-selected sentences within an edit-distance of at most 5 to the original. The graph in Figure 3 (right) shows that the cosine similarity of nearby sentences is quite high for ARAE compared to a standard AE and increases in early rounds of training. To further test this property, we feed noised discrete input to the encoder and (i) calculate the score given to the original input, and (ii) compare the resulting reconstructions. Figure 4 (right) shows results for text where k words are first permuted in each sentence. We observe that ARAE is able to map a noised sentence to a natural sentence (though not necessarily the denoised sentence). Figure 4 (left) shows empirical results for these experiments. We obtain the reconstruction error (negative log likelihood) of the original non-noised sentence under the decoder, utilizing the noised code. We find that when $k = 0$ (i.e. no swaps), the regular AE better reconstructs the exact input. However, as the number of swaps pushes the input further away, ARAE is more likely to produce the original sentence. (Note that unlike denoising autoencoders which require a domain-specific noising function (Hill et al., 2016; Vincent et al., 2008), the ARAE is not explicitly trained to denoise an input.)

Manipulation through the Prior An interesting property of latent variable models such as VAEs and GANs is the ability to manipulate output samples through the prior. In particular, for ARAE, the Gaussian form of the noise sample $\mathbf{s}$ induces the ability to smoothly interpolate between outputs by exploiting the structure. While language models may provide a better estimate of the underlying probability space, constructing this style of interpolation would require combinatorial search, which makes this a useful feature of latent variable text models. In Appendix D we show interpolations from for the text model, while Figure 2 (bottom) shows the interpolations for discretized MNIST ARAE.

A related property of GANs is the ability to move in the latent space via offset vectors.[9] To experiment with this property we generate sentences from the ARAE and compute vector transforms in this space to attempt to change main verbs, subjects and modifier (details in Appendix E). Some examples of successful transformations are shown in Figure 5 (bottom). Quantitative evaluation of the success of the vector transformations is given in Figure 5 (top).

[9]Similar to the case with word vectors (Mikolov et al., 2013), Radford et al. (2016) observe that when the mean latent vector for "men with glasses" is subtracted from the mean latent vector for "men without glasses" and applied to an image of a "woman without glasses", the resulting image is that of a "woman with glasses".

Transform	Match %	Prec
walking	85	79.5
man	92	80.2
two	86	74.1
dog	88	77.0
standing	89	79.3
several	70	67.0

A man in a tie is sleeping and clapping on balloons . A man in a tie is clapping and walking dogs .	$\Rightarrow_{\text{walking}}$
The jewish boy is trying to stay out of his skateboard . The jewish man is trying to stay out of his horse .	$\Rightarrow_{\text{man}}$
Some child head a playing plastic with drink . Two children playing a head with plastic drink .	$\Rightarrow_{\text{Two}}$
The people shine or looks into an area . The dog arrives or looks into an area .	$\Rightarrow_{\text{dog}}$
A women are walking outside near a man . Three women are standing near a man walking .	$\Rightarrow_{\text{standing}}$
A side child listening to a piece with steps playing on a table . Several child playing a guitar on side with a table .	$\Rightarrow_{\text{Several}}$

Figure 5: Top: Quantitative evaluation of transformations. Match % refers to the % of samples where at least one decoder samples (per 100) had the desired transformation in the output, while Prec. measures the average precision of the output against the original sentence. Bottom: Examples where the offset vectors produced successful transformations of the original sentence. See Appendix E for the full methodology.

8. Related Work

While ideally autoencoders would learn latent spaces which compactly capture useful features that explain the observed data, in practice they often learn a degenerate *identity* mapping where the latent code space is free of any structure, necessitating the need for some regularization on the latent space. A popular approach is to regularize through an explicit prior on the code space and use a variational approximation to the posterior, leading to a family of models called variational autoencoders (VAE) (Kingma & Welling, 2014; Rezende et al., 2014). Unfortunately VAEs for discrete text sequences can be challenging to train—for example, if the training procedure is not carefully tuned with techniques like word dropout and KL annealing (Bowman et al., 2016), the decoder simply becomes a language model and ignores the latent code. However there have been some recent successes through employing convolutional decoders (Yang et al., 2017; Semeniuta et al., 2017), training the latent representation as a topic model (Dieng et al., 2017; Wang et al., 2018), using the von Mises–Fisher distribution (Guu et al., 2017), and combining VAE with iterative inference (Kim et al., 2018). There has also been some work on making the prior more flexible through explicit parameterization (Chen et al., 2017; Tomczak & Welling, 2018). A notable technique is adversarial autoencoders (AAE) (Makhzani et al., 2015) which attempt to imbue the model with a more flexible prior implicitly through adversarial training. Recent work on Wasserstein autoencoders (Tolstikhin et al., 2018) provides a theoretical foundation for the AAE and shows that AAE minimizes the Wasserstein distance between the data/model distributions.

The success of GANs on images have led many researchers to consider applying GANs to discrete data such as text. Policy gradient methods are a natural way to deal with the resulting non-differentiable generator objective when training directly in discrete space (Glynn, 1987; Williams, 1992). When trained on text data however, such methods often require pre-training/co-training with a maximum likelihood (i.e. language modeling) objective (Che et al., 2017; Yu et al., 2017; Li et al., 2017). Another direction of work has been through reparameterizing the categorical distribution with the Gumbel-Softmax trick (Jang et al., 2017; Maddison et al., 2017)—while initial experiments were encouraging on a synthetic task (Kusner & Hernandez-Lobato, 2016), scaling them to work on natural language is a challenging open problem. There have also been recent related approaches that work directly with the soft outputs from a generator (Gulrajani et al., 2017; Rajeswar et al., 2017; Shen et al., 2017; Press et al., 2017). For example, Shen et al. (2017) exploits adversarial loss for unaligned style transfer between text by having the discriminator act on the RNN hidden states and using the soft outputs at each step as input to an RNN generator. Our approach instead works entirely in fixed-dimensional continuous space and does not require utilizing RNN hidden states directly. It is therefore also different from methods that discriminate in the joint latent/data space, such as ALI (Vincent Dumoulin, 2017) and BiGAN (Donahue et al., 2017). Finally, our work adds to the recent line of work on unaligned style transfer for text (Hu et al., 2017; Mueller et al., 2017; Li et al., 2018; Prabhumoye et al., 2018; Yang et al., 2018).

9. Conclusion

We present adversarially regularized autoencoders (ARAE) as a simple approach for training a discrete structure autoencoder jointly with a code-space generative adversarial network. Utilizing the Wasserstein autoencoder framework (Tolstikhin et al., 2018), we also interpret ARAE as learning a latent variable model that minimizes an upper bound on the total variation distance between the data/model distributions. We find that the model learns an improved autoencoder and exhibits a smooth latent space, as demonstrated by semi-supervised experiments, improvements on text style transfer, and manipulations in the latent space.

We note that (as has been frequently observed when training GANs) the proposed model seemed to be quite sensitive to hyperparameters, and that we only tested our model on simple structures such as binarized digits and short sentences. Cífka et al. (2018) recently evaluated a suite of sentence generation models and found that models are quite sensitive to their training setup, and that different models do well on different metrics. Training deep latent variable models that can robustly model complex discrete structures (e.g. documents) remains an important open issue in the field.

Acknowledgements

We thank Sam Wiseman, Kyunghyun Cho, Sam Bowman, Joan Bruna, Yacine Jernite, Martín Arjovsky, Mikael Henaff, and Michael Mathieu for fruitful discussions. We are particularly grateful to Tianxiao Shen for providing the results for style transfer. We also thank the NVIDIA Corporation for the donation of a Titan X Pascal GPU that was used for this research. Yoon Kim was supported by a gift from Amazon AWS Machine Learning Research.

References

Arjovsky, M. and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. In *Proceedings of ICML*, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. In *Proceedings of ICML*, 2017.

Bowman, S. R., Angeli, G., Potts, C., and Manning., C. D. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, 2015.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating Sentences from a Continuous Space. 2016.

Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. Maximum-Likelihood Augment Discrete Generative Adversarial Networks. *arXiv:1702.07983*, 2017.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. Variational Lossy Autoencoder. In *Proceedings of ICLR*, 2017.

Cífka, O., Severyn, A., Alfonseca, E., and Filippova, K. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv:1804.07972*, 2018.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In *Proceedings of NIPS*, 2015.

Denton, E. and Birodkar, V. Unsupervised learning of disentangled representations from video. In *Proceedings of NIPS*, 2017.

Dieng, A. B., Wang, C., Gao, J., , and Paisley, J. TopicRNN: A Recurrent Neural Network With Long-Range Semantic Dependency. In *Proceedings of ICLR*, 2017.

Donahue, J., Krahenbühl, P., and Darrell, T. Adversarial Feature Learning. In *Proceedings of ICLR*, 2017.

Glynn, P. Likelihood Ratio Gradient Estimation: An Overview. In *Proceedings of Winter Simulation Conference*, 1987.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proceedings of NIPS*, 2014.

Gozlan, N. and Léonard, C. Transport Inequalities. A Survey. *arXiv:1003.3852*, 2010.

Gulrajani, I., Ahmed, F., Arjovsky, M., and Vincent Dumoulin, A. C. Improved Training of Wasserstein GANs. In *Proceedings of NIPS*, 2017.

Guu, K., Hashimoto, T. B., Oren, Y., and Liang, P. Generating Sentences by Editing Prototypes. *arXiv:1709.08878*, 2017.

Hill, F., Cho, K., and Korhonen, A. Learning distributed representations of sentences from unlabelled data. In *Proceedings of NAACL*, 2016.

Hjelm, R. D., Jacob, A. P., Che, T., Cho, K., and Bengio, Y. Boundary-Seeking Generative Adversarial Networks. In *Proceedings of ICLR*, 2018.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Controllable Text Generation. In *Proceedings of ICML*, 2017.

Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. P. On Unifying Deep Generative Models. In *Proceedings of ICLR*, 2018.

Jang, E., Gu, S., and Poole, B. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of ICLR*, 2017.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of Tricks for Efficient Text Classification. In *Proceedings of ACL*, 2017.

Kim, Y., Wiseman, S., Miller, A. C., Sontag, D., and Rush, A. M. Semi-Amortized Variational Autoencoders. In *Proceedings of ICML*, 2018.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *Proceedings of ICLR*, 2014.

Kusner, M. and Hernandez-Lobato, J. M. GANs for Sequences of Discrete Elements with the Gumbel-Softmax Distribution. *arXiv:1611.04051*, 2016.

Lample, G., Zeghidour, N., Usuniera, N., Bordes, A., Denoyer, L., and Ranzato, M. Fader networks: Manipulating images by sliding attributes. In *Proceedings of NIPS*, 2017.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. Adversarial Learning for Neural Dialogue Generation. In *Proceedings of EMNLP*, 2017.

Li, J., Jia, R., He, H., and Liang, P. Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer. In *Proceedings of NAACL*, 2018.

Maddison, C. J., Mnih, A., and Teh, Y. W. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *Proceedings of ICLR*, 2017.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial Autoencoders. *arXiv:1511.05644*, 2015.

Mikolov, T., tau Yih, S. W., and Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of NAACL*, 2013.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization For Generative Adversarial Networks. In *Proceedings of ICLR*, 2018.

Mueller, J., Gifford, D., and Jaakkola, T. Sequence to Better Sequence: Continuous Revision of Combinatorial Structures. In *Proceedings of ICML*, 2017.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. Style Transfer Through Back-Translation. In *Proceedings of ACL*, 2018.

Press, O., Bar, A., Bogin, B., Berant, J., and Wolf, L. Language Generation with Recurrent Generative Adversarial Networks without Pre-training. *arXiv:1706.01399*, 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Proceedings of ICLR*, 2016.

Rajeswar, S., Subramanian, S., Dutil, F., Pal, C., and Courville, A. Adversarial Generation of Natural Language. *arXiv:1705.10929*, 2017.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of ICML*, 2014.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of ICML*, 2011.

Semeniuta, S., Severyn, A., and Barth, E. A Hybrid Convolutional Variational Autoencoder for Text Generation. In *Proceedings of EMNLP*, 2017.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style Transfer from Non-Parallel Text by Cross-Alignment. In *Proceedings of NIPS*, 2017.

Theis, L., van den Oord, A., and Bethge, M. A note on the evaluation of generative models. In *Proceedings of ICLR*, 2016.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein Auto-Encoders. In *Proceedings of ICLR*, 2018.

Tomczak, J. M. and Welling, M. VAE with a VampPrior. In *Proceedings of AISTATS*, 2018.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of ICML*, 2008.

Vincent Dumoulin, Ishmael Belghazi, B. P. O. M. A. L. M. A. A. C. Adversarially Learned Inference. In *Proceedings of ICLR*, 2017.

Wang, W., Gan, Z., Wang, W., Shen, D., Huang, J., Ping, W., Satheesh, S., and Carin, L. Topic Compositional Neural Language Model. In *Proceedings of AISTATS*, 2018.

Williams, R. J. Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8, 1992.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. In *Proceedings of ICML*, 2017.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. Unsupervised Text Style Transfer using Language Models as Discriminators. *arXiv:1805.11749*, 2018.

Yu, L., Zhang, W., Wang, J., and Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of AAAI*, 2017.

Zhang, X., Zhao, J., and LeCun, Y. Character-level Convolutional Networks for Text Classification. In *Proceedings of NIPS*, 2015.