Direct Optimization through $\arg\max$ for Discrete Variational Auto-Encoder

Guy Lorberbom (Technion), Andreea Gane (MIT),
Tommi Jaakkola (MIT), Tamir Hazan (Technion).

Abstract

Reparameterization of variational auto-encoders with continuous latent spaces is an effective method for reducing the variance of their gradient estimates. However, using the same approach when latent variables are discrete is problematic, due to the resulting non-differentiable objective. In this work, we present a direct optimization method that propagates gradients through a non-differentiable $\arg\max$ prediction operation. We apply this method to discrete variational auto-encoders, by modeling a discrete random variable by the $\arg\max$ function of the Gumbel-Max perturbation model.

1 Introduction

Models with discrete latent variables drive extensive research in machine learning applications, such as language classification and generation [34, 7], molecular synthesis [12], or game solving [18]. Compared to their continuous counterparts, discrete latent variable models can decrease the computational complexity of inference calculations, for instance, by discarding alternatives in hard attention models [13], they can improve interpretability by illustrating which terms contributed to the solution [21, 34], and they can facilitate the encoding of inductive biases in the learning process, such as images consisting of a small number of objects [2] or tasks requiring intermediate alignments [18]. Finally, in some cases, discrete latent variables are natural choices, including when modeling datasets with discrete classes [26].

Nonetheless, models involving discrete latent variables are hard to train, with the key issue being to estimate the gradients of the resulting non-differentiable objectives. While one can use unbiased estimators, such as REINFORCE [33], their variance is typically high [22, 19, 30, 4, 20, 31]. In variational auto-encoders (VAEs) with continuous latent variables, the reparameterization trick provides a successful

alternative [10, 25]. however, it cannot be directly applied to non-differentiable objectives.

Recent work [16, 8] uses a relaxation of the discrete VAE objective, where latent variables follow a Gumbel-Softmax distribution, defined by applying the Gumbel-Max trick for sampling from the categorical distribution, and relaxing the resulting $\arg\max$ operation by a softmax operation. The reformulation results in a continuous loss function, which allows the use of the reparameterization trick. Furthermore, Mena et al. [18] provide an extension of this approach for discrete latent structures, namely distributions over latent matchings.

Our work proposes minimizing the non-differentiable objective, by extending the direct loss minimization technique to generative models [17, 28]. Since categorical variables are represented by the Gibbs distribution, we start from the $\arg\max$ formulation of the Gibbs distribution, which we refer to the Gumbel-Max perturbation model, [23, 29, 5, 6, 15]. We subsequently derive an optimization method that directly propagates (biased) gradients through reparameterized $\arg\max$. The direct differentiation of the resulting expectation is estimated by the difference of gradients of two max-perturbations.

We begin by introducing the notation and the problem formulation in Section 3. In Section 4.1, we use the equivalence between the Gibbs distribution and the Gumbel-Max perturbation model to reformulate the discrete VAE objective, with respect to the $\arg\max$ prediction operation. We subsequently state and prove the main result that allows us to directly optimize through the $\arg\max$ function in Section 4.2. In Section 5, we extend this result to mixed discrete-continuous VAEs and to semi-supervised VAE objectives. Finally, we demonstrate the effectiveness of our approach on image generation.

2 Related work

Variational inference has been extensively studied in machine learning, see [1] for a review paper. In our work, we consider variational Bayes bounds with a discrete latent space. Many approaches to optimizing the variational Bayes objective, that are based on samples from the distribution, can be seen as applications of the REINFORCE gradient estimator [33]. These estimators are unbiased, but without a carefully chosen baseline, their variance tends to be too high for the estimator to be useful and considerable work has gone into finding effective baselines [22]. Other methods use various techniques to reduce the estimator variance [24, 19, 4].

Reparameterization is an effective method to reduce the gradient estimate variance in generative learning. Kingma and Welling have shown its effectiveness in auto-encoding variational Bayes (also called variational auto-encoders, VAEs) for

continuous latent spaces [10]. Rezende et al. demonstrated its effectiveness in deep latent models [25]. The success of these works led to reparameterization approaches in discrete latent spaces. Rolfe et al. and Arash et al. represent the marginal distribution of each binary latent variable as a continuous variable in the unit interval. This reparameterization allows the backpropogation of gradients through the continuous representation [26, 32]. These works are restricted to binary random variables, and as a by-product, it encourages high-dimensional representations for which inference is exponential in the dimension size. In contrast, our work reparameterizes the discrete Gibbs latent model, using a Gumbel-Max perturbation model and directly propagates gradients through the reparameterized objective.

Maddison et al. and Jang et al. recently introduced a novel distribution, the Concrete distribution or the Gumbel-Softmax, that continuously relaxes discrete random variables. Replacing every discrete random variable in a model with a Concrete random variable, results in a continuous model, where the reparameterization trick is applicable [16, 8]. These works are close to ours, with a few notable differences. They use the Gumbel-Softmax function and their model is smooth and reparameterization may use the chain rule to propagate gradients. Similar to our setting, the Gumbel-Softmax operation results in a biased estimate of the gradient. Different from our setting, the softmax operation relaxes the variational Bayes objective and results in a non-tight representation. Our work uses the Gumbel-Max perturbation model, which is an equivalent representation of the Gibbs distribution. With that, we do not relax the variational Bayes objective, while our $\arg\max$ prediction remains non-differentiable and we cannot naively use the chain rule to propagate gradients. Instead, we develop a direct optimization method to propagate gradients through $\arg\max$ operation using the difference of gradients of two max-perturbations. Our gradient estimate is biased, except for its limit.

Differentiating through $\arg\max$ prediction was previously done in discriminative learning, in the context of direct loss minimization [17, 28]. Unfortunately, direct loss minimization cannot be applied to generative learning, since it does not have a posterior distribution around its $\arg\max$ prediction. We apply the Gumbel-Max perturbation model to transform the $\arg\max$ prediction to the Gibbs distribution. This also allows us to overcome the "general position" assumption in [17, 28] using our "prediction generating function".

3 Background

To model the data generating distribution, we consider samples $S = \{x_1, ..., x_m\}$ originating from some unknown underlying distribution. We explain the generation process of a parameterized model $p_\theta(x)$, by minimizing its log-loss when marginal-

izing over its hidden variables z. Using variational Bayes, we upper bound the log-loss of an observed data point

$$-\log p_\theta(x) \leq -\mathbb{E}_{z\sim q_\phi} \log p_\theta(x|z) + KL(q_\phi(z|x)||p_\theta(z)) \tag{1}$$

Typically, the model distribution $p_\theta(x|z)$ is a member of the exponential family $p_\theta(x|z) = e^{-\theta(x,z)}$. When considering a discrete latent space, i.e., $z \in \{1, ..., k\}$, the approximated posterior distribution follows the Gibbs distribution law $q_\phi(z|x) \propto e^{\phi(x,z)}$. If the prior distribution is uniform, then $KL(q_\phi(z|x)||p_\theta(z)) = -H(q_\phi) + \log k$ has a closed form. Thus, the challenge in generative learning is to optimize

$$-\mathbb{E}_{z\sim q_\phi} \log p_\theta(x|z) = \sum_{z=1}^{k} \frac{e^{\phi(x,z)}}{\sum_{\hat{z}} e^{\phi(x,\hat{z})}} \theta(x, z) \tag{2}$$

In our work, we directly optimize the variational bound using the equivalence between Gibbs models and Gumbel-Max perturbation models and propagating gradients directly, through the discrete $\arg\max$ prediction operation.

4 Reparameterization and direct optimization

In the following section, we present how to use the Gumbel-Max perturbation to model a discrete random variable by the $\arg\max$ prediction operation. We then derive an optimization method that directly propagates gradients through the reparameterized $\arg\max$ function.

4.1 Gumbel-Max perturbation models

Perturbation models allow an alternative representation of Gibbs distributions $q_\phi(z|x) \propto e^{\phi(x,z)}$ that is based on the extreme value statistics of Gumbel-Max perturbations. Let γ be a random function that associates random variable $\gamma(z)$ for each $z = 1, ..., k$. When the random perturbations follow the zero mean Gumbel distribution law, whose probability density function is $g(\gamma) = e^{-(\gamma+c+e^{-(\gamma+c)})}$ for the Euler constant $c \approx 0.57$, we obtain the following identity between Gibbs models and Gumbel-Max perturbation models[1] (cf. [11])

$$\frac{e^{\phi(x,z)}}{\sum_{\hat{z}} e^{\phi(x,\hat{z})}} = \mathbb{P}_{\gamma\sim g}[z = z^{\phi+\gamma}], \text{ where } z^{\phi+\gamma} \stackrel{def}{=} \arg\max_{\hat{z}=1,...,k} \{\phi(x, \hat{z}) + \gamma(\hat{z})\} \tag{3}$$

[1]The set $\arg\max_{\hat{z}=1,...,k}\{\phi(x, \hat{z}) + \gamma(\hat{z})\}$ is the set of all maximal arguments, and does not always consist of a single element. However, since the Gumbel distribution is continuous, the γ for which there set $\arg\max_{\hat{z}=1,...,k}\{\phi(x, \hat{z}) + \gamma(\hat{z})\}$ consists more than a single element and has a measure of zero. For notational convenience, when we consider integrals (or probability distributions), we ignore measure zero sets.

With this in mind, Equation (2) takes the form

$$-\mathbb{E}_{z\sim q_\phi} \log p_\theta(x|z) = \sum_{z=1}^{k} \mathbb{P}_{\gamma\sim g}[z = z^{\phi+\gamma}]\theta(x,z) = \mathbb{E}_{\gamma\sim g}[\theta(x, z^{\phi+\gamma})] \qquad (4)$$

Since $g(\gamma)$ is a smooth function, we can use a change of variable to emphasize the connection of the Gumbel-Max representation to the traditional stochastic optimization algorithm (REINFORCE). Set $g_\phi(\gamma) = g(\gamma - \phi)$ and derive $\mathbb{E}_{\gamma\sim g}[\theta(x, z^{\phi+\gamma})] = \mathbb{E}_{\gamma\sim g_\phi}[\theta(x, z^{\gamma})]$. If ϕ is a smooth function, we can differentiate under the integral (cf. [3], Theorem 2.27) and using the identity $\nabla \log g_\phi(\gamma) = \frac{\nabla g_\phi(\gamma)}{g_\phi(\gamma)}$, we can estimate the gradient of Equation (4) as the expectation

$$\nabla\mathbb{E}_{\gamma\sim g_\phi}[\theta(x, z^{\gamma})] = \int \nabla g_\phi(\gamma)\theta(x, z^{\gamma})d\gamma = \mathbb{E}_{\gamma\sim g}[\nabla \log(g(\gamma))\nabla\phi(x, z^{\phi+\gamma})\theta(x, z^{\phi+\gamma})] \quad (5)$$

Therefore, using REINFORCE requires estimating expected gradients $\nabla\phi(x, z^{\phi+\gamma})$ that are stretched by $\nabla \log g(\gamma) \cdot \theta(x, z^{\phi+\gamma})$, thus suffering from high variance. Our direct optimization approach estimates the gradient of Equation (4) directly using $\nabla\phi(x, z^{\phi+\gamma})$ without stretching it, therefore, it does not suffer from this high variance phenomena.

4.2 Direct optimization through $\arg\max$

Our main result is presented in Theorem 1 and shows how to compute the gradient of the reparameterized discrete VAE, i.e., $\mathbb{E}_\gamma[\theta(x, z^{\phi+\gamma})]$. In the following section, we omit $\gamma \sim g$ for brevity. We also note that the KL-divergence, which complements the likelihood objective, has an analytic form, since Gumbel-Max perturbation models are Gibbs models (see Equation (3)).
Instrumental to our approach, is a novel "prediction generating function".

$$G(v, \epsilon) = E_\gamma[\max_{\hat{z}}\{\epsilon\theta(x, \hat{z}) + \phi_v(x, \hat{z}) + \gamma(\hat{z})\}] \qquad (6)$$

The variables v are the parameters of the encoder ϕ. In Theorem 1, we prove that $G(v, \epsilon)$ is a smooth function of v, ϵ. Therefore, the Hessian of $G(v, \epsilon$ exists and it is symmetric, namely $\partial_v\partial_\epsilon G(v, 0) = \partial_\epsilon\partial_v G(v, 0)$. In particular, $\partial_v\partial_\epsilon G(v, 0) = \nabla_v E_\gamma[\theta(x, z^{\phi+\gamma})]$ and $\partial_\epsilon\partial_v G(v, 0)$ is the reparameterized gradient computation (see Equation (7) in Theorem 1).

Theorem 1. *Assume $\phi_v(x, z)$ is a smooth function of v. Then the function $G(v, \epsilon)$ in Equation (6) is smooth and*

$$\nabla_v E_\gamma[\theta(x, z^{\phi_v+\gamma})] = \lim_{\epsilon\to 0} \frac{1}{\epsilon}\Big(E_\gamma[\nabla_v\phi_v(x, z^{\epsilon\theta+\phi_v+\gamma}) - \nabla_v\phi_v(x, z^{\phi_v+\gamma})]\Big) \qquad (7)$$

Proof. First, we prove that $G(v, \epsilon)$ is a smooth function. Recall, $g(\gamma)$ is the zero mean Gumbel probability density function. Applying a change of variable $\hat{\gamma}(z) = \epsilon\theta(x, \hat{z}) + \phi_v(x, \hat{z}) + \gamma(\hat{z})$, we obtain

$$G(v, \epsilon) = \int_{-\infty}^{\infty} g(\gamma) \max_{\hat{z}} \{\epsilon\theta(x, \hat{z}) + \phi_v(x, \hat{z}) + \gamma(\hat{z})\} d\gamma = \int_{-\infty}^{\infty} g(\hat{\gamma} - \epsilon\theta - \phi_v) \max_{\hat{z}} \{\hat{\gamma}(\hat{z})\} d\hat{\gamma}.$$

Since $g(\gamma)$ and $\phi(x, z)$ are smooth functions, we proved that $G(v, \epsilon)$ is also smooth. Therefore, the Hessian of $G(v, \epsilon)$ exists and $\partial_v \partial_\epsilon G(v, \epsilon) = \partial_\epsilon \partial_v G(v, \epsilon)$.
To derive our quantities, we differentiate under the integral, both with respect to ϵ and with respect to v. We are able to differentiate under the integral, since $g(\epsilon\theta(x, \hat{z}) + \phi_v(x, \hat{z}) + \gamma(\hat{z}))$ is a smooth function of ϵ and v and its gradient is bounded by an integrable function (cf. [3], Theorem 2.27). Next, we note that the derivative of $\max_{\hat{z}} \{\epsilon\theta(x, \hat{z}) + \phi_v(x, \hat{z}) + \gamma(\hat{z})\}$ is the derivative of $\epsilon\theta(x, z^{\epsilon\theta+\phi+\gamma}) + \phi_v(x, z^{\epsilon\theta+\phi+\gamma}) + \gamma(z^{\epsilon\theta+\phi+\gamma})$ almost everywhere (as the maximal argument is unique almost everywhere). Combining these two observations: $\partial_\epsilon G(v, \epsilon) = E_\gamma[\theta(x, z^{\epsilon\theta+\phi_v+\gamma})]$. This results in the following identities: $\partial_\epsilon G(v, 0) = E_\gamma[\theta(x, z^{\phi_v+\gamma})]$ and $\partial_v \partial_\epsilon G(v, 0) = \nabla_v E_\gamma[\theta(x, z^{\phi_v+\gamma})]$.
To complete the proof, we differentiate under the integral $\partial_v G(v, \epsilon) = E_\gamma[\nabla_v \phi_v(x, z^{\epsilon\theta+\phi_v+\gamma})]$. Therefore, $\partial_\epsilon \partial_v G(v, 0) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}(E_\gamma[\nabla_v \phi_v(x, z^{\epsilon\theta+\phi_v+\gamma}) - \nabla_v \phi_v(x, z^{\phi_v+\gamma})])$
The theorem follows by combining the identity $\partial_v \partial_\epsilon G(v, 0) = \nabla_v E_\gamma[\theta(x, z^{\phi_v+\gamma})]$ with the identity $\partial_\epsilon \partial_v G(v, 0) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}(E_\gamma[\nabla_v \phi_v(x, z^{\epsilon\theta+\phi_v+\gamma}) - \nabla_v \phi_v(x, z^{\phi_v+\gamma})])$ and the Hessian identity $\partial_\epsilon \partial_v G(v, 0) = \partial_v \partial_\epsilon G(v, 0)$. □

Theorem 1 may be compared to standard stochastic approximation techniques that appear in Equation (5). Theorem 1 gradient computation only requires the gradients $\nabla_v \phi_v(x, z^{\phi_v+\gamma})$ and $\nabla_v \phi_v(x, z^{\epsilon\theta+\phi_v+\gamma})$, while Equation (5) multiplies $\nabla_v \phi_v(x, z^{\phi_v+\gamma})$ by $\nabla \log g(\gamma) \cdot \theta(x, z^{\phi_v+\gamma})$. Consequently, the variance of the gradient estimation in Equation (5) is higher.
The gradient estimate in Theorem 1 is unbiased in the limit $\epsilon \to 0$. However, when we approach the limit, the variance of the estimate increases, as the gradient involves the term $\frac{1}{\epsilon}$. In practice, $\epsilon \geq 0.1$ which means that the gradient estimate is biased. This bias-variance tradeoff is demonstrated in Section 6.
The above theorem closely relates to the direct loss minimization technique (cf. [17, 28]), which, in our setting, can be used to compute the gradient of $\mathbb{E}_x \theta(x, z^\phi)$. The direct loss minimization predicts a single z^ϕ for a given x and, therefore, cannot generate a posterior distribution on all $z = 1, ..., k$, i.e., it lacks a generative model that exists in Gumbel-Max perturbation models.
Next, we present an important extension of discrete VAEs to high-dimensional latent spaces [16, 8]. A single discrete random variable $z \in \{1, .., k\}$ cannot represent the variability of the generative process. Therefore, we encode the discrete space with a

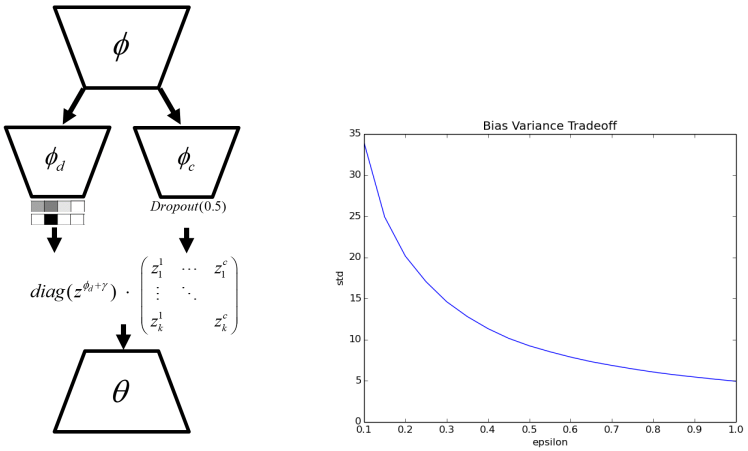

Figure 1: Left: The continuous-discrete latent space is the matrix $diag(z^{\phi_d+\gamma}) \cdot z_c$, where $z^{\phi_d+\gamma}$ is a $k-$length one-hot vector and z_c is a kc matrix of independent Gaussian random variables. If $z^{\phi_d+\gamma} = i$, then the matrix multiplication $diag(z^{\phi_d+\gamma}) \cdot z_c$ is all zero, except for the $i-$th row, which consists of c independent Gaussian random variables. With this we are able to encode both the discrete information (e.g., the image class) and the continuous information (e.g, the image style). Right: demonstrating the variance/bias tradeoff of our gradient estimate. When $\epsilon \to 0$ the gradient estimate is unbiased but it requires many samples to differentiate $\nabla\phi(x, z^{\epsilon\theta+\phi+\gamma}) - \nabla\phi(x, z^{\phi+\gamma})$.

high dimensional variable $z = (z_1, ..., z_n)$, where $z_i \in \{1, ..., k\}$. We note that Theorem 1 holds without any change, if we match the structure of $\phi(x, z)$ and $\gamma(z)$, i.e., we have an independent Gumbel random variable for each $z = (z_1, ..., z_n)$. However, this may be computationally inefficient, since the number of Gumbel random variables in this case is exponential in n. To avoid the exponential complexity, we follow the mean field approximation. We set $\phi(x, z) = \sum_{i=1}^{n} \phi_i(x, z_i)$ and $\gamma(z) = \sum_{i=1}^{n} \gamma_i(z_i)$. Therefore, the perturb-max argument decomposes according to its dimensions, i.e., $z_i^{\phi+\gamma} = \arg\max_{\hat{z}=1,...,k}\{\phi_i(x, \hat{z}) + \gamma_i(\hat{z})\}$ and can be computed efficiently. Unfortunately, $z^{\epsilon\theta+\phi+\gamma}$ cannot be efficiently computed, since $\theta(x, z)$ is not decomposable. Instead, for a given γ, we use a low dimensional approximation $\theta(x, z) = \sum_{i=1}^{n} \tilde{\theta}_i(z_i)$, where $\tilde{\theta}_i(z_i) = \theta(z^{\gamma_1+\phi_1}, ..., z_i, ..., z^{\gamma_n+\phi_n})$. Lastly, the KL-divergence in Equation (1) can be computed efficiently as well: the perturbation model $\mathbb{P}_\gamma[z = z^{\phi+\gamma}]$ is the product of perturbation models $\prod_{i=1}^{n} \mathbb{P}_{\gamma_i}[z_i = z_i^{\phi_i+\gamma_i}]$ and using Equation (3) it reduces to independent Gibbs models.

5 Extensions

5.1 Semi supervised generative models

The main advantage in our framework is that learning a discrete VAE using Gumbel-Max reparameterization is intimately related to predicting the correct discrete latent label. Therefore, semi-supervised VAEs are naturally integrated into our reparameterization framework. As demonstrated in our experiments, discrete VAE's performance may be improved when there exists some form of supervision, to better calibrate the prediction $z^{\phi+\gamma}$. In the semi-supervised setting, the true hidden label is marked for a small part of the training examples [9]. Formally, assume that a subset of the data is labeled, i.e., $S_1 = \{(x_1, z_1), ..., (x_{m_1}, z_{m_1})\}$. In semi-supervised learning, we add to the learning objective the loss function $\ell(z, z^{\phi+\gamma})$, for any $(x, z) \in S_1$, to better control the prediction of latent space. The supervised component of the semi-supervised discrete VAEs is

$$\sum_{x \in S \setminus S_1} \mathbb{E}_{\gamma \sim g_0}[\theta(x, z^{\phi+\gamma})] + \sum_{(x,z) \in S_1} \mathbb{E}_{\gamma}[\theta(x, z^{\phi+\gamma}) + \ell(z, z^{\phi+\gamma})] + \sum_{x \in S} KL(q_\phi(z|x)||p_\theta(z)) \quad (8)$$

The supervised component is explicitly handled by Theorem 1 and optimization of semi-supervised discrete VAEs is straight forward in our framework. Interestingly, our supervised component is intimately related to direct loss minimization [17, 28], which, in our setting, minimizes $\sum_{(x,z) \in S_1} \ell(z, z^{\phi})$. Compared to direct loss minimization, our work adds random perturbation to the encoder and thus overcomes the "general position" assumption of direct loss minimization. This addition allows us to introduce the "prediction generating function", which greatly simplifies our proof.

5.2 Mixing discrete and continuous latent spaces

In many cases, a discrete realization of the latent space is insufficient, since generating data has many nuances. For example, in generating images of digits, the image may be represented both by the discrete digit class (e.g., $0, 1, ...$) and the continuous style (e.g., bold, tilted,...). In the following section, we demonstrate how to apply direct optimization to mixture of discrete and continuous latent spaces. We begin by rephrasing the continuous reparameterization in our framework and then present a simple integration of both spaces.

Continuous VAE relies on reparameterization of Gaussian posterior [10]. The approximated posterior distribution $q_\phi(z|x)$ is a Gaussian centered around $\mu(x)$

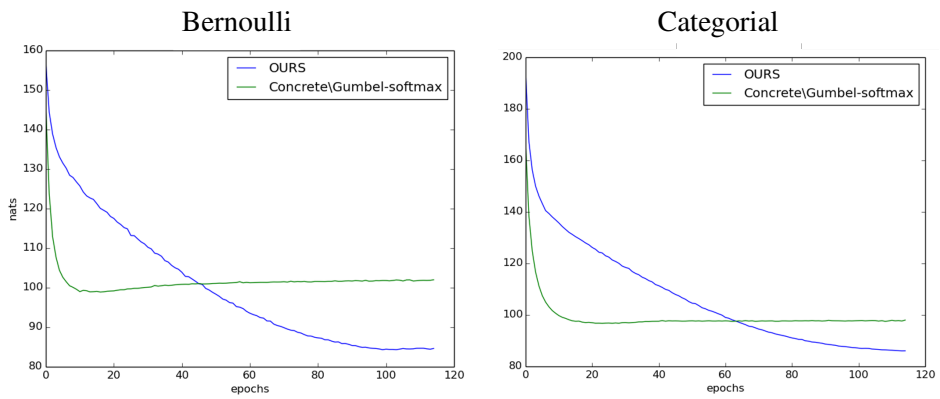

Figure 2: The change in test log-loss as a function of the algorithm epochs. Left: Comparing to Gumbel-Softmax on discrete space of dimension 100×2. Right: Comparing on discrete space of dimension 20×10.

with the probability density function $g_c(z) \propto e^{-\frac{\|z\|^2}{2}}$. Consequently,

$$-\mathbb{E}_{z\sim q_\phi} \log p_\theta(x|z) \quad = \quad \int g_c(z-\mu(x))\theta(x,z)dz = \int g_c(\hat{z})\theta(x,\hat{z}+\mu(x))d\hat{z} \tag{9}$$

The gradient of continuous VAEs with respect to $\mu(x)$ can now be derived by the chain rule over θ.

Now that both discrete and continuous reparameterization are represented by a change of variable, we are able to reparameterize a continuous and discrete mixture model in VAEs. Mixing continuous and discrete latent space, which we denote by $z = (z_c, z_d)$, affects mostly the posterior $q_\phi(z_d, z_c|x) = q_\phi(z_d|x)q_\phi(z_c|z_d, x)$. We set $q_\phi(z_d|x) \propto e^{\phi_d(x,z_d)}$ and $q_\phi(z_c|z_d, x) = g_c(z_c - \mu(x))$. Equivalently, we consider an encoder $\phi(x, z) = \phi_d(x, z_d) + \phi_c(x, z_c)$, with $\phi_c(x, z_c) = \log g_c(z_c - \mu(x))$, where their statistical dependence is due to their shared parameters. In this setting,

$$-\mathbb{E}_{z\sim q_\phi} \log p_\theta(x|z) = \int\int g(\gamma)g_c(\hat{z}_c)\theta(x, \hat{z}_c + \mu(x), z_d^{\phi_d+\gamma})d\hat{z}_c d\gamma \tag{10}$$

An illustration of our mixed discrete-continuous VAE architecture appears in the supplementary material. The gradient of the continuous component of the encoder is computed by the chain rule. The gradient of the discrete component is computed according to Theorem 1.

6 Experiments

We begin our experiments by comparing the test loss of the Gumbel-Softmax optimization to our Gumbel-Max direct optimization. We performed these experiments using the binarized MNIST dataset [27], and the standard 50,000/10,000/10,000 split into training/validation/testing sets. We conducted the experiment, as described by Jang et al. in terms of model architecture, hyperparameters search and annealing schedule for ϵ in our method and τ in Gumbel-Softmax method, [8]. The architecture we consider is 784 - 200 - 784, where the hidden layer is modeled as Bernoulli variables (100×2) or categorial variables (20×10). ϵ/τ is annealed using the schedule $\epsilon = \max(0.5, exp(-rt)$ where r and t are chosen from $\{1e-5, 1e-4\}$ and $\{500, 1000\}$ respectively. Both models were evaluated using importance-sampling with $m = 100$ and trained with $m = 1$. Our best log-likelihood over the test set was 84.16 ± 0.16 for the categorial case and 84.33 ± 0.22 for the binary case. Figure 2 shows that although both approaches use biased estimates for the gradient, their behavior is qualitatively different. The Gumbel-Softmax decreases the test loss faster at first, but our direct optimization reaches a tighter upper bound on the loss. We attribute this qualitative difference to the way these methods consider the training objective. While Gumbel-Softmax relaxes the training objective in Equation (1) and derives an unbiased gradient to the approximated objective, our approach directly tries to optimize the objective in Equation (1), while approximating the gradient. In Figure 1 we also estimate the bias/variance tradeoff for our direct optimization gradients. In this setting, we computed the gradients 1000 times for a fixed ϵ to evaluate the standard deviation for each coordinate of the gradient. We report in the graph the average of these standard deviations.

The main advantage of our framework is that it seamlessly integrates semi-supervised learning. For this experiment, we used the mixed continuous discrete architecture above. Following Jang et al. and Kingma and Welling, we trained on a dataset consisting of $100/300/600$ labeled examples out of the $50,000$ training examples. For labeled examples, we set the perturbed label $z^{\epsilon\theta+\phi+\gamma+\ell}$ to be the true label. This is equivalent to using the indicator function over the space of correct predictions. In Figure 3, we can see that using some supervision improves the image generation class. We also compare the accuracy of our discrete latent space predictor to the state-of-the-art. Our model achieved 96.76% accuracy with 100 labeled samples compared to the Gumbel-Softmax that achieved 93.6%. In addition, our model achieved 97.22% and 97.41% accuracy with 300 and 600 labeled samples respectively. We note that we cannot compare the objective function of both methods, as our objective considers direct loss minimization while Jang et al. objective considers bounds on the log-loss of the latent variables. We attribute our accuracy improvement to the fact we directly minimize the prediction loss, as done in supervised

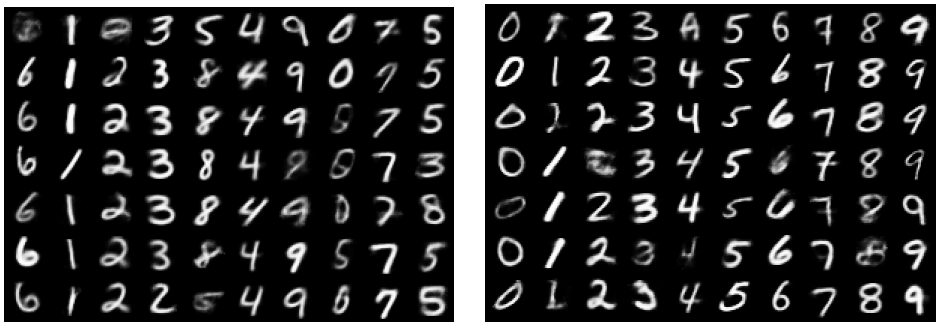

Figure 3: Generated images from an unsupervised (left) and semi-supervised (right) mixture model. The $(i, j) - th$ entry consists of an independent discrete sample and an independent continuous sample from the encoder. One can see that the discrete latent space is able to learn the class representation. The continuous latent space learns the variability within the class. Comparing to unsupervised image generation, we observe that some supervision improves the image generation per class. Right: using direct optimization easily incorporates meaningful loss functions that improve semi-supervised accuracy, compared to Gumbel-Softmax [8].

learning. Figure 3 also shows the posterior representation that mixes continuous and discrete latent variables. Such representation allows us to realize an image both by the discrete digit class (e.g., $0, 1, ...$) and the continuous style (e.g., bold, tilted,...). The generating layer is the matrix multiplication of $diag(z^{\phi_d + \gamma})$ and z_c. The idea is that each digit $i \in \{1, ..., k\}$ can be generated by an independent Gaussian, which appears in the $i-$th row of z_c. The generated images appear in Figure 3. The (i, j)-th entry consists of an independent discrete sample and an independent continuous sample from the encoder. Consequently, the $i-$th column of the image consists of independent samples of a Gaussian random variable z_c, while fixing the discrete random variable z_d. The $j-$th row of the image consists of independent discrete random samples z_d, while fixing the Gaussian random variable z_c.

Semi-supervision also greatly helps to control discrete semantics within images. We learn the attribute representation of the CelebA dataset (cf. [14]) in a semi-supervised manner, while calibrating our prediction with our loss function. For this task, we use convolutional layers for both the encoder and the decoder, except the last two layers of ϕ_c which are linear layers that share parameters over the K possible representations of the image. In Figure 4, we show generated images with discrete semantics turned on/off (with/without glasses, with/without smile, woman/man).

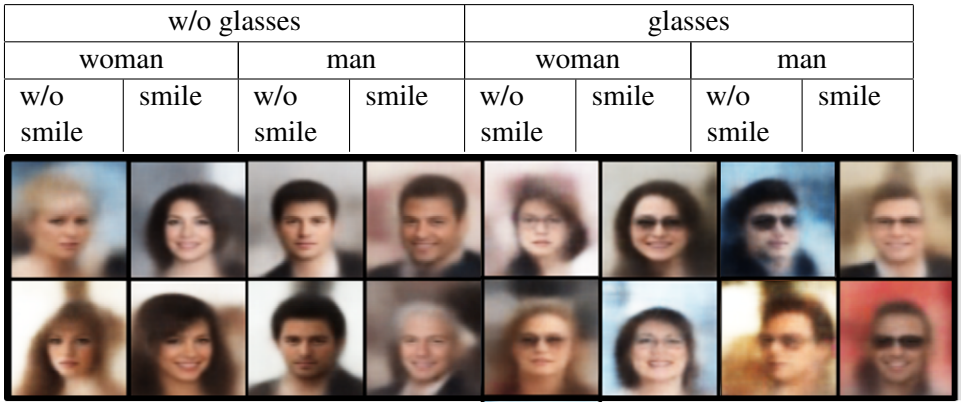

Figure 4: Learning attribute representation in CelebA, using our semi-supervised setting, by calibrating our $\arg\max$ prediction using a loss function. These images here are generated while setting their attributes to get the desired image.

7 Discussion and future work

Dealing with discrete latent variables is a key issue in machine learning. In this work, we use the Gumbel-Max trick to reparameterize discrete VAEs using the $\arg\max$ prediction operator and develop a gradient estimation method that enables direct optimization of the resulting problem by propagating gradients through the non-differentiable $\arg\max$ function. We show that this approach outperforms state-of-the-art methods, and extend it to a mixture of continuous and discrete latent spaces and to semi-supervised learning.

These results can be taken in a number of different directions. Our gradient estimation is practically biased, while REINFORCE is an unbiased estimator. Our methods may benefit from the REBAR framework, which directs our biased gradients towards the unbiased gradient [31]. There are also open problems when fitting this approach to structured latent spaces. Such spaces better fit to generate a complex scene, consisting of several objects and their interactions. However, the direct optimization technique does not immediately apply in this setting, when using a linear number of Gumbel random variables. There are also optimization-related questions that arise from our work: the interplay of ϵ and the learning rate is unexplored and might be correlated. The number of stochastic gradient steps, interleaving Gumbel perturbation with batch samples, might also benefit from a rigorous investigation.

The direct optimization approach we present is general and may be applied beyond VAEs, including reinforcement learning and attention models. Further investigation in this direction is required.

References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[2] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016.

[3] G.B. Folland. Real analysis: Modern techniques and their applications, john wiley & sons. *New York*, 1999.

[4] Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*, 2015.

[5] T. Hazan and T. Jaakkola. On the partition function and random maximum a-posteriori perturbations. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[6] T. Hazan, S. Maji, and T. Jaakkola. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. *NIPS*, 2013.

[7] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596, 2017.

[8] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[9] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] S. Kotz and S. Nadarajah. *Extreme value distributions: theory and applications*. World Scientific Publishing Company, 2000.

[12] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

[13] Dieterich Lawson, George Tucker, Chung-Cheng Chiu, Colin Raffel, Kevin Swersky, and Navdeep Jaitly. Learning hard alignments with variational inference. *arXiv preprint arXiv:1705.05524*, 2017.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[15] Chris Maddison, Danny Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, pages 2085–2093, 2014.

[16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[17] D. McAllester, T. Hazan, and J. Keshet. Direct loss minimization for structured prediction. *Advances in Neural Information Processing Systems*, 23:1594–1602, 2010.

[18] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

[19] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[20] Andriy Mnih and Danilo J Rezende. Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*, 2016.

[21] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

[22] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*, 2012.

[23] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, Barcelona, Spain, November 2011.

[24] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[25] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[26] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[27] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879. ACM, 2008.

[28] Y. Song, A. G. Schwing, R. Zemel, and R. Urtasun. Training Deep Neural Networks via Direct Loss Minimization. In *Proc. ICML*, 2016.

[29] D. Tarlow, R.P. Adams, and R.S. Zemel. Randomized optimum models for structured prediction. In *AISTATS*, pages 21–23, 2012.

[30] Michalis K Titsias. Local expectation gradients for doubly stochastic variational inference. *arXiv preprint arXiv:1503.01494*, 2015.

[31] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2624–2633, 2017.

[32] Arash Vahdat, William G Macready, Zhengbing Bian, and Amir Khoshaman. Dvae++: Discrete variational autoencoders with overlapping transformations. *arXiv preprint arXiv:1802.04920*, 2018.

[33] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[34] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. *arXiv preprint arXiv:1611.09100*, 2016.