RELATIONAL LEARNING WITH VARIATIONAL BAYES

Anonymous authors
Paper under double-blind review

ABSTRACT

In psychology, relational learning refers to the ability to recognize and respond to relationship among objects irrespective of the nature of those objects. Relational learning has long been recognized as a hallmark of human cognition and a key question in artificial intelligence research. In this work, we propose an unsupervised learning method for addressing the relational learning problem where we learn the underlying relationship between a pair of data irrespective of the nature of those data. The central idea of the proposed method is to encapsulate the relational learning problem with a probabilistic graphical model in which we perform inference to learn about data relationships and other relational processing tasks.

1 INTRODUCTION

American Psychological Association defines *relational learning* as (VandenBos & APA, 2007):

Definition 1.1 (**Relational learning**). *Learning to differentiate among stimuli on the basis of relational properties rather than absolute properties.*

In other words, relational learning refers to the ability to recognize and respond to relationship (called *relational property*) among objects irrespective of the nature of those objects (called *absolute property*). Relational learning has long been recognized as a hallmark of human cognition, and there has been substantial research showing that adequate cognitive capacity is necessary for relational processing (Biederman, 1987; Medin et al., 1993; Holyoak, 2012; Doumas & Hummel, 2013; Gentner, 2016). As a machine learning application, relational learning can provide new insight into data analysis by dissecting information in the data into relational property and absolute property. However, in order to discover relationship patterns among raw and unknown data, relational learning is only truly useful if it can be achieved without supervised data. A key challenge in learning relational property with machine learning-based methods is that relational property is an abstract construct; unlike absolute property, which is based on observable data and can be quantitatively measured, relational property is an abstract quantity that is difficult to objectively quantify, especially when the learning is unsupervised.

In this work, we propose an unsupervised learning method—variational relation learning (VRL)—for addressing the relational learning problem. The proposed method is completely unsupervised, which means that the learning does not require a labeled training dataset nor training examples that have the same (or different) relational property. At its core, VRL encapsulates the relational learning problem with a probabilistic graphical model (PGM) in which we perform inference to learn about relational property and other relational processing tasks. Furthermore, our main learning algorithm is derived from the PGM using first principles, which gives us the flexibility to use any compatible computational inference method and still retains the desired properties of the proposed method. Our contribution in this paper is threefold. First, we propose a PGM that encapsulates the relational learning problem. Second, we formulate various relational processing tasks as performing inference and learning in the PGM. Third, we propose an efficient and effective learning algorithm that can be trained end-to-end and unsupervised.

2 PROBLEM DEFINITION

We begin with formulating the relational learning problem as a machine learning problem: we observed a paired dataset $\mathbf{X} = \{ (\mathbf{a}^{(i)}, \mathbf{b}^{(i)}) \mid i \in [1..N] \}$ consisting of N $i.i.d$ samples generated

from a joint distribution $p(\,\mathbf{a}\in\mathcal{A}, \mathbf{b}\in\mathcal{B}\,)$; our goal is to learn a relational property between $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$ irrespective of their absolute property. Furthermore, we want the learning to be unsupervised, e.g., we do not require a labeled dataset, such as $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, z^{(i)})$ where $z^{(i)}$ is a target variable indicating $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$'s relational property, nor do we require training examples that have the same (or different) relational property. There are two distinct features that separate our problem formulation from other unsupervised learning problem formulations:

1. We dissect the information in $\mathbf{X}$ into *relational property* and *absolute property*; relational property characterizes the relationship between $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$, whereas absolute property represents specific features that independently describe $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$.
2. Our goal is to learn a relational property among $\mathbf{X}$ *irrespective* of its absolute property, i.e., we want to learn a relational property that is *decoupled* from the absolute property.

In addition, we are interested in two related relational processing tasks: *relational discrimination*[1] and *relational mapping*[2] (VandenBos & APA, 2007).Relational discrimination allows us to differentiate $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ from $(\mathbf{a}^{(j)}, \mathbf{b}^{(j)})$ based on their relational properties, while relational mapping allows us to apply the relational property of $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ to a different set of data, for example, deduce that $\mathbf{b}^{(j)}$ is related to $\mathbf{a}^{(j)}$ in the same way that $\mathbf{b}^{(i)}$ is related to $\mathbf{a}^{(i)}$.

3 METHOD

Here we introduce the proposed VRL method for addressing the relational learning problem and discuss various optimization challenges unique to VRL.

3.1 VARIATIONAL RELATION LEARNING

The proposed VRL method consists of two parts: first, we encapsulate the relational learning problem with a PGM, called VRL-PGM; we then formulate various relational processing tasks as performing inference and learning in VRL-PGM.

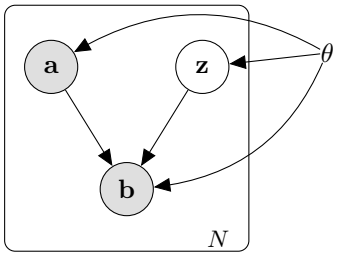


Figure 1: VRL-PGM: a probabilistic graphical model for representing the relational learning problem; the observed random variables $\mathbf{a}$ and $\mathbf{b}$ are generated from some random process (parameterized by θ) involving an unobserved random variable $\mathbf{z}$.

The VRL-PGM model, shown in Fig. 1, generates data $\mathbf{a}$, $\mathbf{z}$, and $\mathbf{b}$ by sampling from PDFs that come from parametric families of distributions— $p_\theta(\,\mathbf{a}\,)$, $p_\theta(\,\mathbf{z}\,)$, $p_\theta(\,\mathbf{b}|\mathbf{a},\mathbf{z}\,)$—that are differentiable almost everywhere with respect to (w.r.t.) $\mathbf{a}$, $\mathbf{z}$, and θ. In practice, we observe only a set of independent realizations $\{\,(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}) \mid i \in [1..N]\,\}$ while the true parameter θ^* and the corresponding latent variables $\mathbf{z}^{(i)}$ are unobserved. A key property of VRL-PGM is that random variables $\mathbf{a}$ and $\mathbf{z}$ are independent but not conditionally independent given $\mathbf{b}$, i.e., $p_\theta(\,\mathbf{a},\mathbf{z}\,) = p_\theta(\,\mathbf{a}\,)p_\theta(\,\mathbf{z}\,)$, $p_\theta(\,\mathbf{a},\mathbf{z}|\mathbf{b}\,) \neq p_\theta(\,\mathbf{a} \mid \mathbf{b}\,)p_\theta(\,\mathbf{z} \mid \mathbf{b}\,)$ (Bishop, 2006). In VRL-PGM, the *absolute property* can be interpreted as representing the dependency between $\mathbf{a}$ and $\mathbf{b}$, i.e., features in $\mathbf{a}$ that can be used to predict $\mathbf{b}$, while

[1]**Definition** (**Relational discrimination** in condition). *A discrimination based on the relationship between or among stimuli rather than on absolute features of the stimuli.*

[2]**Definition** (**Relational mapping**) *The ability to apply what one knows about one set of elements to a different set of elements.*

the *relational property*, represented by the latent variable $\mathbf{z}$, can be interpreted as any additional information not found in $\mathbf{a}$ but can help to better predict $\mathbf{b}$. The key requirement that the learned relational property be decoupled from the absolute property is enforced by VRL-PGM's construction where $\mathbf{z}$, which represents the relational property, and $\mathbf{a}$, from which the absolute property is derived, are independent.

Having established VRL-PGM, our primary learning objective is to approximate the unknown true likelihood function $p_\theta(\,\mathbf{b}\,|\,\mathbf{a},\mathbf{z}\,)$ and posterior $p_\theta(\,\mathbf{z}\,|\,\mathbf{a},\mathbf{b}\,)$ (note that by observing $\mathbf{b}$, random variables $\mathbf{z}$ and $\mathbf{a}$ are no longer independent). Learning $p_\theta(\,\mathbf{z}\mid\mathbf{a},\mathbf{b}\,)$ provides us a way to infer $(\mathbf{a}^{(i)},\mathbf{b}^{(i)})$'s relational property $\mathbf{z}^{(i)}$; moreover, it serves as a basis for performing *relational discrimination* where we compare relational properties between different pairs of data. Learning $p_\theta(\,\mathbf{b}\mid\mathbf{a},\mathbf{z}\,)$ allows us to perform *relational mapping* where we use the relational property of $(\mathbf{a}^{(i)},\mathbf{b}^{(i)})$ to map from $\mathbf{a}^{(j)}$ to $\mathbf{b}^{(j)}$, i.e., $\mathbf{b}^{(j)}\sim p_\theta(\,\mathbf{b}\mid\mathbf{a}^{(j)},\mathbf{z}^{(i)}\,)$ where $\mathbf{z}^{(i)}\sim p_\theta(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$.

3.2 Variational lower bound

We estimate the parameter for $p_\theta(\,\mathbf{b}\,|\,\mathbf{a},\mathbf{z}\,)$ by following the maximum-likelihood (ML) principle, and approximate the true posterior $p_\theta(\,\mathbf{z}\mid\mathbf{a},\mathbf{b}\,)$ with variational Bayesian approach. More specifically, we use a variational distribution $q_\phi(\,\mathbf{z}\mid\mathbf{a},\mathbf{b}\,)$, parameterized by ϕ, to approximate the unknown (and often intractable) true posterior through maximizing a variational lower bound (Bishop, 2006). To derive such a lower bound, we first write the log-evidence as $\log p_\theta(\,\mathbf{X}\,)=\log p_\theta(\,\{\,(\mathbf{a}^{(i)},\mathbf{b}^{(i)})\mid i\in[1..N]\,\}\,)=\sum_{i=1}^{N}\log p_\theta(\,\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$, where each term in the summation can be expressed as:

$$\log p_\theta(\,\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)=D_{\mathrm{KL}}\Big(q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\;\Big\|\;p_\theta(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big)$$
$$+\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{z},\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)-\log q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]\,. \tag{1}$$

The first term on the RHS is the KL-divergence from $p_\theta(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$ to $q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$, which provides a measure of dissimilarity between the two distributions; the second term on the RHS continues as:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{z},\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)-\log q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]\\
&=\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{b}^{(i)}\mid\mathbf{a}^{(i)},\mathbf{z}\,)p_\theta(\,\mathbf{z}\,)p_\theta(\,\mathbf{a}^{(i)}\,)-\log q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]\\
&=\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{b}^{(i)}\mid\mathbf{a}^{(i)},\mathbf{z}\,)+\log p_\theta(\,\mathbf{z}\,)-\log q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]+\log p_\theta(\,\mathbf{a}^{(i)}\,),
\end{aligned}\tag{2}$$

where in the second line we use the fact that random variables $\mathbf{a}$ and $\mathbf{z}$ are independent. Substitute Eq. (2) back in (1) and rearrange terms gives us:

$$\log p_\theta(\,\mathbf{b}^{(i)}\mid\mathbf{a}^{(i)}\,)=D_{\mathrm{KL}}\Big(q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\;\Big\|\;p_\theta(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big)+\mathcal{L}(\theta,\phi;\mathbf{a}^{(i)},\mathbf{b}^{(i)})\tag{3}$$

where

$$\mathcal{L}(\theta,\phi;\mathbf{a}^{(i)},\mathbf{b}^{(i)})=\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{b}^{(i)}|\mathbf{a}^{(i)},\mathbf{z}\,)+\log p_\theta(\,\mathbf{z}\,)-\log q_\phi(\,\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]\,.\tag{4}$$

Since KL-divergence is non-negative, $\mathcal{L}(\theta,\phi;\mathbf{a}^{(i)},\mathbf{b}^{(i)})$ (abbreviated as $\mathcal{L}^{(i)}$ for notation compactness) serves as a lower bound for the conditional log-likelihood $\log p_\theta(\,\mathbf{b}^{(i)}\mid\mathbf{a}^{(i)}\,)$. Maximizing $\mathcal{L}^{(i)}$ w.r.t. ϕ and θ gives us both a ML estimate for $p_\theta(\,\mathbf{b}\mid\mathbf{a},\mathbf{z}\,)$ (by maximizing the first term inside the expectation in Eq. (4)) and a lower KL-divergence (the better $q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$ approximates the true posterior $p_\theta(\,\mathbf{z}\,|\,\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$) as the conditional log-likelihood $\log p_\theta(\,\mathbf{b}^{(i)}\,|\,\mathbf{a}^{(i)}\,)$ does not depend on ϕ. The lower bound $\mathcal{L}^{(i)}$ can be maximized with gradient ascend methods; however, its gradients w.r.t. ϕ is difficult to obtain: the expectation in Eq. (4) is taken w.r.t. the distribution $q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)$, which is a function of ϕ (Paisley et al., 2012). To obtain efficient estimators for both $\mathcal{L}^{(i)}$ and its gradients, we adopt the *reparameterization trick* developed in Kingma & Welling (2014) where the random variable $\mathbf{z}$ is expressed as a transformation of another random variable $\boldsymbol{\epsilon}\sim p(\,\boldsymbol{\epsilon}\,)$ that is independent of $\mathbf{a}$, $\mathbf{b}$, and ϕ: $\mathbf{z}=g(\boldsymbol{\epsilon},\mathbf{a}^{(i)},\mathbf{b}^{(i)},\phi)$ where g is some differentiable and invertible transformation. Given such a change of variable, the lower bound $\mathcal{L}^{(i)}$ can be rewritten as:

$$\mathcal{L}^{(i)}=\mathbb{E}_{p(\boldsymbol{\epsilon})}\Big[\log p_\theta(\,\mathbf{b}^{(i)}\mid\mathbf{a}^{(i)},\mathbf{z}\,)+\log p_\theta(\,\mathbf{z}\,)-\log q_\phi(\,\mathbf{z}\mid\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,)\Big]\,,\tag{5}$$

where $\mathbf{z} = g(\boldsymbol{\epsilon}, \mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \phi)$ and $\boldsymbol{\epsilon} \sim p(\,\boldsymbol{\epsilon}\,)$. Note that the expectation in Eq. (5) is taken w.r.t. $p(\,\boldsymbol{\epsilon}\,)$ and we can now approximate $\mathcal{L}^{(i)}$ with a Monte Carlo estimator:

$$\widetilde{\mathcal{L}}^{(i)} = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,) + \log p_\theta(\,\mathbf{z}^{(i,l)}\,) - \log q_\phi(\,\mathbf{z}^{(i,l)} \mid \mathbf{a}^{(i)}, \mathbf{b}^{(i)}\,), \tag{6}$$

where $\mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \phi)$ and $\boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,)$. The lower bound for a minibatches of data $\mathbf{X}^M = \{\,(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}) \mid i \in [1..M]\,\}$ can be approximated by $\widetilde{\mathcal{L}}(\theta, \phi; \mathbf{X}^M) = \frac{N}{M}\sum_{i=1}^{M} \widetilde{\mathcal{L}}^{(i)}$. And finally, the gradients $\nabla_{\theta,\phi}\widetilde{\mathcal{L}}(\theta, \phi; \mathbf{X}^M) = \frac{N}{M}\sum_{i=1}^{M} \nabla_{\theta,\phi}\widetilde{\mathcal{L}}^{(i)}$ can be computed in a straightforward manner and used to update the parameters θ and ϕ with stochastic optimization methods, such as SGD.

3.3 Optimization challenges

The proposed VRL method introduces unique challenges to the variational lower bound optimization problem (see Sønderby et al. (2016) and Bowman et al. (2016) for other known challenges). To explain these challenges, we first break down VRL's parameter updating process into the following steps (using a single datapoint as an example): (1) a datapoint $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ is selected; (2) sample $\mathbf{z}^{(i)} \sim q_{\phi_k}(\,\mathbf{z} \mid \mathbf{a}^{(i)}, \mathbf{b}^{(i)}\,)$ by using the current parameter ϕ_k; (3) evaluate $\widetilde{\mathcal{L}}^{(i)}$ by using ϕ_k, θ_k; (4) calculate gradients $g = \nabla_{\theta_k,\phi_k}\widetilde{\mathcal{L}}^{(i)}$; (5) use gradients g to update ϕ_k, θ_k and get new parameters ϕ_{k+1}, θ_{k+1}. This parameter updating process can be depicted with an information flow diagram shown in Fig. 2a. Ideally, we would like every path in Fig. 2a to contribute to the evaluation of all the terms in its reachable nodes in order to obtain meaningful gradients for updating its associated parameters; however, there are two situations where this is not the case. The first situation, called

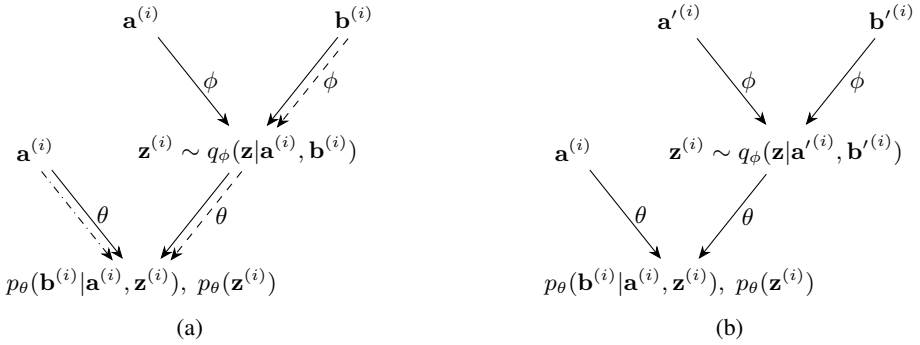


Figure 2: Information flow diagrams depicting VRL's parameter updating process, where each path uses its associated parameters to propagate information in the forward direction and gradients in the backward direction: (a) Unmodified parameter updating process, where overfitting occurs when the learning of $p_\theta(\mathbf{b}^{(i)}|\mathbf{a}^{(i)}, \mathbf{z}^{(i)})$ rely only on the dash-dotted path (*deterministic-mapping*) or the dashed path (*information-shortcut*); (b) Parameter updating process improved with RPDA.

information-shortcut, occurs when the learning of $p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i)}\,)$ rely entirely on the dashed path in Fig. 2a; more specifically, the dashed (shortcut) path directly propagates $\mathbf{b}^{(i)}$ through $\mathbf{z}^{(i)}$ to $p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i)}\,)$ and, as a result, the relational property $\mathbf{z}^{(i)}$ may learn only to encode the absolute property of $\mathbf{b}^{(i)}$. The second situation, called *deterministic-mapping*, occurs when $\mathbf{b}^{(i)}$ can be fully characterized by $\mathbf{a}^{(i)}$; in this case, the learning of $p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i)}\,)$ may rely only on the dash-dotted path in Fig. 2a. While both situations can be viewed as a overfitting problem, *deterministic-mapping* says more about the data itself and exposes a potential limitation of VRL: a decoupled relational property is primarily learned through exploring information not found in $\mathbf{a}$ but can help to better predict $\mathbf{b}$, and when $\mathbf{a}$ fully characterizes $\mathbf{b}$, VRL no longer need to explore information beyond $\mathbf{a}$ to predict $\mathbf{b}$ and this may prevent VRL from learning a meaningful relational property. On the other hand, *information-shortcut* is caused by short-cutting the parameter updating process, which we may overcome with additional regularization techniques.

Algorithm 1 VRL with RPDA

procedure VRL($\mathbf{X},\ p(\boldsymbol{\epsilon}),\ D$) ▷ If RPDA not available, $D = \{\, \mathrm{id}(\cdot) \mid (\mathbf{a}, \mathbf{b}) = \mathrm{id}(\mathbf{a}, \mathbf{b}) \,\}$
  Initialize parameters θ, ϕ
  while not convergence of parameters (θ, ϕ) **do**
    Sample minibatch $\mathbf{X}^M = \{\, (\mathbf{a}^{(i)}, \mathbf{b}^{(i)}) \mid i \in [1..M] \,\}$ from $\mathbf{X}$.
    Sample $\boldsymbol{\epsilon}^{(i,l)} \sim p(\boldsymbol{\epsilon}),\ r^{(i)} \sim \mathcal{U}(R),\ i = 1, ..., M,\ l = 1, ..., L$.
    Run RPDA and obtain $(\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) = d(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}; r^{(i)}),\ i = 1, ..., M$.
    Compute gradients $g = \nabla_{\theta,\phi} \widetilde{\mathcal{L}}\left(\theta, \phi; \mathbf{X}^M\right) = \frac{N}{M} \sum_{i=1}^{M} \nabla_{\theta,\phi} \widetilde{\mathcal{L}}^{(i)}_{\text{RPDA}}$ (see Eq. (7)).
    Update parameters θ, ϕ using gradients g (e.g. SGD).
  end while
  return θ, ϕ
end procedure

Here we propose two approaches for mitigating the *information-shortcut* problem by disrupting the flow of information passing through the shortcut path. In the first approach, we restrict the flow of information by regularizing the latent variable $\mathbf{z}$, e.g., adopting an informative prior with restrictive constraint, such as $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \sigma^2 \mathrm{I}),\ \sigma \ll 1$. In the second approach, we propose a novel data augmentation strategy—relation-preserving data augmentation (RPDA)—that aims to eliminate the shortcut path. First, we define a set of relation preserving functions $D = \{\, d(\mathbf{a}, \mathbf{b}; r) \mid r \in R$(some index set), $d: \mathcal{A} \times \mathcal{B} \to \mathcal{A} \times \mathcal{B} \,\}$ where the data relationship is preserved in the following sense: $p_\theta(\mathbf{z} \mid \mathbf{a}, \mathbf{b}) = p_\theta(\mathbf{z} \mid \mathbf{a}', \mathbf{b}')$ for $(\mathbf{a}', \mathbf{b}') = d(\mathbf{a}, \mathbf{b}; r), \forall r$. Assuming we have access to such a D, the proposed RPDA strategy then seek to optimize a modified lower bound estimator $\widetilde{\mathcal{L}}^{(i)}_{\text{RPDA}}$:

$$\widetilde{\mathcal{L}}^{(i)}_{\text{RPDA}} = \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}) + \log p_\theta(\mathbf{z}^{(i,l)}) - \log q_\phi(\mathbf{z}^{(i,l)} \mid \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}), \tag{7}$$

$$\text{where} \quad (\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) = d(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}; r^{(i)}), \quad r^{(i)} \sim \mathcal{U}(R),$$

and $\mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}, \phi),\ \boldsymbol{\epsilon}^{(i,l)} \sim p(\boldsymbol{\epsilon})$. Note that due to the relation preserving property of D, we have $q_\phi(\mathbf{z}^{(i,l)} \mid \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) = q_\phi(\mathbf{z}^{(i,l)} \mid \mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ and, therefore, $\widetilde{\mathcal{L}}^{(i)}_{\text{RPDA}}$ is equivalent to $\widetilde{\mathcal{L}}^{(i)}$ in Eq. (6). When we optimize with $\widetilde{\mathcal{L}}^{(i)}_{\text{RPDA}}$, the parameter updating process can be redrawn in Fig. 2b, where now the learning of $p_\theta(\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i)})$ can no longer rely solely on the shortcut path to propagate $\mathbf{b}'^{(i)}$ since it differs from $\mathbf{b}^{(i)}$ by a non-deterministic factor $r^{(i)}$. In practice, it may seem unrealistic to assume that we can construct a set of RPDA functions D without extensive knowledge of the underlying relational property. However, we can treat data augmentation as a form of regularization and construct a D that reflects our prior knowledge and belief of the underlying system (Ronneberger et al., 2015; Perez & Wang, 2017). For example, if we want the learning to be rotation invariance (a common theme in computer vision applications), we can construct a D that consists of image rotation augmentations, e.g., $d(\mathbf{a}, \mathbf{b}; r) = (\mathrm{rot}(\mathbf{a}, r), \mathrm{rot}(\mathbf{b}, r))$ where $\mathrm{rot}(\mathbf{x}, r)$ rotates the image $\mathbf{x}$ by $r \in R = [0, 360)$ degrees (note that both $\mathbf{a}$ and $\mathbf{b}$ are rotated by the same amount).

To summarize this section, the proposed VRL method with RPDA is described in Algorithm 1.

4 Related work

Machine learning approaches for relational processing have gained increasing interest and attention in the literature. Most of these methods focus on high-level cognitive tasks, such as visual Q&A and state prediction for complex-physics systems, and derive their relational processing capabilities from learning with clever designed neural networks (Hill et al., 2019; Santoro et al., 2017; Raposo et al., 2017; Battaglia et al., 2018; 2016; Wu et al., 2015; Reed et al., 2015; van Steenkiste et al., 2018; Chang et al., 2016; Fragkiadaki et al., 2015). Our work differ from these methods in two ways: (1) our primary focus is addressing the relational learning problem where we want to learn a decoupled relational property; (2) we enforce the decoupling requirement on the learned relational property with a PGM, which gives us the flexibility to use any compatible inference method or function approximation and still satisfy the decoupling requirement.

Conventional unsupervised learning methods can also be appied to our problem setting (Kingma & Welling, 2014; Goodfellow et al., 2014; Mikolov et al., 2013a;b); however, these methods learn a single representation for the data with superimposing information about their relational and absolute property. The difficulty of decoupling the relational property from the learned representation constitute a major obstacle to relational reasoning.

Other related work include methods on learning a disentangled representations with applications in style-transfer, image-to-image translation, domain adaptation, etc. (Huang et al., 2018; Chen et al., 2016; Tenenbaum & Freeman, 1997; Higgins et al., 2017; Bousmalis et al., 2016; Mathieu et al., 2016; Tulyakov et al., 2017; Denton & Birodkar, 2017; Villegas et al., 2017; Donahue et al., 2017; Shen et al., 2017) Most of these methods strive to learn a disentangled representations of *content* and *style* (or *pose* for video sequence data) where *content* is generically defined as the underling spatial structure, and *style* as the rendering of the structure. In comparison, our work can be viewed as learning a disentangled representations of *relational* and *absolute property*; however, we argue that *style-content* separation is fundamentally different from *relational-absolute* separation. More specifically, we view *absolute property* as incorporating both *style* and *content* information (both describe features of an individual data), while *relational property* provides new information on the (abstract) relationship between the paired data.

5 EXPERIMENT

In this section, we present experimental results from applying the proposed VRL method to a set of relational learning tasks designed with the MNIST dataset (LeCun & Cortes, 2010).

5.1 RELATIONAL LEARNING TASK

To setup a relational learning task, a paired dataset $\mathbf{X} = \{ (\mathbf{a}^{(i)}, \mathbf{b}^{(i)}) \mid i \in [1..N] \}$ was generated by the following steps: (1) the MNIST dataset $\{ (\mathbf{x}^{(i)}, y^{(i)}) \mid i \in [1..T] \}$, where $\mathbf{x}^{(i)}$ and $y^{(i)}$ are the digit images and their labels, was augmented with applying five evenly-spaced rotations to each of the image $\mathbf{x}^{(i)}$ to get $\{ (\mathbf{x}^{(i,j)}, y^{(i)}) \mid i \in [1..T], j \in [1..5] \}$ (examples of augmented images are shown in Fig. 3); (2) individual datapoint $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ of $\mathbf{X}$ was chosen to be randomly rotated images of $\mathbf{x}^{(i)}$, i.e., $\mathbf{X} = \{ (\mathbf{a}^{(i)} = \mathbf{x}^{(j,k)}, \mathbf{b}^{(i)} = \mathbf{x}^{(j,l)}) \mid j \sim \mathcal{U}([1..T]),\ k, l \sim \mathcal{U}([1..5]) \}$. Note that there are

Figure 3: Examples of a MNIST digit augmented with five evenly-spaced rotations (from left to right: $\mathbf{x}^{(i,1)}{:}0^\circ,\ \mathbf{x}^{(i,2)}{:}72^\circ,\ \mathbf{x}^{(i,3)}{:}144^\circ,\ \mathbf{x}^{(i,4)}{:}216^\circ,\ \mathbf{x}^{(i,5)}{:}288^\circ$).

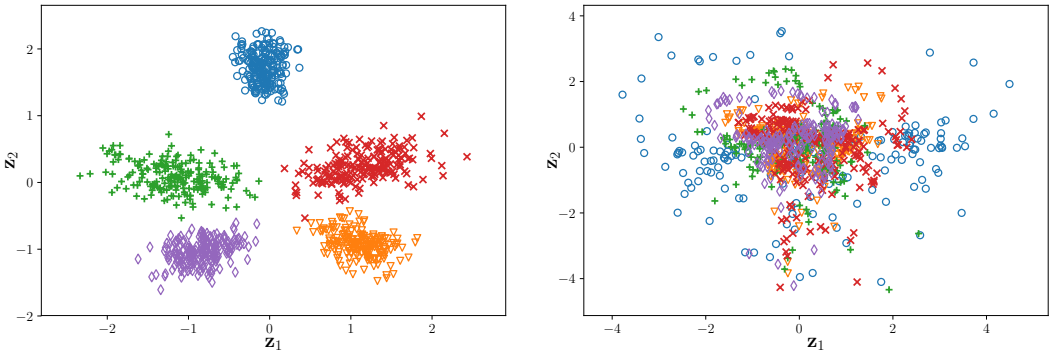
five uniquely defined rotational relationships between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ in $\mathbf{X}$ and since they are randomly selected, the rotational relationship (relational property) is decoupled from the absolute properties of $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$. Here, we use $\mathbf{X}$ to assess VRL's capability of discovering the underlying relational property (rotational relationships) irrespective of their absolute property.

5.2 IMPLEMENTATION

For these experiments, we adopted a two-dimensional latent variable $\mathbf{z} \in \mathbb{R}^2$ and let the prior $p_\theta(\,\mathbf{z}\,)$ be the bivariate normal distribution. We let $p_\theta(\,\mathbf{b} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,)$ be a multivariate Bernoulli distribution whose probability parameters are computed from a given $\mathbf{a}^{(i)}$ and $\mathbf{z}^{(i,l)}$ with an autoencoder-like neural network $f_\theta^{\text{dec}}\left(f_\theta^{\text{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,l)}\right)$. We let the approximated posterior $q_\phi(\,\mathbf{z} \mid \mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}\,)$ be a bivariate Gaussian distribution with a diagonal covariance $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2\mathbf{I})$ where $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$ are the output of a neural network $f_\phi^q(\mathbf{a'}^{(i)}, \mathbf{b'}^{(i)})$. For RPDA, we constructed a D that consists of image rotation augmentations: $D = \{ (\text{rot}(\mathbf{a}, r), \text{rot}(\mathbf{b}, r)) \mid r \in [0, 360) \}$. Detailed experimental setup is described in Appendix.

5.3 RESULTS

Relational discrimination. We trained VRL on X and used the approximated posterior $q_\phi(\mathbf{z} \mid \mathbf{a}, \mathbf{b})$ to infer the relational property of a hold-out dataset. Figure 4a shows a scatter plot of the relational property inferred by VRL where we can see that the approximated posterior accurately cluster(discriminate) data with the same(different) rotational relationship together(apart). Here, we compared VRL with variational autoencoder (VAE) (Kingma & Welling, 2014) and InfoGAN (Chen et al., 2016), both of which can learn data representations of X in a completely unsupervised manner. In the application of VAE, we adopted a 2-D latent space and use the encoder trained on X to inferred the latent varible of a hold-out dataset; the resulting scatter plot is shown in Fig. 4b where we see that VAE failed to discriminate data based on their relational property (rotational relationship). Next, InfoGAN has demonstrated its ability to learn disentangled representations



(a) Relational properties inferred by VRL

(b) Latent variable inferred by VAE

Figure 4: Scatter plots of the 2-D latent variable (showing only the mean $\boldsymbol{\mu}$) of hold-out datasets inferred by VRL in (a) and VAE in (b) (relationship labels: $\bigcirc : 0°$, $\triangledown : 72°$, $+ : 144°$, $\times : 216°$, $\diamondsuit : 288°$).

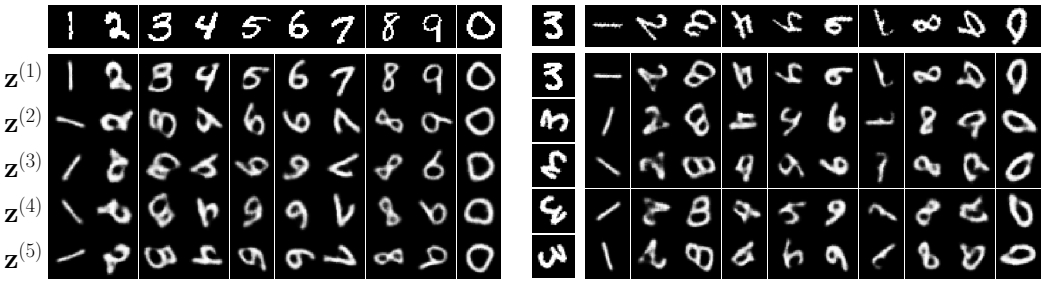
(represented by structured latent codes $c_1, c_2, ..., c_L$) through generative modelling. Although inferring latent codes for a given data point is a non-trivial task for InfoGAN, we can examine the learned latent representation by manipulating the latent codes and visually inspect the generated random samples. We followed the examples in Chen et al. (2016) and modeled the latent codes with one categorical code $c_1 \sim \text{Cat}(K = 10, p = 0.1)$ (model discontinuous variation in data) and two continuous codes $c_2, c_3 \sim \text{Unif}(-1, 1)$ (capture continuous variations in style). Figure 5 shows examples of generated images from manipulating the latent codes; it is clear that none of the latent codes (or a combination of them) distinctively capture the full range of relational property (rotational relationship). Figures 4b and 5 illustrate a major challenge of using VAE and InfoGAN (and other related methods) for relational learning: these methods learn a single representation that encodes both relational and absolute property of X and it is difficult to dissect the relational property from the learned representation. Detailed experimental setup for both VAE and InfoGAN is described in Appendix.

Relational mapping. We evaluated VRL's learned likelihood function $p_\theta(\mathbf{b} \mid \mathbf{a}, \mathbf{z})$ by visualizing the predicted images given $\mathbf{a}$ and $\mathbf{z}$. We chose $\mathbf{a}$ from a hold-out dataset and $\mathbf{z}$ from: (1) direct sampling in the latent space; (2) relational property inferred from a source datapoint $(\mathbf{a}_s, \mathbf{b}_s)$. Figure 6a shows predicted images with $\mathbf{z}$ sampled from the latent space shown in Fig. 4a. Figure 6b shows examples of relational mappings from $\mathbf{a}^{(c)}$ to $\mathbf{b}^{(r,c)}$ by applying the relational property inferred from a source datapoint $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$. At first glance, the results in Fig. 6b resemble that of style-transfer, but they are fundamentally different: in style-transfer, the image $\mathbf{b}^{(r,c)}$ is generated by applying the *style* of $\mathbf{b}_s^{(r)}$ to the *content* of $\mathbf{a}^{(c)}$, whereas VRL generates image $\mathbf{b}^{(r,c)}$ by applying the *relationship* of $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$ to the image $\mathbf{a}^{(c)}$. It is evident from Fig. 6b that predicted images $\mathbf{b}^{(r,c)}$ do not share similar *style* to $\mathbf{b}_s^{(r)}$, but rather the same rotational relationship w.r.t. $\mathbf{a}^{(c)}$ and $\mathbf{a}_s$.

(a) Varying c_2 from -2 to 2 on InfoGAN

(b) Varying c_3 from -2 to 2 on InfoGAN

Figure 5: Manipulating latent codes of InfoGAN on MNIST where each row represents random samples from varying continuous latent code c_2 in (a) and c_3 in (b) while other latent codes and noise are fixed; different rows correspond to different categorical code c_1.



(a) (b)

Figure 6: Examples of images predicted by VRL: (a) images predicted from sampled latent variables (sampling the centroid of each cluster in Fig. 4a: "○" $\rightarrow \mathbf{z}^{(1)}$, "▽" $\rightarrow \mathbf{z}^{(2)}$,"+" $\rightarrow \mathbf{z}^{(3)}$, "×" $\rightarrow \mathbf{z}^{(4)}$, "◇" $\rightarrow \mathbf{z}^{(5)}$) and each image $\mathbf{b}^{(r,c)}, 1 \leq r \leq 5, 1 \leq c \leq 10$, was predicted from an image $\mathbf{a}^{(c)}$ (shown in the top row) and a pre-selected latent variable $\mathbf{z}^{(r)}$ using $\mathbf{b}^{(r,c)} \sim p_\theta(\,\mathbf{b} \mid \mathbf{a}^{(c)}, \mathbf{z}^{(r)}\,)$; (b) examples of relational mappings of top row images by applying relationships inferred from pairs of source images $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$ (shown in the left-most column with $\mathbf{a}_s, \mathbf{b}_s^{(1)}, ..., \mathbf{b}_s^{(5)}$ arranged from top to bottom) and each image $\mathbf{b}^{(r,c)}, 1 \leq r \leq 5, 1 \leq c \leq 10$ was generated by $\mathbf{b}^{(r,c)} \sim p_\theta(\,\mathbf{b} \mid \mathbf{a}^{(c)}, \mathbf{z}^{(r)}\,)$ where $\mathbf{z}^{(r)} \sim q_\phi(\,\mathbf{z} \mid \mathbf{a}_s, \mathbf{b}_s^{(r)}\,)$.

6 Discussion and conclusion

The proposed method comes with both advantages and disadvantages: the main advantage of VRL lies in its relational learning capabilities; however, this may also be one of its disadvantages. More specifically, VRL can learn a decoupled relational property even when it is coupled with the absolute property, i.e., VRL is oblivious to the coupling information between the two properties (an example is provided in Appendix). Nevertheless, such information may be of interest to the user, and in this regard, VRL provides only a partial view of the data.

In conclusion, the proposed VRL method is an efficient and effective unsupervised learning method for addressing the relational learning problem where our goal is to learn a decoupled relational property. By dissecting the data information into decoupled *relational* and *absolute property*, we hope VRL can bring new insight into everyday data analysis and ultimately find applications for a wide variety of problems.

REFERENCES

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *CoRR*, abs/1608.06019, 2016.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002.

Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics. *CoRR*, abs/1612.00341, 2016.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016.

Emily L Denton and vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4414–4423. Curran Associates, Inc., 2017.

Chris Donahue, Akshay Balsubramani, Julian J. McAuley, and Zachary C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. *CoRR*, abs/1705.07904, 2017.

Leonidas A. A. Doumas and John E. Hummel. Comparison and mapping facilitate relation discovery and predication. *PLOS ONE*, 8(6):1–8, 06 2013. doi: 10.1371/journal.pone.0063889.

Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards, 2015.

Dedre Gentner. Language as cognitive tool kit: How language supports relational thought. *American Psychologist*, 71(8):650–657, 11 2016. ISSN 0003-066X. doi: 10.1037/amp0000082.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Felix Hill, Adam Santoro, David G. T. Barrett, Ari S. Morcos, and Timothy Lillicrap. Learning to Make Analogies by Contrasting Abstract Relational Structure. *arXiv e-prints*, art. arXiv:1902.00120, January 2019.

Keith J Holyoak. Analogy and relational reasoning. In Keith J Holyoak and Robert G Morrison (eds.), *The Oxford Handbook of Thinking and Reasoning*, chapter 13, pp. 234–259. Oxford University Press, New York, 2012.

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *CoRR*, abs/1804.04732, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2014.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

Michaël Mathieu, Junbo Jake Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. Disentangling factors of variation in deep representations using adversarial training. *CoRR*, abs/1611.03383, 2016.

Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner. Respects for similarity. *PSYCHOLOGICAL REVIEW*, 100:254–278, 1993.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics.

John Paisley, David M. Blei, and Michael I. Jordan. Variational bayesian inference with stochastic search. In *ICML*, 2012.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

David Raposo, Adam Santoro, David G. T. Barrett, Razvan Pascanu, Timothy P. Lillicrap, and Peter W. Battaglia. Discovering objects and their relations from entangled scene representations. *CoRR*, abs/1702.05068, 2017.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 1252–1260. Curran Associates, Inc., 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4967–4976. Curran Associates, Inc., 2017.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. Style transfer from non-parallel text by cross-alignment. *CoRR*, abs/1705.09655, 2017.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. *CoRR*, abs/1602.02282, 2016.

Joshua B. Tenenbaum and William T. Freeman. Separating style and content. In M. C. Mozer, M. I. Jordan, and T. Petsche (eds.), *Advances in Neural Information Processing Systems 9*, pp. 662–668. MIT Press, 1997.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *CoRR*, abs/1707.04993, 2017.

Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *CoRR*, abs/1802.10353, 2018.

Gary R VandenBos and American Psychological Association. *Apa Dictionary of Psychology*. American Psychological Association, Washington, DC, 2nd edition, 2007.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *CoRR*, abs/1706.08033, 2017.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 127–135. Curran Associates, Inc., 2015.

A EXPERIMENTAL SETUP

Recall that we adopted a two-dimensional latent variable $\mathbf{z} \in \mathbb{R}^2$ and let the prior $p_\theta(\,\mathbf{z}\,)$ be the bivariate normal distribution. For binary valued data (e.g., MNIST dataset), we let the likelihood function $p_\theta(\,\mathbf{b} \in \mathcal{B} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,)$ of VRL be a multivariate Bernoulli distribution whose probability parameters $\mathbf{p}^{(i,l)}$ are computed from a given $\mathbf{a}^{(i)}$ and $\mathbf{z}^{(i,l)}$ with an autoencoder-like neural network $f_\theta^{\text{dec}}\left(f_\theta^{\text{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,l)}\right)$:

$$
\begin{aligned}
f_\theta^{\text{enc}} :&\mathbf{a}^{(i)} \to \text{Conv(3x3x8)} \to \text{Conv(3x3x32)} \to \text{Conv(3x3x128)} \to \text{FC(20)} \to \mathbf{h}^{(i)} \in \mathbb{R}^{20} \\
f_\theta^{\text{dec}} :&[\mathbf{h}^{(i)} \in \mathbb{R}^{20}, \mathbf{z}^{(i,l)} \in \mathbb{R}^2] \to \text{FC} \to \text{Conv}^T\text{(3x3x128)} \to \text{Conv}^T\text{(3x3x32)} \to \text{Conv}^T\text{(3x3x8)} \\
&\to \text{Conv(1x1x1)} \xrightarrow{\text{Sigmoid}} [0,1]^{\dim(\mathcal{B})},
\end{aligned}
$$

where $\text{Conv}(\cdot)$ is a strided (stride 2) convolutional layer and $\text{Conv}^T(\cdot)$ is a transposed convolutional layer. We used batch-normalization after most layers and leaky rectified linear units (with leaky rate 0.01) as nonlinear activation function. We let the approximated posterior $q_\phi(\,\mathbf{z} \mid \mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}\,)$ of VRL be a bivariate Gaussian distribution with a diagonal covariance $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2\mathbf{I})$ where $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$ are the output of a neural network $f_\phi^q(\mathbf{a'}^{(i)}, \mathbf{b'}^{(i)})$:

$$
f_\phi^q : [\mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}] \to \text{Conv(3x3x8)} \to \text{Conv(3x3x32)} \to \text{Conv(3x3x128)} \to \text{FC(4)} \to [\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}].
$$

We sampled from the posterior $\mathbf{z}^{(i,l)} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2\mathbf{I})$ using $\mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}, \phi) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(i,l)}$ where $\boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and set $L = 1$. We used image rotation augmentations: $D = \{\,(\text{rot}(\mathbf{a}, r), \text{rot}(\mathbf{b}, r)) \mid r \in [0, 360)\,\}$ for constructing RPDA. In this case, the learning objective $\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)}$ in Eq. (7) can further be derived as (see next section for the derivation):

$$
\begin{aligned}
\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)} = \frac{1}{2}\sum_{j=1}^{2}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) \\
+ \sum_k b_k^{(i)} \log p_k^{(i,1)} + (1 - b_k^{(i)}) \log(1 - p_k^{(i,1)}).
\end{aligned} \tag{8}
$$

$$
\begin{aligned}
\text{where} \quad (\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}) = f_\phi^q(\mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}), \quad \mathbf{p}^{(i,l)} = f_\theta^{\text{dec}}\left(f_\theta^{\text{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,1)}\right), \\
\mathbf{z}^{(i,1)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(i,1)}, \quad \boldsymbol{\epsilon}^{(i,1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\
(\mathbf{a'}^{(i)}, \mathbf{b'}^{(i)}) = (\text{rot}(\mathbf{a}^{(i)}, r^{(i)}), \text{rot}(\mathbf{b}^{(i)}, r^{(i)})), \quad r^{(i)} \sim \mathcal{U}([0, 360)).
\end{aligned}
$$

Parameters θ and ϕ were jointly trained to maximize Eq. (8) using Adam optimizer (learning rate = 0.0001, $\beta 1 = 0.9,\ \beta 2 = 0.999$) (Kingma & Ba, 2014); minibatches of size $M = 100$ were used.

In our VAE implementation we used the following network architecture for encoder, $\text{Enc}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$, and decoder, $\text{Dec}(\mathbf{z}^{(i,l)})$:

$$
\begin{aligned}
\text{Enc} :&[\mathbf{a}^{(i)}, \mathbf{b}^{(i)}] \to \text{Conv(3x3x16)} \to \text{Conv(3x3x64)} \to \text{Conv(3x3x256)} \to \text{FC(4)} \\
&\to [\boldsymbol{\mu}^{(i)} \in \mathbb{R}^2, \boldsymbol{\sigma}^{(i)} \in \mathbb{R}^2] \\
\text{Dec} :&[\mathbf{z}^{(i,l)} \in \mathbb{R}^2] \to \text{FC} \to \text{Conv}^T\text{(3x3x256)} \to \text{Conv}^T\text{(3x3x64)} \to \text{Conv}^T\text{(3x3x16)} \\
&\to \text{Conv(1x1x1)} \xrightarrow{\text{Sigmoid}} [0,1]^{\dim(\mathcal{B})}.
\end{aligned}
$$

We used Adam optimizer with the same hyperparameter setup as before.

For InfoGAN implementation, we followed the setup described in Chen et al. (2016) except that each datapoint consists a paired MNIST image $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$.

A.1 DERIVATION OF TRAINING OBJECTIVE FUNCTION

Our derivation for Eq. (8) largely follow the work of Kingma & Welling (2014); here for completeness we outline the key steps. First, Eq. (4) in Section 3.2 can equivalently be expressed as:

$$\begin{aligned}\mathcal{L}^{(i)} &= -D_{\mathrm{KL}}\Big(q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)}) \,\Big\|\, p_\theta(\,\mathbf{z}\,)\Big) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)})}\Big[\log p_\theta(\,\mathbf{b}^{(i)}|\mathbf{a}^{(i)},\mathbf{z}\,)\Big] \\ &= -D_{\mathrm{KL}}\Big(q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)}) \,\Big\|\, p_\theta(\,\mathbf{z}\,)\Big) + \mathbb{E}_{p(\boldsymbol{\epsilon})}\Big[\log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)},\mathbf{z}\,)\Big]\,, \end{aligned} \tag{9}$$

where $\mathbf{z} = g(\boldsymbol{\epsilon}, \mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \phi)$, $\boldsymbol{\epsilon} \sim p(\,\boldsymbol{\epsilon}\,)$, and g is some differentiable and invertible transformation. We can approximate $\mathcal{L}^{(i)}$ in Eq. (9) with a Monte Carlo estimator $\widetilde{\mathcal{L}}^{(i)}$:

$$\widetilde{\mathcal{L}}^{(i)} = -D_{\mathrm{KL}}\Big(q_\phi(\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)}) \,\Big\|\, p_\theta(\,\mathbf{z}\,)\Big) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)},\mathbf{z}^{(i,l)}\,), \tag{10}$$

where $\mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \phi)$ and $\boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,)$. Based on the RPDA functions $D = \{\, d(\mathbf{a},\mathbf{b};r) \mid r \in R \,\}$ and its relation preserving assumption that $q_\phi(\,\mathbf{z}|\mathbf{a}^{(i)},\mathbf{b}^{(i)}\,) = q_\phi(\,\mathbf{z}|\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}\,)$ for $(\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}) = d(\mathbf{a}^{(i)},\mathbf{b}^{(i)};r), \forall r$ (see Section 3.3), we can express $\widetilde{\mathcal{L}}^{(i)}$ in Eq. (10) equivalently as $\widetilde{\mathcal{L}}^{(i)}_{\mathrm{RPDA}}$:

$$\widetilde{\mathcal{L}}^{(i)}_{\mathrm{RPDA}} = -D_{\mathrm{KL}}\Big(q_\phi(\,\mathbf{z}|\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}\,) \,\Big\|\, p_\theta(\,\mathbf{z}\,)\Big) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)},\mathbf{z}^{(i,l)}\,), \tag{11}$$

$$\text{where} \qquad \mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}, \phi), \quad \boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,),$$
$$(\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}) = d(\mathbf{a}^{(i)},\mathbf{b}^{(i)};r^{(i)}), \quad r^{(i)} \sim \mathcal{U}(R).$$

The first term in Eq. (11) is the KL-divergence from $p_\theta(\,\mathbf{z}\,)$ to $q_\phi(\,\mathbf{z}|\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}\,)$, which can be computed analytically given that we assume $p_\theta(\,\mathbf{z}\,) = \mathcal{N}(\mathbf{0},\mathbf{I})$ and $q_\phi(\,\mathbf{z}|\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}\,) = \mathcal{N}(\mathbf{z};\boldsymbol{\mu}^{(i)},(\boldsymbol{\sigma}^{(i)})^2\mathbf{I})$:

$$-D_{\mathrm{KL}}\Big(q_\phi(\,\mathbf{z}|\mathbf{a}'^{(i)},\mathbf{b}'^{(i)}\,) \,\Big\|\, p_\theta(\,\mathbf{z}\,)\Big) = \frac{1}{2}\sum_{j=1}^{2}\Big(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\Big)\,. \tag{12}$$

The likelihood function $p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)},\mathbf{z}^{(i,l)}\,)$ is defined as a multivariate Bernoulli distribution whose probability parameters $\mathbf{p}^{(i,l)}$ are computed from the neural network $f_\theta^{\mathrm{dec}}\left(f_\theta^{\mathrm{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,l)}\right)$ and we have:

$$\log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)},\mathbf{z}^{(i,l)}\,) = \sum_k b_k^{(i)} \log p_k^{(i,l)} + (1 - b_k^{(i)}) \log(1 - p_k^{(i,l)}). \tag{13}$$

By substituting Eq. (12) and (13) back in (11) and recall that $\mathbf{z}^{(i,l)} = g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}, \phi) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(i,l)}$, $p(\,\boldsymbol{\epsilon}\,) = \mathcal{N}(\mathbf{0},\mathbf{I})$, $D = \{\, (\mathrm{rot}(\mathbf{a},r), \mathrm{rot}(\mathbf{b},r)) \mid r \in [0,360) \,\}$, $L = 1$, we arrive at Eq. (8).

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 MNIST DATASET

Here, we provide additional experimental results based on the paired MNIST dataset constructed in Section 5.

First, to further test the robustness of the proposed method, we considered a scenarios where the underlying relational property is coupled with the absolute property. An example dataset $\mathbf{X}^2$ was constructed with the rotational relationship between each of the datapoint completely determined by its digit label: $\mathbf{X}^2 = \{\, (\mathbf{a}^{(i)} = \mathbf{x}^{(j,k)}, \mathbf{b}^{(i)} = \mathbf{x}^{(j,l+1)}) \mid j \sim \mathcal{U}([1..T]),\ k \sim \mathcal{U}([1..5]), l = (k + \lfloor y^{(j)}/2 \rfloor) \bmod 5 \,\}$. In this case, it is possible to infer the relational property solely based on the image representation of the digit (absolute property of $\mathbf{a}^{(i)}$), for example, $\mathbf{a}^{(i)} \in$ ['0', '1'] $\rightarrow 0°$

(read: if $\mathbf{a}^{(i)}$ is recognized as either digit 0 or 1, the rotational relationship between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ is $0°$) , $\mathbf{a}^{(i)} \in$ ['2', '3'] $\rightarrow 72°$, $\mathbf{a}^{(i)} \in$ ['4', '5'] $\rightarrow 144°$, $\mathbf{a}^{(i)} \in$ ['6', '7'] $\rightarrow 216°$, $\mathbf{a}^{(i)} \in$ ['8', '9'] $\rightarrow 288°$. The question then arise: is VRL capable of learning a decoupled relational property even when it is coupled with the absolute property? To test this idea, we trained VRL (with the same setup as that described in Appendix A) on $\mathbf{X}^2$ but validated it on a hold-out dataset with a decoupled relational property (much like how $\mathbf{X}$ was constructed). The resulting scatter plot is shown in Fig. 7, where we can see that the approximated posterior $q_\phi(\,\mathbf{z} \mid \mathbf{a}, \mathbf{b}\,)$ accurately cluster(discriminate) data with the same(different) rotational relationship together(apart). This result shows that VRL was indeed capable of learning a decoupled relational property irrespective of the digit representation of $\mathbf{a}^{(i)}$. If this were not the case, we would expect to see a scatter plot with heavily overlapped relationship labels since the validation dataset was constructed with random rotational relationships. Examples

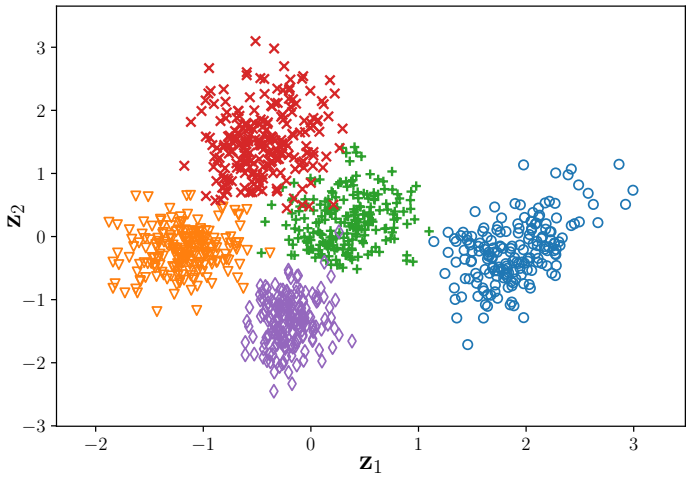
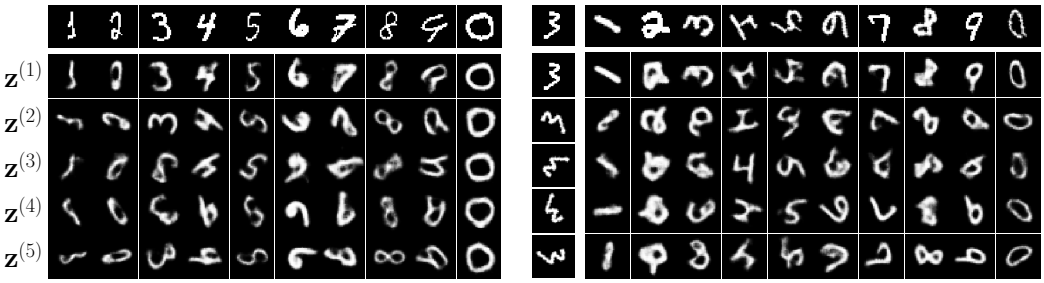


Figure 7: Scatter plot of the relational property (showing only the mean $\boldsymbol{\mu}$) of a hold-out dataset inferred by a VRL model that was trained on $\mathbf{X}^3$ (relationship labels: $\bigcirc : 0°$, $\triangledown : 72°$, $+ : 144°$, $\times : 216°$, $\diamondsuit : 288°$).

of images predicted by the learned likelihood function $p_\theta(\,\mathbf{b} \mid \mathbf{a}, \mathbf{z}\,)$ are shown in Fig. 8, where Fig. 8a shows predicted images based on direct sampling in the latent space (shown in Fig. 7), and Fig. 8b shows examples of relational mappings. Figures 8a and 8b further corroborate our claim that



(a) (b)

Figure 8: Examples of images predicted by a VRL model that was trained on $\mathbf{X}^2$: (a) images predicted from sampled latent variables (sampling the centroid of each cluster in Fig. 7: "$\bigcirc$" $\rightarrow \mathbf{z}^{(1)}$, "$\triangledown$" $\rightarrow \mathbf{z}^{(2)}$, "$+$" $\rightarrow \mathbf{z}^{(3)}$, "$\times$" $\rightarrow \mathbf{z}^{(4)}$, "$\diamondsuit$" $\rightarrow \mathbf{z}^{(5)}$); (b) examples of relational mappings of top row images by applying relationships inferred from pairs of source images $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$ shown in the left-most column with $\mathbf{a}_s, \mathbf{b}_s^{(1)}, ..., \mathbf{b}_s^{(5)}$ arranged from top to bottom.

VRL is capable of learning a decoupled relational property and generalizing it to unseen data, e.g., VRL trained on $\mathbf{X}^2$ learned to rotate *any* digit by *any* amount despite not having seen most of the

digit-rotation pairs during training. Note that this experiment is an example of *deterministic-mapping* discussed in Section 3.3, nevertheless VRL was able to utilize all paths in Fig. 2b effectively to learn about the data.

Next, we setup a more complex relational learning task that includes both rotational and resizing relationships between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$: each MNIST image $\mathbf{x}^{(i)}$ was augmented with five evenly-spaced rotations and three different resizing transformations to get $\{ (\mathbf{x}^{(i,j,k)}, y^{(i)}) \mid i \in [1..T], j \in [1..5], k \in [1..3] \}$. Examples of augmented images are shown in Fig. 9, where top, middle, and bottom row images are resized by a factor of $\times 0.66$, $\times 1$, and $\times 1.5$, respectively. We considered a relational

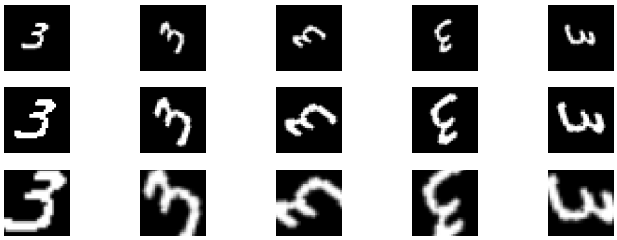


Figure 9: Examples of a MNIST digit augmented with rotational and resizing transformations (from left to right: $\mathbf{x}^{(i,1,k)}: 0°$, $\mathbf{x}^{(i,2,k)}: 72°$, $\mathbf{x}^{(i,3,k)}: 144°$, $\mathbf{x}^{(i,4,k)}: 216°$, $\mathbf{x}^{(i,5,k)}: 288°$; from top to bottom: $\mathbf{x}^{(i,j,1)}: \times 0.66$, $\mathbf{x}^{(i,j,2)}: \times 1$, $\mathbf{x}^{(i,j,3)}: \times 1.5$).

learning task where each datapoint $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ has a decoupled rotational and/or resizing relational property; more specifically, we constructed a dataset $\mathbf{X}^3$ in the following way:

$$\{ (\mathbf{a}^{(i)} = \mathbf{x}^{(j,k,m)}, \mathbf{b}^{(i)} = \mathbf{x}^{(j,l,n)}) \mid j \sim \mathcal{U}([1..T]), k, l \sim \mathcal{U}([1..5]), m \sim \mathcal{U}([1,2]), n \sim \mathcal{U}(V) \}$$
$$\text{where} \qquad V = \begin{cases} [1,2], & \text{if } m = 1 \\ [2,3], & \text{if } m = 2 \end{cases}$$

Note that in this case, $\mathbf{b}^{(i)}$ is either the same size as $\mathbf{a}^{(i)}$ or is $\times 1.5$ larger than $\mathbf{a}^{(i)}$, and there are a total of 10 different relationships between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ in $\mathbf{X}^3$ (combinations of 5 rotational and 2 resizing transformations). We trained VRL on $\mathbf{X}^3$ with the same model setup as that described in Appendix A and used the trained model to infer the relational property of a hold-out dataset. The inference result is shown in Fig. 10, where we can see that the approximated posterior $q_\phi(\mathbf{z} \mid \mathbf{a}, \mathbf{b})$ accurately cluster(discriminate) data with the same(different) relationship together(apart). Examples

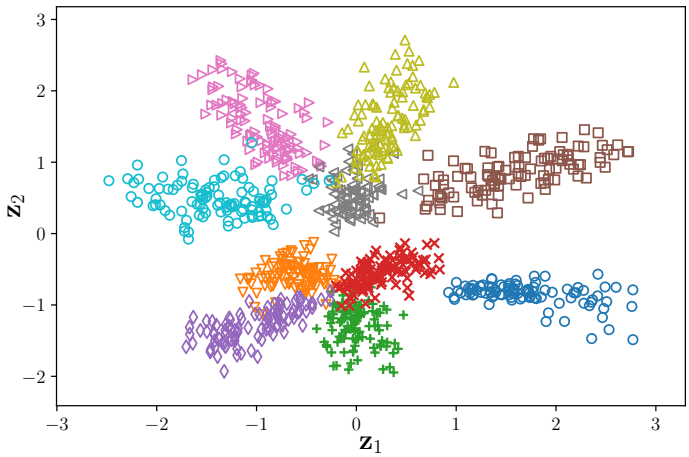
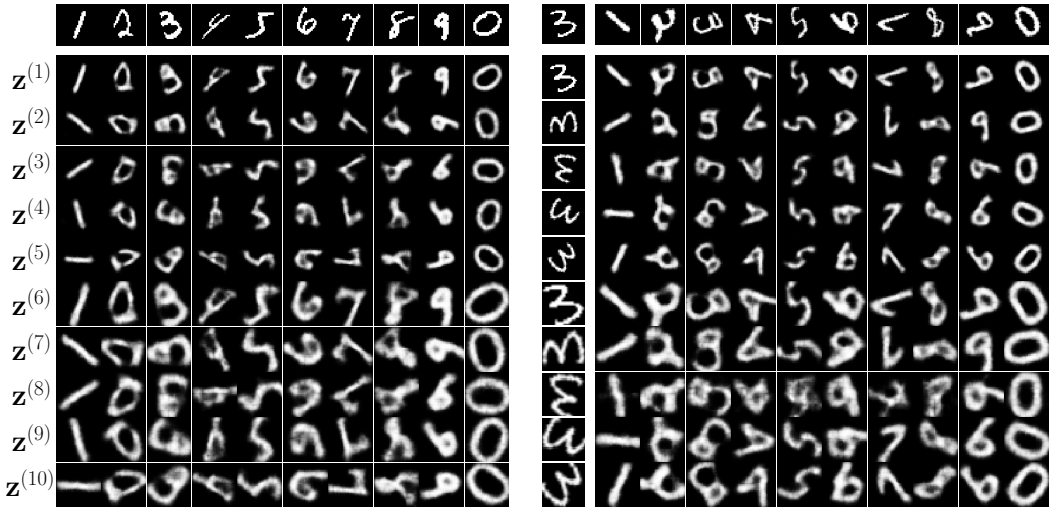


Figure 10: Scatter plot of the relational property (showing only the mean $\boldsymbol{\mu}$) of a hold-out dataset inferred by a VRL model that was trained on $\mathbf{X}^3$ (relationship labels: ○(blue) : 0°, ▽ : 72°, + : 144°, × : 216°, ◇ : 288°, □ : 0°, ×1.5, ▷ : 72°, ×1.5, ◁ : 144°, ×1.5, △ : 216°, ×1.5, ○(cyan) : 288°, ×1.5).

of images predicted by the learned likelihood function $p_\theta(\mathbf{b} \mid \mathbf{a}, \mathbf{z})$ are shown in Fig. 11, where

Fig. 11a shows predicted images based on direct sampling in the latent space (shown in Fig. 10), and Fig. 11b shows examples of relational mappings. These results are consistent with the findings presented and discussed in Section 5.3.



(a) (b)

Figure 11: Examples of images predicted by a VRL model that was trained on $\mathbf{X}^3$: (a) images predicted from sampled latent variables (sampling the centroid of each cluster in Fig. 10: "○"(blue)$\rightarrow \mathbf{z}^{(1)}$, "▽"$\rightarrow \mathbf{z}^{(2)}$,"+"$\rightarrow \mathbf{z}^{(3)}$, "×"$\rightarrow \mathbf{z}^{(4)}$, "◇"$\rightarrow \mathbf{z}^{(5)}$, "□"$\rightarrow \mathbf{z}^{(6)}$, "▷"$\rightarrow \mathbf{z}^{(7)}$, "◁"$\rightarrow \mathbf{z}^{(8)}$, "△"$\rightarrow \mathbf{z}^{(9)}$, "○"(cyan) $\rightarrow \mathbf{z}^{(10)}$); (b) examples of relational mappings of top row images by applying relationships inferred from pairs of source images $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$ shown in the left-most column with $\mathbf{a}_s, \mathbf{b}_s^{(1)}, ..., \mathbf{b}_s^{(10)}$ arranged from top to bottom.

Finally, we present an example with a *continuous* relational property. Based on the MNIST dataset $\{\, (\mathbf{x}^{(i)}, y^{(i)}) \mid i \in [1..T] \,\}$, we constructed a paired dataset $\mathbf{X}^4$ in the following way:

$$\mathbf{X}^4 = \{\, (\mathbf{a}^{(i)} = \mathbf{x}^{(j)}, \mathbf{b}^{(i)} = \mathrm{rot}(\mathbf{x}^{(j)}, r^{(i)})) \mid j \sim \mathcal{U}([1..T]), r^{(i)} \sim \mathcal{U}([0, 360)) \,\},$$

where $\mathrm{rot}(\mathbf{x}^{(j)}, r^{(i)})$ rotates the image $\mathbf{x}^{(j)}$ by $r^{(i)} \in [0, 360)$ degrees. In this case, $\mathbf{b}^{(i)}$ is a random rotation of $\mathbf{a}^{(i)}$ and there is a continuous (and decoupled) rotational relationship between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ in $\mathbf{X}^4$. We trained VRL on $\mathbf{X}^4$ with the same model setup as that described in Appendix A and used the trained model to infer the relational property of a hold-out dataset. A scatter plot of the relational property infered by the approximated posterior $q_\phi(\, \mathbf{z} \mid \mathbf{a}, \mathbf{b} \,)$ is shown in Fig. 12, and examples of images predicted by the learned likelihood function $p_\theta(\, \mathbf{b} \mid \mathbf{a}, \mathbf{z} \,)$ are shown in Fig. 13, where Fig. 13a shows predicted images based on direct sampling in the latent space (denoted by markers "×" in Fig. 12), and Fig. 13b shows examples of relational mappings. From Fig. 12 and 13 we can see that VRL learned a decoupled relational property that encodes a continuous data (rotational) relationship; however, there is a small region in Fig. 12 with overlapping relational property that leads to an ambiguous interpretation of the rotational relationship ($120°$ vs. $240°$). This ambiguity is likely caused by compressing the continuous data (rotational) relationship down to a two-dimensional latent space, $\mathbf{z} \in \mathbb{R}^2$, and motivates us to adopt a higher-dimensional latent space, e.g., $\mathbf{z} \in \mathbb{R}^3$. Figure 14 shows inference result from repeating the previous experiment but with setting $\mathbf{z} \in \mathbb{R}^3$; we can see that VRL learned a three-dimensional relational property that unambiguously represents the underlying continuous data (rotational) relationship.

B.2 YALE FACE DATASET

Here we present another relational learning example based on the Yale face dataset (Belhumeur et al., 1997). The Yale face dataset consists of 15 subjects, each with 8 facial expressions and 3

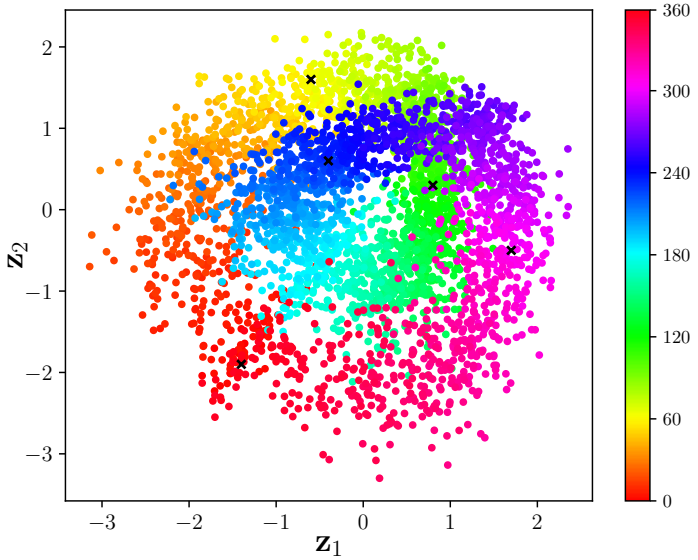
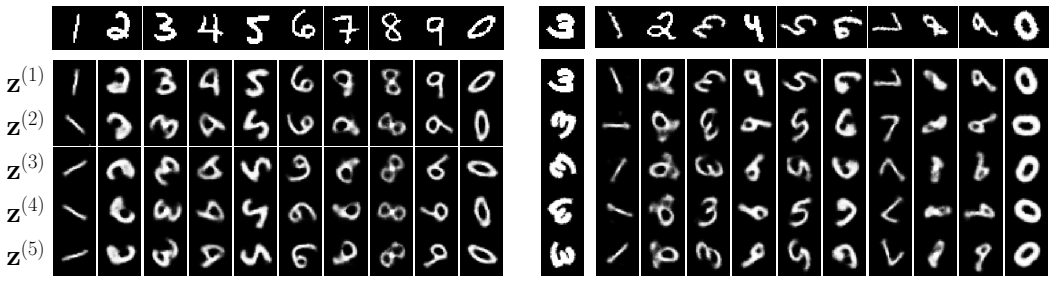


Figure 12: Scatter plot of the relational property (showing only the mean $\boldsymbol{\mu}$) of a hold-out dataset inferred by a VRL model that was trained on $\mathbf{X}^4$; each point is color-coded (best viewed in color) by the degrees of rotation between the corresponding datapoint (markers "$\times$" denote sampled latent varibles $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(5)}$ used for image prediction in Fig. 13a).



(a) (b)

Figure 13: Examples of images predicted by a VRL model that was trained on $\mathbf{X}^4$: (a) images predicted from sampled latent variables (denoted by markers "$\times$" in Fig. 12); (b) examples of relational mappings of top row images by applying relationships inferred from pairs of source images $(\mathbf{a}_s, \mathbf{b}_s^{(r)})$ shown in the left-most column with $\mathbf{a}_s, \mathbf{b}_s^{(1)}, \ldots, \mathbf{b}_s^{(5)}$ arranged from top to bottom.

lighting configurations. We extracted three facial expressions (happy, surprised, sad) of each subject to form $\{\, \mathbf{x}^{(i,j)} \mid i \in [1..15], j \in [1..3] \,\}$. Examples of face images are shown in Fig. 15a, where we center-cropped, resized to 64×64, and normalized pixel values to be within $[0, 5]$. We constructed a dataset $\mathbf{X}^F$ where each datapoint $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ represents a subject with different facial expressions (*emotions*):

$$\mathbf{X}^F = \{\, (\mathbf{a}^{(i)} = \mathbf{x}^{(j,k)}, \mathbf{b}^{(i)} = \mathbf{x}^{(j,l)}) \mid j \sim \mathcal{U}([1..15]),\ k \sim \mathcal{U}([1..3]),\ l \sim \mathcal{U}([1..3] \setminus k) \,\}.$$

Our initial intention was to apply VRL to $\mathbf{X}^F$ to learn about the *"emotional change"* between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ irrespective of the subject; however, because $\mathbf{X}^F$ is an extremely limited dataset that consists of only 45 different images (15 subjects, each with 3 facial expressions), we settled for a less ambitious goal of learning an *undirected* emotional change, i.e., "happy→sad" = "sad→happy" (simply denote as "happy-sad"). In this case, there are three different emotional changes between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ in $\mathbf{X}^F$: "happy-sad", "happy-surprised", "surprised-sad". To setup a VRL model for training, we adopted a two-dimensional latent variable $\mathbf{z} \in \mathbb{R}^2$ and let the prior $p_\theta(\mathbf{z})$ be the

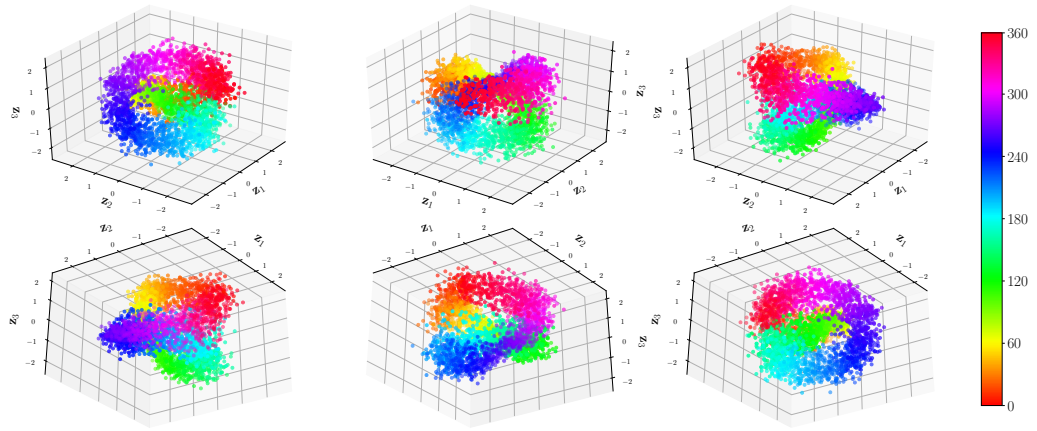
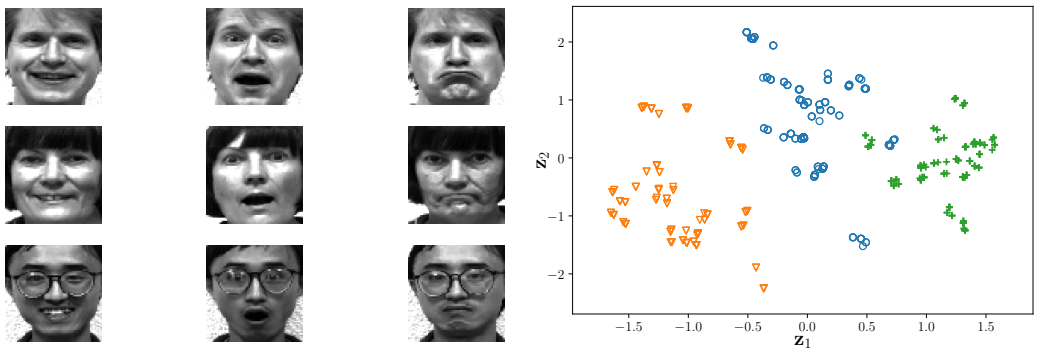


Figure 14: Scatter plot of the three-dimensional relational property (showing only the mean $\boldsymbol{\mu}$) of a hold-out dataset inferred by a VRL model (with $\mathbf{z} \in \mathbb{R}^3$) that was trained on $\mathbf{X}^4$; each plot shows a different vantage point of the 3D scatter plot, and each point is color-coded (best viewed in color) by the degrees of rotation between the corresponding datapoint.



(a) (b)

Figure 15: Learning emotional changes among the Yale face dataset: (a) examples of subjects with different facial expressions: happy, surprised, and sad; (b) scatter plot of the relational property (showing only the mean $\boldsymbol{\mu}$) inferred by the approximated posterior (relationship labels: $\triangledown$: "happy-sad", $\bigcirc$: "happy-surprised", $+$: "surprised-sad").

bivariate normal distribution. Since this is a real-valued dataset, we let $p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,)$ be a multivariate Gaussian distribution with a fixed diagonal covariance $\mathcal{N}(\mathbf{b}^{(i)}; \boldsymbol{\mu}_{\mathbf{b}}^{(i,l)}, \boldsymbol{\sigma}^2\mathbf{I})$ where $\boldsymbol{\sigma} = 0.1$ and $\boldsymbol{\mu}_{\mathbf{b}}^{(i,l)}$ is computed from a given $\mathbf{a}^{(i)}$ and $\mathbf{z}^{(i,l)}$ with an autoencoder-like neural network $f_\theta^{\text{dec}}\left(f_\theta^{\text{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,l)}\right)$:

$$f_\theta^{\text{enc}} : \mathbf{a}^{(i)} \rightarrow \text{Conv}(3\text{x}3\text{x}4) \rightarrow \text{Conv}(3\text{x}3\text{x}16) \rightarrow \text{Conv}(3\text{x}3\text{x}64) \rightarrow \text{FC}(50) \rightarrow \mathbf{h}^{(i)} \in \mathbb{R}^{50}$$
$$f_\theta^{\text{dec}} : [\mathbf{h}^{(i)} \in \mathbb{R}^{50}, \mathbf{z}^{(i,l)} \in \mathbb{R}^2] \rightarrow \text{FC} \rightarrow \text{Conv}^T(3\text{x}3\text{x}64) \rightarrow \text{Conv}^T(3\text{x}3\text{x}16) \rightarrow \text{Conv}^T(3\text{x}3\text{x}4)$$
$$\rightarrow \text{Conv}(1\text{x}1\text{x}1) \rightarrow \boldsymbol{\mu}_{\mathbf{b}}^{(i,l)}$$

and we have:

$$\log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,) = -\frac{1}{2}\frac{\left\|\mathbf{b}^{(i)} - \boldsymbol{\mu}_{\mathbf{b}}^{(i,1)}\right\|^2}{0.01} + \text{const.} \tag{14}$$

Again, we let the approximated posterior $q_\phi(\,\mathbf{z} \mid \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}\,)$ be a bivariate Gaussian distribution with a diagonal covariance $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, (\boldsymbol{\sigma}^{(i)})^2\mathbf{I})$ where $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$ are the output of a neural network

$f_\phi^q(\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)})$:

$$f_\phi^q : [\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}] \rightarrow \text{Conv(3x3x4)} \rightarrow \text{Conv(3x3x16)} \rightarrow \text{Conv(3x3x64)} \rightarrow \text{FC(4)} \rightarrow [\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}]$$

Next, based on the premise that we are only interested in learning an undirected emotional change, we have $p_\theta(\,\mathbf{z} \mid \mathbf{a}, \mathbf{b}\,) = p_\theta(\,\mathbf{z} \mid \mathbf{b}, \mathbf{a}\,)$ and this prompted us to construct RPDA functions D with random image rotation and swapping operations: $D = \{\, \text{swap}\,(\text{rot}(\mathbf{a}, r), \text{rot}(\mathbf{b}, r)) \mid r \in [0, 360)\,\}$ where

$$\text{swap}(\mathbf{a}, \mathbf{b}) = \begin{cases} (\mathbf{a}, \mathbf{b}), & p = 0.5 \\ (\mathbf{b}, \mathbf{a}), & p = 0.5 \end{cases}$$

Finally, we combine the above settings with $\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(i,l)}$, $p(\,\boldsymbol{\epsilon}\,) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $L = 1$, and substituting Eq. (12) and (14) back in (11) to derive the following lower bound estimator:

$$\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)} = \frac{0.01}{2} \sum_{j=1}^{2} \left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right) - \frac{1}{2} \left\| \mathbf{b}^{(i)} - \boldsymbol{\mu}_{\mathbf{b}}^{(i,1)} \right\|^2 + \text{const.} \tag{15}$$

$$\begin{aligned} \text{where} \quad (\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}) &= f_\phi^q(\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}), \quad \boldsymbol{\mu}_{\mathbf{b}}^{(i,1)} = f_\theta^{\text{dec}}\left(f_\theta^{\text{enc}}(\mathbf{a}^{(i)}),\ \mathbf{z}^{(i,1)} \right), \\ \mathbf{z}^{(i,1)} &= \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(i,1)}, \quad \boldsymbol{\epsilon}^{(i,1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ (\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) &= \text{swap}\big(\text{rot}(\mathbf{a}, r^{(i)}), \text{rot}(\mathbf{b}, r^{(i)})\big), \quad r^{(i)} \sim \mathcal{U}([0, 360)). \end{aligned}$$

The rest of the training setup remains the same as that described in Appendix A. We trained VRL on $\mathbf{X}^F$ and then used the approximated posterior $q_\phi(\,\mathbf{z} \mid \mathbf{a}, \mathbf{b}\,)$ to infer the relational property of $\mathbf{X}^F$. The inference result is shown in Fig. 15b, where we can see that VRL learned a relational property that accurately differentiates emotional changes irrespective of the subject. It is worth noting that VRL does not learn the facial expression for each of $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$ independently and then classify them based on their difference, but instead directly learns the emotional relationship between $(\mathbf{a}^{(i)}, \mathbf{b}^{(i)})$.

C Ablation Study

In this work, we introduced relation-preserving data augmentation (RPDA) as a practical data augmentation strategy for overcoming the information-shortcut problem—an unique overfitting problem introduced by VRL (see Section 3.3). In order to evaluate the contribution of RPDA, we performed ablation study on the paired MNIST dataset $\mathbf{X}$ constructed in see Section 5.1. First, recall that in Section 5.2 we constructed RPDA functions D with image rotation augmentations and optimized the following lower bound estimator (cf. Eq. (7)):

$$\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)} = \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\,\mathbf{b}^{(i)} \mid \mathbf{a}^{(i)}, \mathbf{z}^{(i,l)}\,) + \log p_\theta(\,\mathbf{z}^{(i,l)}\,) - \log q_\phi(\,\mathbf{z}^{(i,l)} \mid \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}\,), \tag{16}$$

$$\begin{aligned} \text{where} \quad \mathbf{z}^{(i,l)} &= g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}, \phi), \quad \boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,), \\ (\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) &= d(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}; r^{(i)}), \quad r^{(i)} \sim \mathcal{U}(R). \end{aligned}$$

In the first ablation study, we experimented with removing RPDA from the VRL training, which amounts to optimizing with the original lower bound estimator $\widetilde{\mathcal{L}}^{(i)}$ in Eq. (6). With the rest of the training setup remained unchanged (see Appendix A), we trained VRL on $\mathbf{X}$ *without* RPDA and used the trained model to infer the relational property of a hold-out dataset; the inference result is shown in Fig. 16a. In the second ablation study, we repeated the VRL training, but instead of following the proposed RPDA strategy, we applied RPDA functions D in a conventional data augmentation routine. More specifically, we optimized the lower bound $\widetilde{\mathcal{L}}^{(i)}$ in Eq. (6) over a minibatch of data augmented with D, $\mathbf{X}'^M = \{\, (\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) \mid i \in [1..M]\,\}$, which leads to optimizing with the following lower bound estimator:

$$\widetilde{\mathcal{L}}_{\text{DA}}^{(i)} = \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\,\mathbf{b}'^{(i)} \mid \mathbf{a}'^{(i)}, \mathbf{z}^{(i,l)}\,) + \log p_\theta(\,\mathbf{z}^{(i,l)}\,) - \log q_\phi(\,\mathbf{z}^{(i,l)} \mid \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}\,), \tag{17}$$

$$\begin{aligned} \text{where} \quad \mathbf{z}^{(i,l)} &= g(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}, \phi), \quad \boldsymbol{\epsilon}^{(i,l)} \sim p(\,\boldsymbol{\epsilon}\,), \\ (\mathbf{a}'^{(i)}, \mathbf{b}'^{(i)}) &= d(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}; r^{(i)}), \quad r^{(i)} \sim \mathcal{U}(R). \end{aligned}$$

With the rest of the training setup remained unchanged (see Appendix A), we trained VRL on $\mathbf{X}$ with maximizing $\widetilde{\mathcal{L}}_{\text{DA}}^{(i)}$ and used the trained model to infer the relational property of a hold-out dataset; the inference result is shown in Fig. 16b. Comparing the results from both ablation studies, shown in

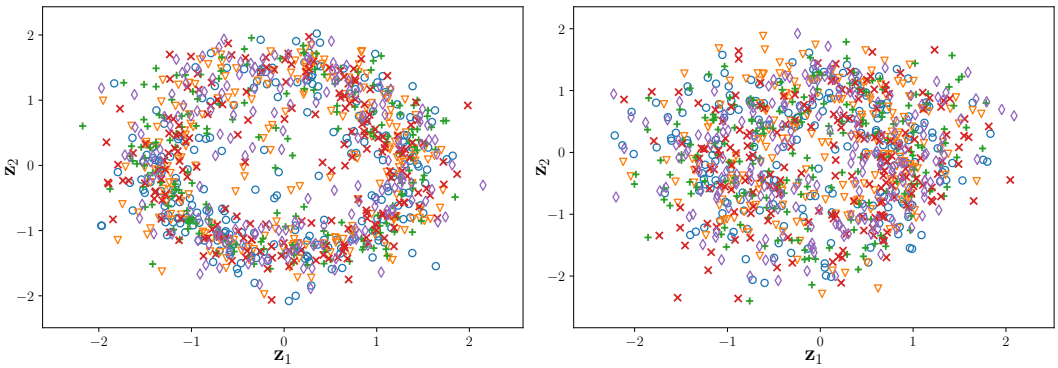


Figure 16: Scatter plots of the relational properties (showing only the mean $\boldsymbol{\mu}$) generated from VRL ablation studies (relationship labels: $\bigcirc : 0°$, $\triangledown : 72°$, $+ : 144°$, $\times : 216°$, $\diamondsuit : 288°$): (a) relational property inferred by a VRL model that was trained without RPDA; (b) relational property inferred by a VRL model that was trained with applying RPDA functions D in a conventional data augmentation routine.

Fig. 16, with inference result from VRL training *with* RPDA, shown in Fig. 4a, we can make the following observations: first, RPDA is a critical component necessary for VRL to learn a meaningful and decoupled relational property, especially when flexible function approximations such as deep neural networks are used; second, by looking at the difference between $\widetilde{\mathcal{L}}_{\text{DA}}^{(i)}$ and $\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)}$ (observe that $\widetilde{\mathcal{L}}_{\text{DA}}^{(i)}$ applies RPDA functions D to every term in Eq. (17), whereas $\widetilde{\mathcal{L}}_{\text{RPDA}}^{(i)}$ applies D only to selected terms in Eq. (16) according to the RPDA strategy), we can draw the conclusion that the improvements brought by RPDA (and its key innovation) comes not from *what* data augmentation functions are applied but *how* they are applied.