Fork or Fail: Cycle-Consistent Training with Many-to-One Mappings

Qipeng Guo[1,2] **Zhijing Jin**[2,3] **Ziyu Wang**[4] **Xipeng Qiu**[1]
Weinan Zhang[5] **Jun Zhu**[4] **Zheng Zhang**[2] **David Wipf**[2]

[1]Fudan University [2]Amazon Shanghai AI Lab [3]Max Planck Institute for Intelligent Systems, Tübingen
[4]Tsinghua University [5]Shanghai Jiao Tong University

Abstract

Cycle-consistent training is widely used for jointly learning a forward and inverse mapping between two domains of interest without the cumbersome requirement of collecting matched pairs within each domain. In this regard, the implicit assumption is that there exists (at least approximately) a ground-truth bijection such that a given input from either domain can be accurately reconstructed from successive application of the respective mappings. But in many applications no such bijection can be expected to exist and large reconstruction errors can compromise the success of cycle-consistent training. As one important instance of this limitation, we consider practically-relevant situations where there exists a many-to-one or surjective mapping between domains. To address this regime, we develop a conditional variational autoencoder (CVAE) approach that can be viewed as converting surjective mappings to implicit bijections whereby reconstruction errors in both directions can be minimized, and as a natural byproduct, realistic output diversity can be obtained in the one-to-many direction. As theoretical motivation, we analyze a simplified scenario whereby minima of the proposed CVAE-based energy function align with the recovery of ground-truth surjective mappings. On the empirical side, we consider a synthetic image dataset with known ground-truth, as well as a real-world application involving natural language generation from knowledge graphs and vice versa, a prototypical surjective case. For the latter, our CVAE pipeline can capture such many-to-one mappings during cycle training while promoting textural diversity for graph-to-text tasks.[1]

1 Introduction

Given data $\boldsymbol{x} \in \mathcal{X}$ from domain $\mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ from domain $\mathcal{Y}$, it is often desirable to learn bidirectional mappings $f : \mathcal{Y} \to \mathcal{X}$ and $g : \mathcal{X} \to \mathcal{Y}$ such that for matched pairs $\{\boldsymbol{x}, \boldsymbol{y}\}$, we have that $\boldsymbol{x} \approx \hat{\boldsymbol{x}} \triangleq f(\boldsymbol{y})$ and $\boldsymbol{y} \approx \hat{\boldsymbol{y}} \triangleq g(\boldsymbol{x})$. When provided with a corpus of suitably aligned data, this amounts to a straightforward supervised learning problem. However, in many applications spanning computer vision (Zhu et al., 2017a), natural language processing (Lample et al., 2018a; Artetxe et al., 2018) and speech recognition (Hori et al., 2019), we may only have access to individual samples from $\mathcal{X}$ and $\mathcal{Y}$ but limited or no labeled ground-truth matches between domains, since, for example, the labeling process may be prohibitively expensive. To address this commonly-encountered situation, cycle-consistent training represents an unsupervised means of jointly learning f and g by penalizing the cycle-consistency reconstruction losses $\|\boldsymbol{x} - f[g(\boldsymbol{x})]\|$ and $\|\boldsymbol{y} - g[f(\boldsymbol{y})]\|$ using non-parallel samples from $\mathcal{X}$ and $\mathcal{Y}$ and some norm or distance metric $\|\cdot\|$ (Zhu et al., 2017a).

However, this process implicitly assumes that there exists a suitable bijection between domains (implying $f = g^{-1}$ and $g = f^{-1}$), an assumption that frequently does not hold for practical applications of cycle-consistent training. As a representative example related to natural language understanding, each $\boldsymbol{x}$ may represent a text segment while $\boldsymbol{y}$ corresponds with the underlying knowledge graph describing the text content. The relationship between these domains is surjective, but not bijective, in the sense that multiple

Preliminary work.

[1]Our code is available `https://github.com/QipengGuo/CycleGT`.

sentences with equivalent meaning but different syntactic structure can be mapped to the same knowledge graph. Hence if we follow any possible learned mapping $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$, there will often be significant error between $\boldsymbol{x}$ and the reconstructed $\hat{\boldsymbol{x}}$. In other words, no invertible transformation exists between domains and there will necessarily be information about $\boldsymbol{x}$ that is lost when we map through $\mathcal{Y}$ space. Additionally, deterministic mappings do not reflect the ground-truth conditional distribution $p_{gt}(\boldsymbol{x}|\boldsymbol{y})$, which is necessary for the generation of diverse text consistent with a given knowledge graph.

Despite these limitations, there has been relatively little effort or rigorous analysis devoted to explicitly addressing the lack of a bijection in applications of cycle-consistent training; Section 2 on related work will discuss this point in greater detail. As a step towards filling this void, in Section 3 we will consider replacing the typical deterministic cycle training pipeline with a stochastic model reflecting $p_{gt}(\boldsymbol{x}|\boldsymbol{y})$ and $p_{gt}(\boldsymbol{y}|\boldsymbol{x})$ for the $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ and $\boldsymbol{y} \rightarrow \hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}}$ cycles respectively. In doing so, we apply a conditional variational autoencoder (CVAE) formulation (Doersch, 2016; Sohn et al., 2015) to deal with the intractable integrals that arise. Note that although the proposed CVAE methodology can be generalized, we will herein restrict ourselves to situations where there exists a many-to-one mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ (i.e., a surjection) as originally motivated by our interest in conversions between knowledge graphs and diverse, natural language text.

Proceeding further, Section 4 provides theoretical support by analyzing a simplified scenario whereby minima of the proposed CVAE-based energy function align with the recovery of ground-truth surjective mappings. To the best of our knowledge, this is the only demonstration of a cycle-consistent model with any type of performance guarantee within a non-trivial, non-bijective context. We then turn to empirical validation in Section 5 that corroborates our theory via a synthetic image example and demonstrates real-world practicality on an application involving the conversion between diverse natural language and knowledge graphs taken from the WebNLG dataset. Overall, experimental results indicate that our proposed CVAE pipeline can approximate surjective mappings during cycle training, with performance on par with supervised alternatives, while promoting diversity for the graph-to-text direction.

2 Related Work

General Cycle-Consistent Training The concept of leveraging the transitivity of two functions that serve as inverses to one another has been applied to a variety of tasks. For example, in computer vision, forward-backward consistency has been used extensively in computer vision (Kalal et al., 2010; Sundaram et al., 2010), and cycle-consistent training pipelines underlie image style transfer (Zhu et al., 2017a; Liu et al., 2017), depth estimation (Godard et al., 2017), and unsupervised domain adaptation (Hoffman et al., 2018) pipelines. Turning to natural language processing (NLP), back translation (Sennrich et al., 2016; Edunov et al., 2018; Jin et al., 2020) and dual learning (Cheng et al., 2016; He et al., 2016) have been widely deployed for unsupervised machine translation. Similar techniques have also contributed to applications such as language style transfer (Shen et al., 2017; Jin et al., 2019). However, the above models primarily rely on deterministic pipelines that implicitly assume a bijection even if one does not actually exist.

And finally, a VAE-inspired model for converting between knowledge graphs and text is considered in (Tseng et al., 2020). But again there is no explicit accounting for non-bijective data as a shared latent space is assumed to contain all information from *both* $\boldsymbol{x}$ and $\boldsymbol{y}$ domains, and the proposed model is designed and tested for semi-supervised learning (not fully unsupervised cycle-consistency). Moreover, for tractable inference, some terms from the variational bound on the log-likelihood (central to all VAE models) are heuristically removed; hence the relationship with the original, motivating probabilistic model remains unclear.

Non-Bijective Mappings Non-bijective mappings are investigated in applications such as multi-domain image-to-image translation (Choi et al., 2018), voice conversion (Kameoka et al., 2018), multi-attribute text style transfer (Lample et al., 2018b), music transfer (Bitton et al., 2018), and multi-modal generation (Shi et al., 2019). Most of this work uses adversarial neural networks, or separate decoders (Lee et al., 2019; Mor et al., 2018), and one case even applies a CVAE model (Jha et al., 2018). However, all the above assume multiple *pre-defined* style domains and require data be clearly separated a priori according to these domains to train non-bijective mappings. In contrast, our proposed model assumes a completely arbitrary surjective mapping that can be learned from the data without such additional domain-specific side information pertaining to styles or related (so ours can fit unknown styles mixed within an arbitrary dataset). One exception is (Zhu et al., 2017b), which handles general non-bijective image mappings using a hybrid VAE-GAN model but unlike our approach, it requires matched $\{\boldsymbol{x}, \boldsymbol{y}\}$ pairs for training.

3 Model Development

We will first present a stochastic alternative to deterministic cycle-consistency that, while useful in principle for handling surjective (but explicitly non-bijective)

mappings, affords us with no practically-realizable inference procedure. To mitigate this shortcoming, we then derive a tractable CVAE approximation and discuss some of its advantages. Later in Section 4 we will analyze the local and global minima of this cycle-consistent CVAE model in the special case where the decoder functions are restricted to being affine.

3.1 Stochastic Cycle-Consistent Formulation

Although the proposed methodology can be generalized, we will herein restrict ourselves to situations where there exists a many-to-one mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ (i.e., a surjection) and the resulting asymmetry necessitates that the $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ and $\boldsymbol{y} \rightarrow \hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}}$ cycles be handled differently. In this regard, our starting point is to postulate an additional latent variable $\boldsymbol{u} \in \mathcal{U}$ that contributes to a surjective matched pair $\{\boldsymbol{x}, \boldsymbol{y}\}$ via

$$\boldsymbol{x} = h_{gt}\left(\boldsymbol{y}, \boldsymbol{u}\right) \quad \text{and} \quad \boldsymbol{y} = h_{gt}^{+}\left(\boldsymbol{x}\right), \tag{1}$$

where $h_{gt} : \mathcal{Y} \times \mathcal{U} \rightarrow \mathcal{X}$ and $h_{gt}^{+} : \mathcal{X} \rightarrow \mathcal{Y}$ represent ground-truth mappings we would ultimately like to estimate. For this purpose we adopt the approximations $h_\theta : \mathcal{Y} \times \mathcal{Z} \rightarrow \mathcal{X}$ and $h_\theta^{+} : \mathcal{X} \rightarrow \mathcal{Y}$ with trainable parameters θ, noting that the second input argument of h_θ is now $\boldsymbol{z} \in \mathcal{Z}$ instead of $\boldsymbol{u} \in \mathcal{U}$. This is because the latter is unobservable and it is sufficient to learn a mapping that preserves the surjection between $\boldsymbol{x}$ and $\boldsymbol{y}$ without necessarily reproducing the exact same functional form of h_{gt}. For example, if hypothetically $\boldsymbol{u} = \pi(\boldsymbol{z})$ in (1) for some function π, we could redefine h_{gt} as a function of $\boldsymbol{z}$ without actually changing the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$.

We are now prepared to define a negative conditional log-likelihood loss for both cycle directions, averaged over the distributions of $\boldsymbol{y}$ and $\boldsymbol{x}$ respectively. For the simpler $\boldsymbol{y} \rightarrow \hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}}$ cycle, we define

$$\ell_y(\theta) \;\triangleq\; -\int \left(\log \int p_\theta\left(\boldsymbol{y}|\hat{\boldsymbol{x}}\right) p(\boldsymbol{z}) d\boldsymbol{z} \right) \rho_{gt}^{y}(d\boldsymbol{y}), \tag{2}$$

where $\hat{\boldsymbol{x}} = h_\theta\left(\boldsymbol{y}, \boldsymbol{z}\right)$, $p_\theta\left(\boldsymbol{y}|\hat{\boldsymbol{x}}\right)$ is determined by h_θ^{+} and an appropriate domain-specific distribution, and $p(\boldsymbol{z})$ is assumed to be fixed and known (e.g., a standardized Gaussian). Additionally, ρ_{gt}^{y} denotes the ground-truth probability measure associated with $\mathcal{Y}$. Consequently, $\rho_{gt}^{y}(d\boldsymbol{y})$ is the measure assigned to the infinitesimal $d\boldsymbol{y}$, from which it obviously follows that $\int \rho_{gt}^{y}(d\boldsymbol{y}) = 1$. Note that the resulting derivations can apply even if no ground-truth density $p_{gt}(\boldsymbol{y})$ exists, e.g., a counting measure over training samples or discrete domains can be assumed within this representation.

Given that there should be no uncertainty in $\boldsymbol{y}$ when conditioned on $\boldsymbol{x}$, we would ideally like to learn parameters whereby $p_\theta\left(\boldsymbol{y}|\hat{\boldsymbol{x}}\right)$, and therefore $\ell_y(\theta)$, degenerates while reflecting a many-to-one mapping. By this we mean that

$$\boldsymbol{y} \approx \hat{\boldsymbol{y}} = h_\theta^{+}(\hat{\boldsymbol{x}}) = h_\theta^{+}\left(h_\theta\left[\boldsymbol{y}, \boldsymbol{z}\right]\right), \quad \forall \boldsymbol{z} \sim p(\boldsymbol{z}). \tag{3}$$

Hence the $\boldsymbol{y} \rightarrow \hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}}$ cycle only serves to favor (near) perfect reconstructions of $\boldsymbol{y}$ while ignoring any $\boldsymbol{z}$ that can be drawn from $p(\boldsymbol{z})$. The latter stipulation is unique relative to typical deterministic cycle-consistent training, which need not learn to ignore randomness from an additional confounding latent factor.

In contrast, the $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ cycle operates somewhat differently, with the latent $\boldsymbol{z}$ serving a more important, non-degenerate role. Similar to before, we would ideally like to minimize the negative conditional log-likelihood given by

$$\ell_x(\theta) \;\triangleq\; -\int \left(\log \int p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}}, \boldsymbol{z}\right) p(\boldsymbol{z}) d\boldsymbol{z} \right) \rho_{gt}^{x}(d\boldsymbol{x}), \tag{4}$$

where now $\hat{\boldsymbol{y}} = h_\theta^{+}\left(\boldsymbol{x}\right)$, $p_\theta\left(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}\right)$ depends on h_θ, and ρ_{gt}^{x} represents the ground-truth probability measure on $\mathcal{X}$. Here $\hat{\boldsymbol{y}}$ can be viewed as an estimate of all the information pertaining to the unknown paired $\boldsymbol{y}$ as preserved through the mapping h_θ^{+} from $\boldsymbol{x}$ to $\boldsymbol{y}$ space. Moreover, if cycle-consistent training is ultimately successful, both $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ should be independent of $\boldsymbol{z}$, and it should be the case that $\int p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}}, \boldsymbol{z}\right) p(\boldsymbol{z}) d\boldsymbol{z} = p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}}\right) \approx p_{gt}\left(\boldsymbol{x}|\boldsymbol{y}\right)$.

Per these definitions, it is also immediately apparent that $p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}}\right)$ is describing the distribution of $h_\theta\left(\boldsymbol{y}, \boldsymbol{z}\right)$ conditioned on $\boldsymbol{y}$ being fixed to $h_\theta^{+}\left(\boldsymbol{x}\right)$. This can be viewed as a stochastic version of the typical $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ cycle, whereas now the latent $\boldsymbol{z}$ allows us to spread probability mass across all $\boldsymbol{x}$ that are consistent with a given $\hat{\boldsymbol{y}}$. In fact, if we set $\boldsymbol{z} = \boldsymbol{z}'$ to a fixed null value (i.e., change $p(\boldsymbol{z})$ to a Dirac delta function centered at some arbitrary $\boldsymbol{z}'$), we recover this traditional pipeline exactly, with $-\log p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}')$ simply defining the implicit loss function, with limited ability to assign high probability to multiple different values of $\boldsymbol{x}$ for any given $\boldsymbol{y}$.

3.2 CVAE Approximation

While $\ell_y(\theta)$ from Section 3.1 can be efficiently minimized using stochastic sampling from $p(\boldsymbol{z})$ to estimate the required integral, the $\ell_x(\theta)$ term is generally intractable. Note that unlike $\ell_y(\theta)$, directly sampling $\boldsymbol{z}$ is not a viable solution for $\ell_x(\theta)$ since $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{y}}, \boldsymbol{z})$ can be close to zero for nearly all values of $\boldsymbol{z}$, and therefore, a prohibitively large number of samples would be needed to obtain reasonable estimates of the integral. Fortunately though, we can form a trainable upper bound using a CVAE architecture that dramatically improves

sample efficiency (Doersch, 2016; Sohn et al., 2015). Specifically, we define

$$\ell_x(\theta,\phi) \;\triangleq\; \int \Big\{ -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}},\boldsymbol{z}\right)\right] p(\boldsymbol{z}) d\boldsymbol{z} + \mathbb{KL}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right] \Big\} \rho^x_{gt}(d\boldsymbol{x}), \quad (5)$$

where in this context, $p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}},\boldsymbol{z}\right)$ is referred to as a *decoder* distribution while $q_\phi\left(\boldsymbol{z}|\boldsymbol{x}\right)$ represents a trainable *encoder* distribution parameterized by ϕ. And by design of general VAE-based models, we have that $\ell_x(\theta,\phi) \geq \ell_x(\theta)$ for all ϕ (Kingma and Welling, 2014; Rezende et al., 2014). Note that we could also choose to condition $q_\phi\left(\boldsymbol{z}|\boldsymbol{x}\right)$ and $p(\boldsymbol{z})$ on $\hat{\boldsymbol{y}}$, although this is not required to form a valid or maximally-tight bound. In the case of $q_\phi\left(\boldsymbol{z}|\boldsymbol{x}\right)$, $\hat{\boldsymbol{y}}$ is merely a function of $\boldsymbol{x}$ and therefore contains no additional information beyond direct conditioning on $\boldsymbol{x}$ (and the stated bound holds regardless of what we choose for q_ϕ). In contrast, $p(\boldsymbol{z})$ defines the assumed generative model which we are also free to choose; however, conditioning on $\hat{\boldsymbol{y}}$ can be absorbed into $p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}},\boldsymbol{z}\right)$ such that there is no change in the representational capacity of $p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}}\right) = \int p_\theta\left(\boldsymbol{x}|\hat{\boldsymbol{y}},\boldsymbol{z}\right) p(\boldsymbol{z}) d\boldsymbol{z}$.

Given (5) as a surrogate for $\ell_x(\theta)$ and suitable distributional assumptions, the combined cycle-consistent loss

$$\ell_{\text{cycle}}(\theta,\phi) \;=\; \ell_x(\theta,\phi) + \ell_y(\theta) \quad (6)$$

can be minimized over $\{\theta,\phi\}$ using stochastic gradient descent and the reparameterization trick from (Kingma and Welling, 2014; Rezende et al., 2014). We henceforth refer to this formulation as *CycleCVAE*. And as will be discussed in more detail later, additional constraints, regularization factors, or inductive biases can also be included to help ensure identifiability of ground-truth mappings. For example, we may consider penalizing or constraining the divergence between the distributions of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}} = h^+_\theta(\boldsymbol{x})$, both of which can be estimated from unpaired samples from ρ^y_{gt} and ρ^x_{gt} respectively. This is useful for disambiguating the contributions of $\boldsymbol{y}$ and $\boldsymbol{z}$ to $\boldsymbol{x}$ and will be equal to zero (or nearly so) if $h^+_\theta \approx h^+_{gt}$ (more on this in Section 4 below).

3.3 CycleCVAE Inference

Once trained and we have obtained some optimal CycleVAE parameters $\{\theta^*,\phi^*\} \approx \arg\min_{\theta,\phi} \ell_{\text{cycle}}(\theta,\phi)$, we can compute matches for test data in either the $\boldsymbol{x}_{test} \rightarrow \boldsymbol{y}_{test}$ or $\boldsymbol{y}_{test} \rightarrow \boldsymbol{x}_{test}$ direction. For the former, we need only compute $\hat{\boldsymbol{y}}_{test} = h^+_{\theta^*}(\boldsymbol{x}_{test})$ and there is no randomness involved. In contrast, for the other direction (one-to-many) we can effectively draw approximate samples from the posterior distribution $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{y}_{test})$ by first drawing samples $\boldsymbol{z} \sim p(\boldsymbol{z})$ and then computing $\hat{\boldsymbol{x}}_{test} = h_{\theta^*}(\boldsymbol{y}_{test},\boldsymbol{z})$.

3.4 CycleCVAE Advantages in Converting Surjections to Implicit Bijections

Before proceeding to a detailed theoretical analysis of CycleCVAE, it is worth examining a critical yet subtle distinction between CycleCVAE and analogous, deterministic baselines. In particular, given that the $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ cycle will normally introduce reconstruction errors because of the lack of a bijection as discussed previously, we could simply augment traditional deterministic pipelines with a $\boldsymbol{z}$ such that $\boldsymbol{x} \rightarrow \{\hat{\boldsymbol{y}},\boldsymbol{z}\} \rightarrow \hat{\boldsymbol{x}}$ forms a bijection. But there remain (at least) two unresolved problems. First, it is unclear how to choose the dimensionality and distribution of $\boldsymbol{z} \in \mathcal{Z}$ such that we can actually obtain a bijection. For example, if $\dim[\mathcal{Z}]$ is less than the unknown $\dim[\mathcal{U}]$, then the reconstruction error $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|$ can still be large, while conversely, if $\dim[\mathcal{Z}] > \dim[\mathcal{U}]$ there can now exist a many-to-one mapping from $\{\boldsymbol{y},\boldsymbol{z}\}$ to $\boldsymbol{x}$, in which case the superfluous degrees of freedom can interfere with our ability to disambiguate the role $\boldsymbol{y}$ plays in predicting $\boldsymbol{x}$. This issue, combined with the fact that we have no mechanism for choosing a suitable $p(\boldsymbol{z})$, implies that deterministic cycle-consistent training is difficult to instantiate.

In contrast, CycleCVAE can effectively circumvent these issues via two key mechanisms that underpin VAE-based models. First, assume that $\dim[\mathcal{X}] > \dim[\mathcal{Y}]$, meaning that $\mathcal{X}$ represents a higher-dimensional space or manifold relative to $\mathcal{Y}$, consistent with our aforementioned surjective assumption. Then provided we choose $\dim[\mathcal{Z}]$ sufficiently large, e.g., $\dim[\mathcal{Z}] \geq \dim[\mathcal{U}]$, we would ideally prefer that CVAE regularization somehow prune away the superfluous dimensions of $\boldsymbol{z}$ that are not required to produce good reconstructions of $\boldsymbol{x}$.[2] For example, VAE pruning could potentially be instantiated by setting the posterior distribution of unneeded dimensions of the vector $\boldsymbol{z}$ to the prior. By this we mean that if dimension j is not needed, then $q_\phi(z_j|\boldsymbol{x}) = p(z_j)$, uninformative noise that plays no role in improving reconstructions of $\boldsymbol{x}$. This capability has been noted in traditional VAE models (Dai et al., 2018; Dai and Wipf, 2019), but never rigorously analyzed in the context of CVAE extensions or cycle-consistent training. In this regard, the analysis in Section 4 will elucidate special cases whereby CycleCVAE can provably lead to optimal pruning, the first such analysis of CVAE models, cycle-consistent or otherwise. This serves to motivate the proposed pipeline as a vehicle for learning an implicit bijection even without knowing the dimensionality or distribution of data from $\mathcal{U}$, a particularly relevant notion given the difficulty in directly estimating $\dim[\mathcal{U}]$ in practice.

[2] Note that $\dim[\mathcal{X}]$ refers to the intrinsic dimensionality of $\mathcal{X}$, which could be a low-dimensional manifold embedded in a higher-dimensional ambient space; same for $\dim[\mathcal{Y}]$.

Secondly, because the CycleCVAE model is explicitly predicated upon a *known* prior $p(\boldsymbol{z})$ (as opposed to the unknown distribution of $\boldsymbol{u}$), other model components are calibrated accordingly such that there is no need to provide an empirical estimate of an unknown prior. Consequently, there is no barrier to cycle-consistent training or the generation of new $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$.

4 Formal Analysis of Special Cases

To the best of our knowledge, there is essentially no existing analysis of cycle-consistent training in the challenging yet realistic scenarios where a bijection between $\boldsymbol{x}$ and $\boldsymbol{y}$ *cannot* be assumed to hold.[3] In this section we present a simplified case whereby the proposed CycleCVAE objective (with added distributional constraints) is guaranteed to have no bad local minimum in a specific sense to be described shortly. The forthcoming analysis relies on the assumption of a Gaussian CVAE with an affine model for the functions $\{h_\theta, h_\theta^+\}$; however, the conclusions we draw are likely to be loosely emblematic of behavior in broader regimes of interest. While admittedly simplistic, the resulting CVAE objective remains non-convex, with a combinatorial number of distinct local minima. Hence it is still non-trivial to provide any sort of guarantees in terms of associating local minima with 'good' solutions, e.g., solutions that recover the desired latent factors, etc. In fact, prior work has adopted similar affine VAE decoder assumptions, but only in the much simpler case of vanilla VAE models (Dai et al., 2018; Lucas et al., 2019), i.e., no cycle training or conditioning as is our focus herein.

4.1 Affine CycleCVAE Model

For analysis purposes, we consider a CVAE model of continuous data $\boldsymbol{x} \in \mathbb{R}^{r_x}$, $\boldsymbol{y} \in \mathbb{R}^{r_y}$, and $\boldsymbol{z} \in \mathbb{R}^{r_z}$, where r_x, r_y, and r_z are the respective sizes of $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$. We assume $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$ and a typical Gaussian decoder $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{y}}, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, where the mean network satisfies the affine parameterizations

$$\begin{aligned} &\boldsymbol{\mu}_x = h_\theta(\hat{\boldsymbol{y}}, \boldsymbol{z}) = \boldsymbol{W}_x\hat{\boldsymbol{y}} + \boldsymbol{V}_x\boldsymbol{z} + \boldsymbol{b}_x, \\ &\text{with } \; \hat{\boldsymbol{y}} = h_\theta^+(\boldsymbol{x}) = \boldsymbol{W}_y\boldsymbol{x} + \boldsymbol{b}_y. \end{aligned} \tag{7}$$

In this expression, $\{\boldsymbol{W}_x, \boldsymbol{W}_y, \boldsymbol{V}_x, \boldsymbol{b}_x, \boldsymbol{b}_y\}$ represents the set of all weight matrices and bias vectors which define the decoder mean $\boldsymbol{\mu}_x$. And as is often assumed in practical VAE models, we set $\boldsymbol{\Sigma}_x = \gamma\boldsymbol{I}$, where $\gamma > 0$ is a scalar parameter within the parameter set θ. Despite these affine assumptions, the CVAE energy function can still have a combinatorial number of distinct local minima as mentioned previously. However, we will closely examine conditions whereby all these local minima are actually global minima that correspond with the optimal inversion of a non-trivial generative model.

Although we could proceed by allowing the encoder to be arbitrarily complex, when the decoder mean function is forced to be affine and $\boldsymbol{\Sigma}_x = \gamma\boldsymbol{I}$, a Gaussian encoder with affine moments is sufficient to achieve the optimal CVAE cost. Specifically, without any loss of representational capacity, we may choose $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ with $\boldsymbol{\mu}_z = \boldsymbol{W}_z\boldsymbol{x} + \boldsymbol{b}_z$ and a diagonal $\boldsymbol{\Sigma}_z = \text{diag}[\boldsymbol{s}]^2$, where $\boldsymbol{s}$ is an arbitrary parameter vector independent of $\boldsymbol{x}$.[4] Collectively, these specifications lead to the complete parameterization $\theta = \{\boldsymbol{W}_x, \boldsymbol{W}_y, \boldsymbol{V}_x, \boldsymbol{b}_x, \gamma\}$, $\phi = \{\boldsymbol{W}_z, \boldsymbol{b}_z, \boldsymbol{s}\}$, and the CVAE energy given by $\ell_x(\theta, \phi) \equiv$

$$\int \Big\{ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\tfrac{1}{\gamma} \left\| (\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)\boldsymbol{x} - \boldsymbol{V}_x\boldsymbol{z} - \boldsymbol{b}_x \right\|_2^2 \right] \tag{8}$$

$$+ d\log\gamma + \sum_{k=1}^{r_z} \left(s_k^2 - \log s_k^2\right) + \left\|\boldsymbol{W}_z\boldsymbol{x} + \boldsymbol{b}_z\right\|_2^2 \Big\} \rho_{gt}^x(d\boldsymbol{x}),$$

noting that, without loss of generality, we have absorbed a $\boldsymbol{W}_x\boldsymbol{b}_y$ factor into $\boldsymbol{b}_x$.

And finally, for the corresponding $\ell_y(\theta)$ model we specify $p_\theta(\boldsymbol{y}|\hat{\boldsymbol{x}}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ using a shared, cycle-consistent parameterization borrowed from (7). For this purpose, we adopt

$$\boldsymbol{\mu}_y = h_\theta^+(\hat{\boldsymbol{x}}) = \boldsymbol{W}_y\hat{\boldsymbol{x}} + \boldsymbol{b}_y, \;\; \text{with } \hat{\boldsymbol{x}} = \boldsymbol{W}_x\boldsymbol{y} + \boldsymbol{V}_x\boldsymbol{z} + \boldsymbol{b}_x, \tag{9}$$

and $\boldsymbol{\Sigma}_y = \gamma\boldsymbol{I}$. Given these assumptions, we have

$$\ell_y(\theta) \equiv \int \left(\boldsymbol{\epsilon}_y^\top \boldsymbol{\Sigma}_{\epsilon_y}^{-1} \boldsymbol{\epsilon}_y + \log\left|\boldsymbol{\Sigma}_{\epsilon_y}\right| \right) \rho_{gt}^y(d\boldsymbol{y}) \tag{10}$$

excluding irrelevant constants, where $\boldsymbol{\epsilon}_y \triangleq (\boldsymbol{I} - \boldsymbol{W}_y\boldsymbol{W}_x)\boldsymbol{y} - \boldsymbol{b}_y$, $\boldsymbol{\Sigma}_{\epsilon_y} \triangleq \gamma\boldsymbol{I} + \boldsymbol{W}_y\boldsymbol{V}_x\boldsymbol{V}_x^\top\boldsymbol{W}_y^\top$ and again, analogous to before we have absorbed $\boldsymbol{W}_y\boldsymbol{b}_x$ into $\boldsymbol{b}_y$ without loss of generality.

4.2 Properties of Global/Local Minima

As a preliminary thought experiment, we can consider the minimization of $\ell_{\text{cycle}}(\theta, \phi)$, where $\ell_x(\theta, \phi)$ is defined via (8) and $\ell_y(\theta)$ via (10), but no assumptions are placed on the distributions ρ_{gt}^x and ρ_{gt}^y. In this situation, it is obvious that even CycleCVAE global minima, were they obtainable, will not generally recover the ground-truth mappings between paired $\boldsymbol{x} \sim \rho_{gt}^x$ and $\boldsymbol{y} \sim \rho_{gt}^y$; there simply will not generally be sufficient capacity. Furthermore, it can even be shown that under

[3] Note that (Grover et al., 2020) addresses identifiability issues that arise during cycle training, but only in the context of strictly bijective scenarios.

[4] Note also that because $\hat{y} = \boldsymbol{W}_y\boldsymbol{x} + \boldsymbol{b}_y$ is an affine function of $\boldsymbol{x}$, including this factor in the encoder representation is redundant, i.e., it can be absorbed into $\boldsymbol{\mu}_z = \boldsymbol{W}_z\boldsymbol{x} + \boldsymbol{b}_z$ without loss of generality.

quite broad conditions there will exist a combinatorial number of *non-global* local minima, meaning local minimizers that fail to achieve the lowest possible cost.

Hence we now present a narrower scenario with constraints on the ground-truth data to better align with the affine simplification described in Section 4.1. This will allow us to formulate conditions whereby all local minima are actually global minima capable of accurately modeling the ground-truth surjection. To this end, we define the following affine ground-truth model:

Definition 1 (Affine Surjective Model) *We define an affine surjective model whereby all matched* $\{\boldsymbol{x}, \boldsymbol{y}\}$ *pairs satisfy*

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{c} \;\; \textit{and} \;\; \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{e}, \tag{11}$$

with $\boldsymbol{B} \in \text{null}[\boldsymbol{D}]$, $\boldsymbol{D}\boldsymbol{A} = \boldsymbol{I}$, $\boldsymbol{D}\boldsymbol{c} = -\boldsymbol{e}$, $\text{rank}[\boldsymbol{A}] = r_y < r_x$ *and* $\text{rank}[\boldsymbol{B}] \leq r_x - r_y$. *Furthermore, we assume that* $\boldsymbol{y} \sim \rho^y_{gt}$ *and* $\boldsymbol{u} \sim \rho^u_{gt}$ *are uncorrelated, and the measure assigned to the transformed random variable* $\boldsymbol{W}\boldsymbol{y} + \boldsymbol{V}\boldsymbol{u}$ *is equivalent to* ρ^y_{gt} *iff* $\boldsymbol{W} = \boldsymbol{I}$ *and* $\boldsymbol{V} = \boldsymbol{0}$. *We also enforce that* $\boldsymbol{y}$ *and* $\boldsymbol{u}$ *have zero mean and identity covariance, noting that any nonzero mean components can be absorbed into* $\boldsymbol{c}$. *Among other things, the stated conditions of the affine surjective model collectively ensure that the mappings* $\boldsymbol{y} \rightarrow \boldsymbol{x}$ *and* $\boldsymbol{x} \rightarrow \boldsymbol{y}$ *can be mutually satisfied.*

Additionally, for later convenience, we also define $r_c \triangleq \text{rank}\left(\mathbb{E}_{\rho^x_{gt}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right]\right) \leq r_x$. We then have the following:

Proposition 2 *Assume that matched pairs* $\{\boldsymbol{x}, \boldsymbol{y}\}$ *follow the affine surjective model from Definition 1. Then subject to the constraint* $\rho^y_{gt} = \rho^{\hat{y}}_\theta$, *where* $\rho^{\hat{y}}_\theta$ *defines the* θ*-dependent distribution of* $\hat{y}$, *all local minima of the CycleVAE objective* $\ell_{\text{cycle}}(\theta, \phi)$, *with* $\ell_x(\theta, \phi)$ *taken from (8) and* $\ell_y(\theta)$ *from (10), will be global minima in the limit* $\gamma \rightarrow 0$ *assuming* $r_z \geq r_c - r_y$. *Moreover, the globally optimal parameters* $\{\theta^*, \phi^*\}$ *will satisfy*

$$\begin{aligned} \boldsymbol{W}^*_x = \boldsymbol{A}, \;\; \boldsymbol{V}^*_x = \left[\widetilde{\boldsymbol{B}}, \boldsymbol{0}\right]\boldsymbol{P}, \;\; \boldsymbol{b}^*_x = \boldsymbol{c}, \\ \boldsymbol{W}^*_y = \widetilde{\boldsymbol{D}}, \;\; \boldsymbol{b}^*_y = -\widetilde{\boldsymbol{D}}\boldsymbol{c}, \end{aligned} \tag{12}$$

where $\widetilde{\boldsymbol{B}}$ *has* $\text{rank}[\boldsymbol{B}]$ *columns,* $\textit{span}[\widetilde{\boldsymbol{B}}] = \textit{span}[\boldsymbol{B}]$, $\boldsymbol{P}$ *is a permutation matrix, and* $\widetilde{\boldsymbol{D}}$ *satisfies* $\widetilde{\boldsymbol{D}}\boldsymbol{A} = \boldsymbol{I}$ *and* $\boldsymbol{B} \in \textit{null}[\widetilde{\boldsymbol{D}}]$.

Note that in practice, we are free to choose r_z as large as we want, so the requirement that $r_z \geq r_c - r_y$ is not significant. Additionally, the constraint $\rho^y_{gt} = \rho^{\hat{y}}_\theta$ can be instantiated (at least approximately) by including a penalty on the divergence between these two distributions. This is feasible using only unpaired samples of $\boldsymbol{y}$ (for estimating ρ^y_{gt}) and $\boldsymbol{x}$ (for estimating $\rho^{\hat{y}}_\theta$), and most cycle-consistent training pipelines contain some analogous form of penalty on distributional differences between cycles (Lample et al., 2018a).

And finally, if $r_y + \text{rank}[\boldsymbol{B}] = r_x$, then the dual requirements that $\widetilde{\boldsymbol{D}}\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{B} \in \text{null}[\widetilde{\boldsymbol{D}}]$ will ensure that $\widetilde{\boldsymbol{D}} = \boldsymbol{D}$ and $\boldsymbol{b}^*_y = \boldsymbol{e}$. However, even if $\widetilde{\boldsymbol{D}} \neq \boldsymbol{D}$ it is inconsequential for effective recovery of the ground-truth model since any $\boldsymbol{x}$ produced by Definition 1 will nonetheless still map to the correct $\boldsymbol{y}$ when applying $\widetilde{\boldsymbol{D}}$ instead of $\boldsymbol{D}$.

Corollary 3 *Given the same setup as Proposition 2, let* $\{\boldsymbol{W}^*_z, \boldsymbol{b}^*_z\}$ *denote the CVAE encoder parameters of any minimum. Then* $\boldsymbol{W}^*_z = \boldsymbol{P}\begin{bmatrix} \widetilde{\boldsymbol{W}}^*_z \\ \boldsymbol{0} \end{bmatrix}$ *and* $\boldsymbol{b}^*_z = \boldsymbol{P}\begin{bmatrix} \widetilde{\boldsymbol{b}}^*_z \\ \boldsymbol{0} \end{bmatrix}$, *where* $\widetilde{\boldsymbol{W}}^*_z$ *has* $\text{rank}[\boldsymbol{B}]$ *rows,* $\widetilde{\boldsymbol{b}}^*_z \in \mathbb{R}^{\text{rank}[\boldsymbol{B}]}$, *and there exists a bijection between* $\boldsymbol{x}$ *and* $\{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\}$, *with* $\widetilde{\boldsymbol{\mu}}_z \triangleq \widetilde{\boldsymbol{W}}^*_z\boldsymbol{x} + \widetilde{\boldsymbol{b}}^*_z$ *(i.e., the nonzero elements of* $\boldsymbol{\mu}_z$*).*

We will now discuss various take-home messages related to these results.

4.3 Practical Implications

As alluded to in Section 3.4, we will generally not know in advance $p(\boldsymbol{u})$ or even $\dim[\mathcal{U}]$, which reduces to $r_c - r_y$ in the simplified affine case. Hence inducing a bijection may seem problematic on the surface. Fortunately though, Proposition 2 and Corollary 3 indicate that as long as we choose $r_z \geq \dim[\mathcal{U}]$ in our CVAE model, we can nonetheless still learn an *implicit* bijection between $\boldsymbol{x}$ and $\{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\}$, where $\widetilde{\boldsymbol{\mu}}_z$ are the informative (nonzero) dimensions of $\boldsymbol{\mu}_z$ that actually contribute to the reconstruction of $\boldsymbol{x}$. In the affine case, these are the dimensions of $\boldsymbol{z}$ aligned with nonzero columns of $\widetilde{\boldsymbol{B}}$, but in general these dimensions could more loosely refer to the degrees-of-freedom in $\boldsymbol{z}$ that, when altered, lead to changes in $\hat{\boldsymbol{x}}$. The remaining superfluous dimensions of $\boldsymbol{z}$ are set to the uninformative prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$ and subsequently filtered out by the CVAE decoder module parameterized by $h_\theta(\boldsymbol{y}, \boldsymbol{z})$. In this diminutive role, they have no capacity for interfering with any attempts to learn a bijection.

Even so, in non-affine cases it is impossible to guarantee that all local minima correspond with the recovery of h_{gt} and h^+_{gt}, or that these functions are even identifiable. Indeed it is not difficult to produce counterexamples whereby recovery is formally impossible. However, if h_θ and h^+_θ are chosen with inductive biases reasonably well-aligned with their ground-truth counterparts, up to unidentifiable latent transformations of the unobservable $\boldsymbol{u}$, we may expect that an approximate bijection between the true $\boldsymbol{x}$ and $\{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\}$ can nonetheless be

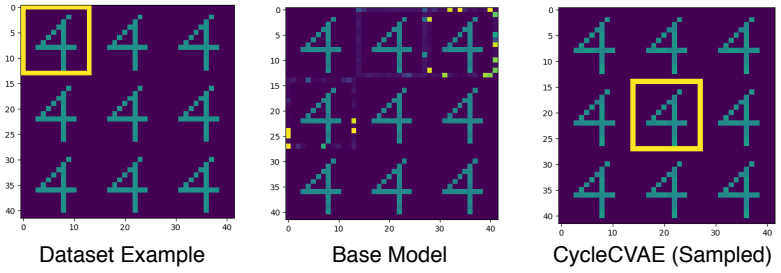

Figure 1: Left: example image from dataset. Middle: image produced by baseline cycle training with $\boldsymbol{y} = 4$. Right: a sample image generated by CycleCVAE conditioned on $\boldsymbol{y} = 4$. For the latter, the position of yellow border is random. In contrast, the base model fails to learn the random border distribution.

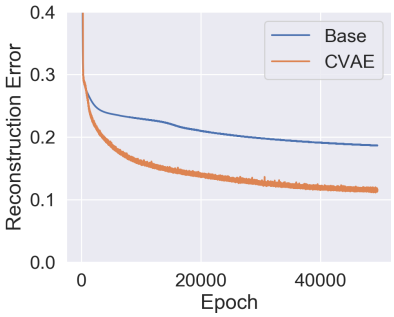

Figure 2: Cycle-consistent reconstruction errors of baseline and CycleCVAE models.

inferred via cycle-consistent training, at least provided we can avoid suboptimal local minimizers that can be introduced by more complex nonlinear decoder models.

5 Experiments

In this section we first consider a synthetic image experiment that supports the theoretical motivation for CycleCVAE. We then turn to practical real-world evaluations involving the conversion between knowledge graphs and text sequences, a core ingredient of natural language understanding/generation and the application that initially motivated our work. We then conclude with an enhanced pipeline involving the recent pretrained T5 model (Raffel et al., 2020).

5.1 Synthetic Image Experiment

We first conduct an experiment designed such that the surjective conditions of Definition 1 in Section 4.2 are loosely satisfied (see supplementary for reasons). As shown in Figure 1 (left panel), each data sample has three components: a digit, its image, and a decorative yellow border. The digit takes value in $\{0, \ldots, 9\}$ and is represented by a 10-dim one-hot vector $\boldsymbol{y}$. The corresponding image $\boldsymbol{x}$ involves 3×3 tiles, and each tile contains the same image of digit $\boldsymbol{y}$. One of the 9 tiles is decorated with a 1-pixel-wide yellow border, and which tile will have this border is determined by $\boldsymbol{u}$, a 9-dim one-hot vector indicating the 9 possible tiles.

We train two models on this dataset, a base model using standard cycle training, and our CycleCVAE that incorporates the proposed CVAE into a baseline cycle model (see supplementary for network description and details). After training, generated samples of the two approaches when presented with the digit '4' are shown in Figure 1 (middle and right panels). The base model fails to learn the yellow border as it cannot handle the one-to-many mapping from digits to images. Meanwhile the random CycleCVAE sample correctly places the border around one of the tiles (different CycleCVAE samples simply move the border to different tiles as desired; see supplementary). Finally, consistent with these generation results, the training curves from Figure 2 reveal that the reconstruction error of the base model, which assumes a bijection, plateaus at a significantly higher value than the CycleCVAE model.

5.2 Knowledge Graph to Text Conversion

We now turn to more challenging real-world experiments involving the surjective mapping between knowledge graphs and text sequences. Here the ideal goal is to generate diverse, natural text from a fixed knowledge graph, or extract the knowledge graph from a piece of text. To this end we compare CycleCVAE against SOTA methods on the widely-used WebNLG graph-to-text dataset (Gardent et al., 2017).

WebNLG Dataset and Test Setup WebNLG data is extracted from DBPedia, where each graph consists of 2–7 nodes and the corresponding text is descriptions of these graphs collected by crowd-sourcing. We follow the preprocessing of (Moryossef et al., 2019) and obtain 13K training, 1.6K validation, and 5K test text-graph pairs. Please see the supplementary for details of the CycleCVAE architecture explicitly designed for handling text and graph data. Note that we did not include any additional penalty function on the divergence between ρ^{y}_{gt} and $\rho^{\hat{y}}_{\theta}$; the architecture inductive biases were sufficient for good performance.

Metrics We measure performance using three metrics: (1) text generation quality with the standard BLEU score (Papineni et al., 2002),[5] (2) graph construction accuracy via the F1 score of the edge predic-

[5]BLEU (%) counts the 1- to 4-gram overlap between the generated sentence and ground truth.

tions among given entity nodes, and (3) text diversity. Text diversity is an increasingly important criterion for NLP because the same meaning can be conveyed in various expressions, and intelligent assistants should master such variations. We evaluate diversity by reporting the number of distinct sentence variations obtained after running the generation model 10 times.

Accuracy Results Since cycle training only requires unsupervised data, we have to break the text-graph pairs to evaluate unsupervised performance. In this regard, there are two ways to process the data. First, we can use 100% of the training data and just shuffle the text and graphs so that the matching/supervision is lost. This is the setting in Table 1, which allows for direct head-to-head comparisons with SOTA supervised methods (assuming no outside training data). The supervised graph-to-text baselines include *Melbourne* (introduced in Gardent et al., 2017), *StrongNeural*, *BestPlan* (Moryossef et al., 2019), *Seg&Align* (Shen et al., 2020), and *G2T* (Koncel-Kedziorski et al., 2019). Supervised text-to-graph models include *OnePass* (Wang et al., 2019a), and *T2G*, a BiLSTM model we implemented. Unsupervised methods include *RuleBased* and *GT-BT* both by (Schmitt et al., 2020). Finally, *CycleBase* is our deterministic cycle training model with the architectural components borrowed from CycleCVAE. Notably, from Table 1 we observe that our model outperforms other unsupervised methods, and it is even competitive with SOTA supervised models in both the graph-to-text (BLEU) and text-to-graph (F1) directions.

	Text(BLEU)	Graph(F1)	#Variations
Supervised (100%)			
Melbourne	45.0	–	1
StrongNeural	46.5	–	1
BestPlan	47.4	–	1
Seg&Align	46.1	–	1
G2T	45.8	–	1
OnePass	–	66.2	–
T2G	–	60.6	–
Unsupervised (100%, Shuffled)			
RuleBased	18.3	0	1
GT-BT	37.7	39.1	1
CycleBase (Ours)	46.2	61.2	1
CycleCVAE (Ours)	46.5	62.6	4.67

Table 1: Performance on the full WebNLG dataset.

	Text(BLEU)	Graph(F1)	#Variations
Supervised (50%)			
G2T	44.5	–	1
T2G	–	59.7	–
Unsupervised (first 50% text, last 50% graph)			
CycleBase (Ours)	43.1	59.8	1
CycleCVAE (Ours)	43.3	60.0	4.01

Table 2: Performance on WebNLG with 50% data.

In contrast, a second, stricter unsupervised protocol involves splitting the dataset into two halves, extracting text from the first half, and graphs from the second half. This is the setting in Table 2, which avoids the possibility of seeing any overlapping entities during training. Although performance is slightly worse given less training data, the basic trends are the same.

Diversity Results From Tables 1 and 2 we also note that CycleCVAE can generate on average more than 4 different sentence types for a given knowledge graph; all other SOTA methods can only generate a single sentence per graph. Additionally, we have calculated that CycleCVAE generates more than two textual paraphrases for 99% of test instances, and the average edit distance between two paraphrases is 12.24 words (see supplementary). Moreover, CycleCVAE text diversity does not harm fluency and semantic relevance as the BLEU score is competitive with SOTA methods as mentioned previously.

Diverse Text Output Generated by CycleCVAE
– The population density of Arlington, Texas is 1472.0. – Arlington, Texas has a population density of 1472.0.
– Alan Bean, who was born in Wheeler, Texas, is now "retired." – Alan Bean is a United States citizen who was born in Wheeler, Texas. He is now "retired."

Table 3: Every two variations are generated by CycleCVAE from the same knowledge graph.

We list text examples generated by our model in Table 3, with more in the supplementary. The diverse generation is a significant advantage for many real applications. For example, it can make automated conversations less boring and simulate different scenarios. And diversity can push model generated samples closer to the real data distribution because there exist different ways to verbalize the same knowledge graph (although diversity will not in general improve BLEU scores, and can sometimes actually lower them).

5.3 Integrating CycleCVAE with T5

Previous graph-text results are all predicated on no outside training data beyond WebNLG. However, we now consider an alternative testing scenario whereby outside training data can be incorporated by integrating CycleCVAE with a large pretrained T5 sequence-to-sequence model (Raffel et al., 2020). Such models have revolutionized many NLP tasks and can potentially improve the quality of the graph-to-text direction in cycle training on WebNLG. To this end, we trained a CycleCVAE model, with the function $h_\theta(\boldsymbol{y}, \boldsymbol{z})$ formed from a pretrained T5 architecture (see supplementary for details). Results are shown in Table 4, where un-

supervised CycleCVAE+T5 produces a competitive BLEU score relative to fully supervised T5 baselines. It also maintains diversity of generated text sequences.

Model	BLEU	#Vars.
Supervised T5 (Kale, 2020)	57.1	1
Supervised T5 (Ribeiro et al., 2020)	57.4	1
Supervised T5 (Our Impl.)	56.4	1
Unsupervised CycleCVAE+T5	55.7	3.84

Table 4: Text generation results with T5 on WebNLG.

6 Conclusion

We have proposed CycleCVAE for explicitly handling non-bijective surjections commonly encountered in real-world applications of unsupervised cycle-consistent training. Our framework has both a solid theoretical foundation and strong empirical performance on practical knowledge graph-to-text conversion problems. For future work we can consider extending CycleCVAE to handle many-to-many (non-bijective, non-surjective) mappings, or unsolved applications such as conversions between scene graphs and realistic images (which remains extremely difficult even with supervision).

References

Artetxe, Mikel, Labaka, Gorka, Agirre, Eneko, and Cho, Kyunghyun (2018). "Unsupervised Neural Machine Translation". *6th International Conference on Learning Representations, ICLR 2018*.

Bitton, Adrien, Esling, Philippe, and Chemla-Romeu-Santos, Axel (2018). "Modulated variational auto-encoders for many-to-many musical timbre transfer". *arXiv preprint arXiv:1810.00222*.

Cheng, Yong, Xu, Wei, He, Zhongjun, He, Wei, Wu, Hua, Sun, Maosong, and Liu, Yang (2016). "Semi-Supervised Learning for Neural Machine Translation". *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Choi, Yunjey, Choi, Minje, Kim, Munyoung, Ha, Jung-Woo, Kim, Sunghun, and Choo, Jaegul (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dai, Bin, Wang, Yu, Aston, John, Hua, Gang, and Wipf, David (2018). "Connections with Robust PCA and the Role of Emergent Sparsity in Variational Autoencoder Models". *Journal of Machine Learning Research*.

– (2019). "Hidden Talents of the Variational Autoencoder". *arXiv preprint arXiv:1706.05148*.

Dai, Bin and Wipf, David (2019). "Diagnosing and Enhancing VAE Models". *International Conference on Learning Representations*.

Doersch, Carl (2016). "Tutorial on variational autoencoders". *arXiv preprint arXiv:1606.05908*.

Edunov, Sergey, Ott, Myle, Auli, Michael, and Grangier, David (2018). "Understanding Back-Translation at Scale". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Gardent, Claire, Shimorina, Anastasia, Narayan, Shashi, and Perez-Beltrachini, Laura (2017). "The WebNLG Challenge: Generating Text from RDF Data". *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017*.

Godard, Clément, Mac Aodha, Oisin, and Brostow, Gabriel J. (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

Grover, Aditya, Chute, Christopher, Shu, Rui, Cao, Zhangjie, and Ermon, Stefano (2020). "AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows." *AAAI Conference on Artificial Intelligence*.

He, Di, Xia, Yingce, Qin, Tao, Wang, Liwei, Yu, Nenghai, Liu, Tie-Yan, and Ma, Wei-Ying (2016). "Dual Learning for Machine Translation". *Annual Conference on Neural Information Processing Systems 2016*.

Hoffman, Judy, Tzeng, Eric, Park, Taesung, Zhu, Jun-Yan, Isola, Phillip, Saenko, Kate, Efros, Alexei A., and Darrell, Trevor (2018). "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Hori, Takaaki, Astudillo, Ramón Fernández, Hayashi, Tomoki, Zhang, Yu, Watanabe, Shinji, and Roux, Jonathan Le (2019). "Cycle-consistency Training for End-to-end Speech Recognition". *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*.

Jha, Ananya Harsh, Anand, Saket, Singh, Maneesh, and Veeravasarapu, V. S. R. (2018). "Disentangling Factors of Variation with Cycle-Consistent Variational Auto-encoders". *Computer Vision - ECCV 2018 - 15th European Conference*.

Jin, Di, Jin, Zhijing, Zhou, Joey Tianyi, and Szolovits, Peter (2020). "A Simple Baseline to Semi-Supervised Domain Adaptation for Machine Translation". *CoRR*.

Jin, Zhijing, Jin, Di, Mueller, Jonas, Matthews, Nicholas, and Santus, Enrico (2019). "IMaT: Unsupervised Text Attribute Transfer via Iterative Matching and Translation". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*.

Kalal, Zdenek, Mikolajczyk, Krystian, and Matas, Jiri (2010). "Forward-Backward Error: Automatic Detection of Tracking Failures". *20th International Conference on Pattern Recognition, ICPR 2010*.

Kale, Mihir (2020). "Text-to-Text Pre-Training for Data-to-Text Tasks". *arXiv preprint arXiv:2005.10433*.

Kameoka, Hirokazu, Kaneko, Takuhiro, Tanaka, Kou, and Hojo, Nobukatsu (2018). "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks". *CoRR*.

Kingma, Diederik P. and Welling, Max (2014). "Auto-Encoding Variational Bayes". *2nd International Conference on Learning Representations, ICLR 2014*.

Koncel-Kedziorski, Rik, Bekal, Dhanush, Luan, Yi, Lapata, Mirella, and Hajishirzi, Hannaneh (2019). "Text Generation from Knowledge Graphs with Graph Transformers". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*.

Lample, Guillaume, Conneau, Alexis, Denoyer, Ludovic, and Ranzato, Marc'Aurelio (2018a). "Unsupervised Machine Translation Using Monolingual Corpora Only". *6th International Conference on Learning Representations, ICLR 2018*.

Lample, Guillaume, Subramanian, Sandeep, Smith, Eric, Denoyer, Ludovic, Ranzato, Marc'Aurelio, and Boureau, Y-Lan (2018b). "Multiple-attribute text rewriting". *International Conference on Learning Representations*.

Lee, Keonnyeong, Yoo, In-Chul, and Yook, Dongsuk (2019). "Many-to-Many Voice Conversion using Cycle-Consistent Variational Autoencoder with Multiple Decoders". *arXiv*.

Liu, Ming-Yu, Breuel, Thomas, and Kautz, Jan (2017). "Unsupervised image-to-image translation networks". *Advances in neural information processing systems*.

Lucas, James, Tucker, George, Grosse, Roger B, and Norouzi, Mohammad (2019). "Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse". *Advances in Neural Information Processing Systems*.

Mor, Noam, Wolf, Lior, Polyak, Adam, and Taigman, Yaniv (2018). "A universal music translation network". *arXiv preprint arXiv:1805.07848*.

Moryossef, Amit, Goldberg, Yoav, and Dagan, Ido (2019). "Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing (2002). "Bleu: a Method for Automatic Evaluation of Machine Translation". *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*.

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". *Journal of Machine Learning Research* 140.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". *International Conference on Machine Learning*.

Ribeiro, Leonardo FR, Schmitt, Martin, Schütze, Hinrich, and Gurevych, Iryna (2020). "Investigating Pre-trained Language Models for Graph-to-Text Generation". *arXiv preprint arXiv:2007.08426*.

Schmitt, Martin, Sharifzadeh, Sahand, Tresp, Volker, and Schütze, Hinrich (2020). "An Unsupervised Joint System for Text Generation from Knowledge Graphs and Semantic Parsing". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Sennrich, Rico, Haddow, Barry, and Birch, Alexandra (2016). "Improving Neural Machine Translation Models with Monolingual Data". *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Shen, Tianxiao, Lei, Tao, Barzilay, Regina, and Jaakkola, Tommi S. (2017). "Style Transfer from Non-Parallel Text by Cross-Alignment". *Advances in neural information processing systems*.

Shen, Xiaoyu, Chang, Ernie, Su, Hui, Niu, Cheng, and Klakow, Dietrich (2020). "Neural Data-to-Text Generation via Jointly Learning the Segmentation and Correspondence". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*.

Shi, Yuge, Narayanaswamy, Siddharth, Paige, Brooks, and Torr, Philip H. S. (2019). "Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models". *Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*.

Sohn, Kihyuk, Lee, Honglak, and Yan, Xinchen (2015). "Learning structured output representation using deep conditional generative models". *Advances in neural information processing systems*.

Sundaram, Narayanan, Brox, Thomas, and Keutzer, Kurt (2010). "Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow". *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision*.

Tseng, Bo-Hsiang, Cheng, Jianpeng, Fang, Yimai, and Vandyke, David (2020). "A Generative Model for Joint Natural Language Understanding and Generation". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*.

Walker, Christopher, Strassel, Stephanie, Medero, Julie, and Maeda, Kazuaki (2006). "ACE 2005 multilingual training corpus". *Linguistic Data Consortium, Philadelphia*.

Wang, Haoyu, Tan, Ming, Yu, Mo, Chang, Shiyu, Wang, Dakuo, Xu, Kun, Guo, Xiaoxiao, and Potdar, Saloni (2019a). "Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers". *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*.

Wang, Minjie, Yu, Lingfan, Zheng, Da, Gan, Quan, Gai, Yu, Ye, Zihao, Li, Mufei, Zhou, Jinjing, Huang, Qi, Ma, Chao, Huang, Ziyue, Guo, Qipeng, Zhang, Hao, Lin, Haibin, Zhao, Junbo, Li, Jinyang, Smola, Alexander J., and Zhang, Zheng (2019b). "Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs". *CoRR*.

Wipf, David and Nagarajan, Srikantan (2007). "Beamforming Using the Relevance Vector Machine". *International Conference on Machine Learning*.

Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. (2017a). "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". *IEEE International Conference on Computer Vision, ICCV 2017*.

Zhu, Jun-Yan, Zhang, Richard, Pathak, Deepak, Darrell, Trevor, Efros, Alexei A., Wang, Oliver, and Shechtman, Eli (2017b). "Toward Multimodal Image-to-Image Translation". *Annual Conference on Neural Information Processing Systems*.

Supplementary Materials

The supplementary file includes additional content related to the following:

1. Synthetic Experiments (Section 5.1 in main paper): We explain why the synthetic data loosely align with Definition 1, describe the network architectures of the CycleBase and CycleCVAE models, and include additional generation results.
2. WebNLG Experiments (Section 5.2 in main paper): We describe the experimental setup for WebNLG, including the task description, cycle-consistency model design, and all baseline and implementation details. We also include an ablation study varying dim[$\boldsymbol{z}$].
3. T5 Extension (Section 5.3 in main paper): We provide details of the CycleCVAE+T5 extension and include additional generated samples showing textual diversity.
4. Proof of Proposition 2.
5. Proof of Corollary 3.

7 Synthetic Dataset Experimental Details and Additional Results

7.1 Dataset Description and Relation to Definition 1

To motivate how the synthetic data used in Section 5.1 from the main paper at least partially align with Definition 1, we let $\boldsymbol{c}$ and $\boldsymbol{e}$ be zero vectors and $\boldsymbol{A} \in \mathbb{R}^{d \times 10}$ be a $d \times 10$ transformation matrix from images to digits, where d is the total number of pixels in each image $\boldsymbol{x}$. In other words, each column $i \in \{0, 1, \ldots, 9\}$ of $\boldsymbol{A}$ is a linearized pixel sequence of the 2D image of digit i from top left to bottom right. Based on $\boldsymbol{A}$, we construct an example inverse matrix $\boldsymbol{D}$ so that $\boldsymbol{DA} = \boldsymbol{I}$. Specifically, $\boldsymbol{D}$ can be a $10 \times d$ matrix where each row $i \in \{0, 1, \ldots, 9\}$ is a linearized pixel sequence of a masked version of the image of the digit i, and this image can have, for example, only one non-zero pixel that is sufficient to distinguish the digit i from all other nine possibilities. We also construct $\boldsymbol{B}$, a $d \times 9$ transformation matrix from the image to the border position, which surrounds one out of the nine tiles in each image. Each column $i \in \{0, 1, \ldots, 8\}$ of $\boldsymbol{B}$ is a linearized pixel sequence of the 2D image of the border surrounding the i-th tile. Since the patterns of the digit and border do not share any non-zero pixels, we should have that $\boldsymbol{DB} = \boldsymbol{0}$. Moreover, each digit's image is distinct and cannot be produced by combining other digit images, so rank[$\boldsymbol{A}$] $= r_y$ and also $r_y \leq r_x$ because border patterns are orthogonal to digit patterns. Hence, we also have rank[$\boldsymbol{B}$] $\leq r_x - r_y$. Note however that the synthetic data do not guarantee that $\boldsymbol{Wy} + \boldsymbol{Vu}$ is equivalent to ρ^y_{gt} iff $\boldsymbol{W} = \boldsymbol{I}$ and $\boldsymbol{V} = \boldsymbol{0}$.

7.2 Network Architectures

We train two models on this dataset, a base model CycleBase using standard cycle training, and our CycleCVAE that incorporates the proposed CVAE into a baseline cycle model.

CycleBase The base model uses multilayer perceptrons (MLPs) for both the image($\boldsymbol{x}$)-to-digit($\boldsymbol{y}$) mapping $h^+_\theta(\boldsymbol{x})$ (shared with CycleCVAE), and the digit($\boldsymbol{y}$)-to-image($\boldsymbol{x}$) mapping denoted $h^{\text{Base}}_\theta(\boldsymbol{y})$. Each MLP hidden layer (two total) has 50 units with the tanh activation function. The last layer of $h^+_\theta(\boldsymbol{x})$ uses a softmax function to output a vector of probabilities $\boldsymbol{\alpha}$ over the ten digits, and therefore we can apply $p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \text{Cat}(\boldsymbol{y}|\boldsymbol{\alpha})$, a categorical distribution conditioned on $\boldsymbol{\alpha}$, for training purposes. The last layer of digit-to-image $h^{\text{Base}}_\theta(\boldsymbol{y})$ adopts a per-pixel sigmoid function (since the value of each pixel is between 0 and 1), and we assume $p_\theta(\boldsymbol{x}|\boldsymbol{y})$ is based on the binary cross entropy loss.

CycleCVAE Our CycleCVAE uses the same function $h_\theta^+(\boldsymbol{x})$ as the base model. However, for the digit-to-image generation direction, CycleCVAE includes a 1-dimensional latent variable $\boldsymbol{z}$ sampled from $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, where $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are both learned by 50-dimensional, 3-layer MLPs (including output layer) with input $\boldsymbol{x}$. Then $h_\theta(\boldsymbol{y}, \boldsymbol{z})$ takes the digit $\boldsymbol{y}$ and latent variable $\boldsymbol{z}$ as inputs to another 3-layer MLP with 50 hidden units and the same activation function as the base model.

7.3 Generation Results

In addition to Figure 1 in the main paper, we list more example images generated by our model in the figure below. As we can see, the base model fails to learn the diverse border which should randomly surround only one of the nine tiles. However, CycleCVAE learns the border in its latent variable $\boldsymbol{z}$ and by random sampling, CycleCVAE can generate an arbitrary border around one of the nine digits as expected.

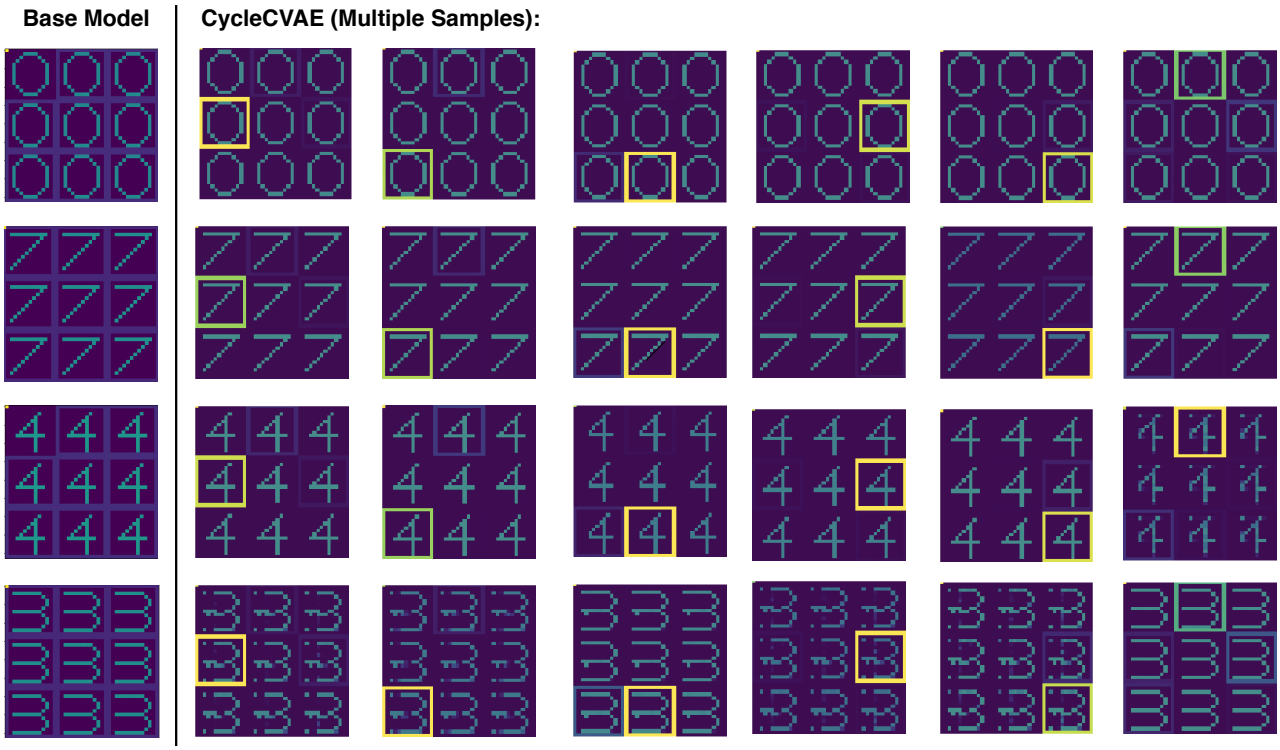

Figure 3: Example images generated by CycleCVAE.

8 WebNLG Experimental Setup and Ablation Study

The WebNLG dataset[6] is widely used for conversions between graph and text. Note that WebNLG is the most appropriate dataset for our purposes because in other candidates (e.g., relation extraction datasets (Walker et al., 2006)) the graphs only contain a very small subset of the information in the text.

8.1 Task Description

The WebNLG experiment includes two directions: text-to-graph (T2G) and graph-to-text (G2T) generation. The G2T task aims to produce descriptive text that verbalizes the graphical data. For example, the knowledge graph triplets "*(Allen Forest, genre, hip hop), (Allen Forest, birth year, 1981)*" can be verbalized as "*Allen Forest, a hip hop musician, was born in 1981.*" This has wide real-world applications, for instance, when a digital assistant needs to translate some structured information (e.g., the properties of the restaurant) to the human user. The other task, T2G is also important, as it extracts structures in the form of knowledge graphs from the text, so that

[6]It can be downloaded from `https://webnlg-challenge.loria.fr/challenge_2017/`.

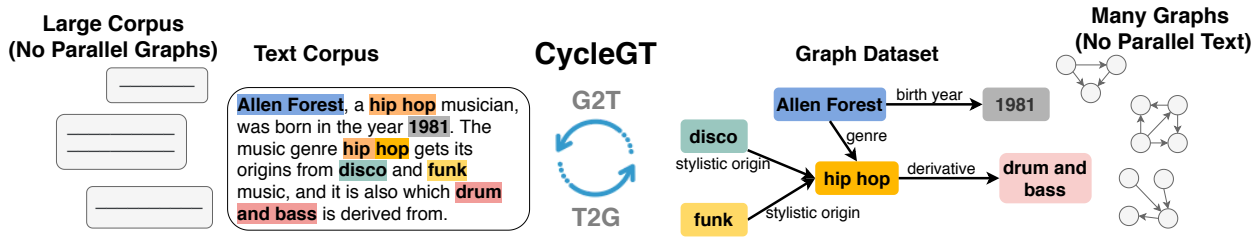

Figure 4: The graph-to-text generation task aims to verbalize a knowledge graph, while the text-to-graph task extracts the information of text into the form of a knowledge graph.

all entities become nodes, and the relationships among entities form edges. It can help many downstream tasks, such as information retrieval and reasoning. The two tasks can be seen as a dual problem, as shown in Figure 4.

Specifically, for unsupervised graph-to-text and text-to-graph generation, we have two non-parallel datasets:

- A text corpus $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ consisting of N text sequences, and
- A graph dataset $\boldsymbol{Y} = \{\boldsymbol{y}_j\}_{j=1}^{M}$ consisting of M graphs.

The constraint is that the graphs and text contain the same distribution of latent content, but are different forms of surface realizations, i.e., there is no alignment providing matched pairs. Our goal is to train two models in an unsupervised manner: h_θ that generates text based on the graph, and h_θ^+ that produces a graph based on text.

8.2 Cycle Training Models

CycleBase Similar to the synthetic experiments mentioned above, we first propose the base cycle training model CycleBase that jointly learns graph-to-text and text-to-graph generation. To be consistent with our main paper, we denote text as $\boldsymbol{x}$ and graphs as $\boldsymbol{y}$, and the graph-to-text generation is a one-to-many mapping. The graph cycle, $\boldsymbol{y} \rightarrow \hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}}$ is as follows: Given a graph $\boldsymbol{y}$, the cycle-consistent training first generates synthetic text $\hat{\boldsymbol{x}} = h_\theta^{\text{Base}}(\boldsymbol{y})$, and then uses it to reconstruct the original graph $\hat{\boldsymbol{y}} = h_\theta^+(\hat{\boldsymbol{x}})$. The loss function is imposed to align the generated graph $\hat{\boldsymbol{y}}$ with the original graph $\boldsymbol{y}$. Similarly, the text cycle, $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$, is to align $\boldsymbol{x}$ and the generated $\hat{\boldsymbol{x}}$. Both loss functions adopt the cross entropy loss.

Specifically, we instantiate the graph-to-text module $h_\theta^{\text{Base}}(\boldsymbol{y})$ with the GAT-LSTM model proposed by (Koncel-Kedziorski et al., 2019), and the text-to-graph module $h_\theta^+(\boldsymbol{x})$ with a simple BiLSTM model we implemented. The GAT-LSTM module has two layers of graph attention networks (GATs) with 512 hidden units, and two layers of a LSTM text decoder with multi-head attention over the graph node embeddings produced by GAT. This attention mechanism uses four attention heads, each with 128 dimensions for self-attention and 128 dimension for cross-attention between the decoder and node features. The BiLSTM for text-to-graph construction uses 2-layer bidirectional LSTMs with 512 hidden units.

CycleCVAE Our CycleCVAE uses the same $h_\theta^+(\boldsymbol{x})$ as the base model. As for $h_\theta(\boldsymbol{y}, \boldsymbol{z})$ (the CycleCVAE extension of CycleBase), we first generate a 10-dimensional latent variable $\boldsymbol{z}$ sampled from $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, where $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are both learned by bidirectional LSTMs plus a fully connected feedforward layer. We form $p(\boldsymbol{z}|\boldsymbol{y})$ as a Gaussian distribution whose mean and variance are learned from a fully connected feedforward layer which takes in the feature of the root node of the GAT to represent the graph. Note that applying this $p(\boldsymbol{z}|\boldsymbol{y})$ as the CycleCVAE prior is functionally equivalent to using a more complicated encoder, as mentioned in the main paper.

Implementation Details For both cycle models, we adopt the Adam optimizer with a learning rate of 5e−5 for the text-to-graph modules, and learning rate of 2e−4 for graph-to-text modules. For the graph-to-text module, we re-implement the GAT-LSTM model (Koncel-Kedziorski et al., 2019) using the DGL library (Wang et al., 2019b). Our code is available `https://github.com/QipengGuo/CycleGT`.

8.3 Details of Competing Methods

Unsupervised Baselines As cycle training models are unsupervised learning methods, we first compare with unsupervised baselines. *RuleBased* is a heuristic baseline proposed by (Schmitt et al., 2020) which simply iterates through the graph and concatenates the text of each triplet. For example, the triplet "(AlanShepard, occupation, TestPilot)" will be verbalized as "Alan Shepard occupation test pilot." If there are multiple triplets, their text expressions will be concatenated by "and." The other baseline, *UMT* (Schmitt et al., 2020), formulates the graph and text conversion as a sequence-to-sequence task and applies a standard unsupervised machine translation (UMT) approach. It serializes each triplet of the graph in the same way as RuleBased, and concatenates the serialization of all triplets in a random order, using special symbols as separators.

Supervised Baselines We also compare with *supervised* systems using the original supervised training data. Since there is no existing work that jointly learns graph-to-text and text-to-graph in a supervised way, we can only use models that address one of the two tasks. For graph-to-text generation, we list the performance of state-of-the-art supervised models including (1) *Melbourne*, the best supervised system submitted to the WebNLG challenge 2017 (Gardent et al., 2017), which uses an encoder-decoder architecture with attention, (2) *StrongNeural* (Moryossef et al., 2019) which improves the common encoder-decoder model, (3) *BestPlan* (Moryossef et al., 2019) which uses a special entity ordering algorithm before neural text generation, (4) *G2T* (Koncel-Kedziorski et al., 2019) which is the same as the GAT-LSTM architecture adopted in our cycle training models, and (5) *Seg&Align* (Shen et al., 2020), which segments the text into small units, and learns the alignment between data and target text segments. The generation process uses the attention mechanism over the corresponding data piece to generate the corresponding text. For text-to-graph generation, we compare with state-of-the-art models including *OnePass* (Wang et al., 2019a), a BERT-based relation extraction model, and *T2G*, the BiLSTM model that we adopt as the text-to-graph component in the cycle training of CycleBase and CycleCVAE.

8.4 Ablation Study

We conduct an ablation study using the 50%:50% unsupervised data of WebNLG. Note that our models do not use an adversarial term, so we only tune the CVAE latent dimension to test robustness to this factor. The hyperparameter tuning of the size of the latent dimension is shown in Table 5, where we observe that our CycleCVAE is robust against different $\boldsymbol{z}$ dimensions. Note that because $\boldsymbol{z}$ is continuous while generated text is discrete, just a single dimension turns out to be adequate for good performance for these experiments. Even so, the encoder variance can be turned up to avoid 'overusing' any continuous latent dimension to roughly maintain a bijection.

	Text (BLEU)	Diversity (# Variations)
Latent Dimension		
$\boldsymbol{z} = 1$	46.3	4.62
$\boldsymbol{z} = 10$	46.5	4.67
$\boldsymbol{z} = 50$	46.2	4.65

Table 5: Text quality (by BLEU scores) and diversity (by the number of variations) under different dimensions of $\boldsymbol{z}$.

9 T5 Model Details and More Generated Samples

9.1 CycleCVAE+T5 Implementational Details

We adopted the pretrained T5 model (Raffel et al., 2020) to replace the GAT-LSTM architecture that we previously used for the graph-to-text module within the cycle training. T5 is a sequence-to-sequence model that takes as input a serialized graph (see the serialization practice in Schmitt et al., 2020; Ribeiro et al., 2020; Kale, 2020) and generates a text sequence accordingly. We finetune the T5 during training with the Adam optimizer using a learning rate of 5e−5.

9.2 Additional Text Diversity Examples

We list the text diversity examples generated by CycleCVAE+T5 in Table 6.

No.	Variations
1	– Batagor, a variation of Siomay and Shumai, can be found in Indonesia, where the leader is Joko Widodo and Peanut sauce is an ingredient. – Batagor is a dish from Indonesia, where the leader is Joko Widodo and the main ingredient is Peanut sauce. It can also be served as a variation of Shumai and Siomay.
2	– The AMC Matador, also known as "American Motors Matador", is a Mid-size car with an AMC V8 engine and is assembled in Thames, New Zealand. – AMC Matador, also known as "American Motors Matador", is a Mid-size car. It is made in Thames, New Zealand and has an AMC V8 engine.
3	– Aleksandr Chumakov was born in Moscow and died in Russia. The leader of Moscow is Sergey Sobyanin. – Aleksandr Chumakov, who was born in Moscow, was a leader in Moscow where Sergey Sobyanin is a leader. He died in Russia.
4	– A Wizard of Mars is written in English language spoken in Great Britain. It was published in the United States, where Barack Obama is the president. – A Wizard of Mars comes from the United States where Barack Obama is the leader and English language spoken in Great Britain.
5	– The Addiction (journal), abbreviated to "Addiction", has the ISSN number "1360-0443" and is part of the academic discipline of Addiction. – Addiction (journal), abbreviated to "Addiction", has the ISSN number "1360-0443".
6	– Atlantic City, New Jersey is part of Atlantic County, New Jersey Atlantic County, New Jersey, in the United States. – Atlantic City, New Jersey is part of Atlantic County, New Jersey, United States.
7	– Albuquerque, New Mexico, United States, is lead by the New Mexico Senate, led by John Sanchez and Asian Americans. – Albuquerque, New Mexico, in the United States, is lead by the New Mexico Senate, where John Sanchez is a leader and Asian Americans are an ethnic group.
8	– Aaron Turner plays the Electric guitar and plays Black metal, Death metal and Black metal. He also plays in the Twilight (band) and Old Man Gloom. – Aaron Turner plays the Electric guitar and plays Black metal. He is associated with the Twilight (band) and Old Man Gloom. He also plays Death metal.

Table 6: Examples of diverse text generated by CycleCVAE based on the same input knowledge graph.

10 Proof of Proposition 2

The high-level proof proceeds in several steps. First we consider optimization of $\ell_x(\theta, \phi)$ over ϕ to show that no suboptimal local minima need be encountered. We then separately consider optimizing $\ell_x(\theta, \phi)$ and $\ell_y(\theta)$ over the subset of θ unique to each respective loss. Next we consider jointly optimizing the remaining parameters residing between both terms. After assimilating the results, we arrive at the stated result of Proposition 2. Note that with some abuse of notation, we reuse several loss function names to simplify the exposition; however, the meaning should be clear from context.

10.1 Optimization over encoder parameters ϕ in $\ell_x(\theta, \phi)$

The energy term from the $\boldsymbol{x} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{x}}$ cycle can be modified as

$$\begin{aligned}
\ell_x(\theta,\phi) &= \int \left\{ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\tfrac{1}{\gamma} \left\| \boldsymbol{x} - \boldsymbol{\mu}_x \right\|_2^2 \right] + d \log \gamma + \sum_{k=1}^{r_z} \left(s_k^2 - \log s_k^2 \right) + \left\| \boldsymbol{\mu}_z \right\|_2^2 \right\} \rho_{gt}^x(d\boldsymbol{x}) \\
&= \int \left\{ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\tfrac{1}{\gamma} \left\| \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} - \boldsymbol{V}_x \boldsymbol{z} - \boldsymbol{W}_x \boldsymbol{b}_y - \boldsymbol{b}_x \right\|_2^2 \right] \right. \\
&\quad \left. + d \log \gamma + \sum_{k=1}^{r_z} \left(s_k^2 - \log s_k^2 \right) + \left\| \boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z \right\|_2^2 \right\} \rho_{gt}^x(d\boldsymbol{x}) \qquad (13) \\
&= \int \left\{ \tfrac{1}{\gamma} \left\| \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} - \boldsymbol{V}_x \left(\boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z \right) - \boldsymbol{W}_x \boldsymbol{b}_y - \boldsymbol{b}_x \right\|_2^2 \right. \\
&\quad \left. + d \log \gamma + \sum_{k=1}^{\kappa} \left(s_k^2 - \log s_k^2 + \tfrac{1}{\gamma} s_k^2 \| \boldsymbol{v}_{x,k} \|_2^2 \right) + \left\| \boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z \right\|_2^2 \right\} \rho_{gt}^x(d\boldsymbol{x}),
\end{aligned}$$

where $\boldsymbol{v}_{x,k}$ denotes the k-th column of $\boldsymbol{V}_x$. Although this expression is non-convex in each s_k^2, by taking derivatives and setting them equal to zero, it is easily shown that there is a single stationary point that operates as the unique minimum. Achieving the optimum requires only that $s_k^2 = \left[\frac{1}{\gamma} \| \boldsymbol{v}_{x,k} \|_2^2 + 1 \right]^{-1}$ for all k. Plugging this value into (13) then leads to the revised objective

$$\begin{aligned}
\ell_x(\theta,\phi) \equiv \int \left\{ \right. & \tfrac{1}{\gamma} \left\| \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} - \boldsymbol{V}_x \left(\boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z \right) - \boldsymbol{W}_x \boldsymbol{b}_y - \boldsymbol{b}_x \right\|_2^2 \qquad (14) \\
& \left. + \sum_{k=1}^{\kappa} \log \left(\tfrac{1}{\gamma} \| \boldsymbol{v}_{x,k} \|_2^2 + 1 \right) + d \log \gamma + \left\| \boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z \right\|_2^2 \right\} \rho_{gt}^x(d\boldsymbol{x})
\end{aligned}$$

ignoring constant terms. Similarly we can optimize over $\boldsymbol{\mu}_z = \boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z$ in terms of the other variables. This is just a convex, ridge regression problem, with the optimum uniquely satisfying

$$\boldsymbol{W}_z \boldsymbol{x} + \boldsymbol{b}_z = \boldsymbol{V}_x^\top \left(\gamma \boldsymbol{I} + \boldsymbol{V}_x \boldsymbol{V}_x^\top \right)^{-1} \left[\left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} - \boldsymbol{W}_x \boldsymbol{b}_y - \boldsymbol{b}_x \right], \qquad (15)$$

which is naturally an affine function of $\boldsymbol{x}$ as required. After plugging (15) into (14), defining $\boldsymbol{\epsilon}_x \triangleq \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} - \boldsymbol{W}_x \boldsymbol{b}_y - \boldsymbol{b}_x$, and applying some linear algebra manipulations, we arrive at

$$\begin{aligned}
\bar{\ell}_x(\theta) &\triangleq \min_{\phi} \ \ell_x(\theta,\phi) \qquad (16) \\
&= \int \left\{ \boldsymbol{\epsilon}_x^\top \left(\boldsymbol{V}_x \boldsymbol{V}_x^\top + \gamma \boldsymbol{I} \right)^{-1} \boldsymbol{\epsilon}_x \right\} \rho_{gt}^x(d\boldsymbol{x}) + \sum_{k=1}^{\kappa} \log \left(\| \boldsymbol{v}_{x,k} \|_2^2 + \gamma \right) + (d - \kappa) \log \gamma,
\end{aligned}$$

noting that this minimization was accomplished without encountering any suboptimal local minima.

10.2 Optimization over parameters θ that are unique to $\bar{\ell}_x(\theta)$

The optimal $\boldsymbol{b}_x$ is just the convex maximum likelihood estimator given by the mean

$$\boldsymbol{b}_x = \int \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{x} \rho_{gt}^x(d\boldsymbol{x}) - \boldsymbol{W}_x \boldsymbol{b}_y = \left(\boldsymbol{I} - \boldsymbol{W}_x \boldsymbol{W}_y\right) \boldsymbol{c} - \boldsymbol{W}_x \boldsymbol{b}_y, \qquad (17)$$

where the second equality follows from Definition 1 in the main text. Plugging this value into (16) and applying a standard trace identity, we arrive at

$$\bar{\ell}_x(\theta) \equiv \operatorname{tr} \left[\boldsymbol{S}_{\epsilon_x} \left(\boldsymbol{V}_x \boldsymbol{V}_x^\top + \gamma \boldsymbol{I} \right)^{-1} \right] + \sum_{k=1}^{\kappa} \log \left(\| \boldsymbol{v}_{x,k} \|_2^2 + \gamma \right) + (d - \kappa) \log \gamma, \qquad (18)$$

where

$$\boldsymbol{S}_{\epsilon_x} \triangleq \text{Cov}_{\rho^x_{gt}}[\boldsymbol{\epsilon}_x] = (\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)\,\text{Cov}_{\rho^x_{gt}}[\boldsymbol{x}]\,(\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)^\top. \tag{19}$$

The remaining parameters $\{\boldsymbol{W}_x, \boldsymbol{W}_y, \boldsymbol{V}_x\}$ are all shared with the $\boldsymbol{y} \to \hat{\boldsymbol{x}} \to \hat{\boldsymbol{y}}$ cycle loss $\ell_y(\theta)$, so ostensibly we must include the full loss $\bar{\ell}_x(\theta) + \ell_y(\theta)$ when investigating local minima with respect to these parameters. However, there is one subtle exception that warrants further attention here. More specifically, the loss $\ell_y(\theta)$ depends on $\boldsymbol{V}_x$ only via the outer product $\boldsymbol{V}_x\boldsymbol{V}_x^\top$. Consequently, if $\boldsymbol{V}_x = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{V}}^\top$ denotes the singular value decomposition of $\boldsymbol{V}_x$, then $\ell_y(\theta)$ is independent of $\bar{\boldsymbol{V}}$ since $\boldsymbol{V}_x\boldsymbol{V}_x^\top = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{\Lambda}}^\top\bar{\boldsymbol{U}}^\top$, noting that $\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{\Lambda}}^\top$ is just a square matrix with squared singular values along the diagonal. It then follows that we can optimize $\bar{\ell}_x(\theta)$ over $\bar{\boldsymbol{V}}$ without influencing $\ell_y(\theta)$.

To this end we have the following:

Lemma 1 *At any minimizer (local or global) of $\bar{\ell}_x(\theta)$ with respect to $\bar{\boldsymbol{V}}$, it follows that $\bar{\boldsymbol{V}} = \boldsymbol{P}$ for some permutation matrix $\boldsymbol{P}$ and the corresponding loss satisfies*

$$\bar{\ell}_x(\theta) = \text{tr}\left[\boldsymbol{S}_{\epsilon_x}\boldsymbol{\Sigma}_{\epsilon_x}^{-1}\right] + \log|\boldsymbol{\Sigma}_{\epsilon_x}|, \quad \textit{where } \boldsymbol{\Sigma}_{\epsilon_x} \triangleq \boldsymbol{V}_x\boldsymbol{V}_x^\top + \gamma\boldsymbol{I}. \tag{20}$$

This result follows (with minor modification) from (Dai et al., 2019)[Corollary 3]. A related result also appears in (Lucas et al., 2019).

10.3 Optimization over parameters θ that are unique to $\ell_y(\theta)$

Since $\boldsymbol{y}$ has zero mean per Definition 1, the optimal $\boldsymbol{b}_y$ is the convex maximum likelihood estimator satisfying $\boldsymbol{b}_y = -\boldsymbol{W}_y\boldsymbol{b}_x$ (this assumes that $\boldsymbol{W}_y\boldsymbol{b}_x$ has not been absorbed into $\boldsymbol{y}$ as mentioned in the main text for notational simplicity). This leads to

$$\ell_y(\theta) \equiv \text{tr}\left[\boldsymbol{S}_{\epsilon_y}\boldsymbol{\Sigma}_{\epsilon_y}^{-1}\right] + \log|\boldsymbol{\Sigma}_{\epsilon_y}|, \quad \text{where } \boldsymbol{S}_{\epsilon_y} \triangleq (\boldsymbol{I} - \boldsymbol{W}_y\boldsymbol{W}_x)(\boldsymbol{I} - \boldsymbol{W}_y\boldsymbol{W}_x)^\top \tag{21}$$

and $\boldsymbol{\Sigma}_{\epsilon_y}$ is defined in the main text.

10.4 Optimizing the combined loss $\bar{\ell}_{cycle}(\theta)$

The above results imply that we may now consider jointly optimizing the combined loss

$$\bar{\ell}_{cycle}(\theta) \triangleq \bar{\ell}_x(\theta) + \ell_y(\theta) \tag{22}$$

over $\{\boldsymbol{W}_x, \boldsymbol{W}_y, \boldsymbol{V}_x\boldsymbol{V}_x^\top\}$; all other terms have already been optimized out of the model without encountering any suboptimal local minima. To proceed, consider the distribution $\rho^{\hat{y}}_{gt}$ of

$$\hat{\boldsymbol{y}} = \boldsymbol{W}_y\boldsymbol{x} + \boldsymbol{b}_y = \boldsymbol{W}_y\boldsymbol{A}\boldsymbol{y} + \boldsymbol{W}_y\boldsymbol{B}\boldsymbol{u} + \boldsymbol{W}_y\boldsymbol{c} + \boldsymbol{b}_y. \tag{23}$$

To satisfy the constraint the stipulated constraint $\rho^{\hat{y}}_{gt} = \rho^y_{gt}$ subject to the conditions of Definition 1, it must be that $\boldsymbol{W}_y\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{B} \in \text{null}[\boldsymbol{W}_y]$ (it will also be the case that $\boldsymbol{b}_y = -\boldsymbol{W}_y\boldsymbol{c}$ to ensure that $\hat{\boldsymbol{y}}$ has zero mean). From this we may conclude that

$$\begin{aligned}
\boldsymbol{S}_{\epsilon_x} &= (\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)\,\text{Cov}_{\rho^x_{gt}}[\boldsymbol{x}]\,(\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)^\top \\
&= (\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)\left[\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top\right](\boldsymbol{I} - \boldsymbol{W}_x\boldsymbol{W}_y)^\top \\
&= (\boldsymbol{A} - \boldsymbol{W}_x)(\boldsymbol{A} - \boldsymbol{W}_x)^\top + \boldsymbol{B}\boldsymbol{B}^\top,
\end{aligned} \tag{24}$$

where the middle equality follows because $\boldsymbol{y}$ and $\boldsymbol{u}$ are uncorrelated with identity covariance. Furthermore, let $\widetilde{\boldsymbol{D}} \in \mathbb{R}^{r_y \times r_x}$ denote any matrix such that $\widetilde{\boldsymbol{D}}\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{B} \in \text{null}[\widetilde{\boldsymbol{D}}]$. It then follows that $\boldsymbol{W}_y$ must equal some such $\widetilde{\boldsymbol{D}}$ and optimization of (22) over $\boldsymbol{W}_x$ will involve simply minimizing

$$\bar{\ell}_{cycle}(\theta) \equiv \text{tr}\left[(\boldsymbol{A} - \boldsymbol{W}_x)(\boldsymbol{A} - \boldsymbol{W}_x)^\top\boldsymbol{\Sigma}_{\epsilon_x}^{-1}\right] + \text{tr}\left[\left(\boldsymbol{I} - \widetilde{\boldsymbol{D}}\boldsymbol{W}_x\right)\left(\boldsymbol{I} - \widetilde{\boldsymbol{D}}\boldsymbol{W}_x\right)^\top\boldsymbol{\Sigma}_{\epsilon_y}^{-1}\right] + C \tag{25}$$

over $\boldsymbol{W}_x$, where C denotes all terms that are independent of $\boldsymbol{W}_x$. This is a convex problem with unique minimum at $\boldsymbol{W}_x = \boldsymbol{A}$. Note that this choice sets the respective $\boldsymbol{W}_x$-dependent terms to zero, the minimum possible value. Plugging $\boldsymbol{W}_x = \boldsymbol{A}$ into (25) and expanding the terms in C, we then arrive at the updated loss

$$\begin{aligned}\bar{\ell}_{cycle}(\theta) &\equiv \text{tr}\left[\boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{\Sigma}_{\epsilon_x}^{-1}\right] + \log|\boldsymbol{\Sigma}_{\epsilon_x}| + \log\left|\boldsymbol{\Sigma}_{\epsilon_y}\right| \\ &= \text{tr}\left[\boldsymbol{B}\boldsymbol{B}^\top\left(\boldsymbol{V}_x\boldsymbol{V}_x^\top + \gamma\boldsymbol{I}\right)^{-1}\right] + \log\left|\boldsymbol{V}_x\boldsymbol{V}_x^\top + \gamma\boldsymbol{I}\right| + \log\left|\widetilde{\boldsymbol{D}}\boldsymbol{V}_x\boldsymbol{V}_x^\top\widetilde{\boldsymbol{D}}^\top + \gamma\boldsymbol{I}\right|.\end{aligned} \tag{26}$$

Minimization of this expression over $\boldsymbol{V}_x$ as γ becomes arbitrarily small can be handled as follows. If any $\boldsymbol{V}_x$ and γ are a local minima of (26), then $\{\alpha = 1, \beta = 0\}$ must also be a local minimum of

$$\begin{aligned}&\bar{\ell}_{cycle}(\alpha,\beta) \triangleq \\ &\quad \text{tr}\left[\boldsymbol{B}\boldsymbol{B}^\top\left(\alpha\boldsymbol{\Sigma}_{\epsilon_x} + \beta\boldsymbol{B}\boldsymbol{B}^\top\right)^{-1}\right] + \log\left|\alpha\boldsymbol{\Sigma}_{\epsilon_x} + \beta\boldsymbol{B}\boldsymbol{B}^\top\right| + \log\left|\alpha\boldsymbol{\Sigma}_{\epsilon_y} + \beta\widetilde{\boldsymbol{D}}\boldsymbol{B}\boldsymbol{B}^\top\widetilde{\boldsymbol{D}}^\top\right| \\ &\quad = \text{tr}\left[\boldsymbol{B}\boldsymbol{B}^\top\left(\alpha\boldsymbol{\Sigma}_{\epsilon_x} + \beta\boldsymbol{B}\boldsymbol{B}^\top\right)^{-1}\right] + \log\left|\alpha\boldsymbol{\Sigma}_{\epsilon_x} + \beta\boldsymbol{B}\boldsymbol{B}^\top\right| + \log\left|\alpha\boldsymbol{\Sigma}_{\epsilon_y}\right|.\end{aligned} \tag{27}$$

If we exclude the second log-det term, then it has been shown in (Wipf and Nagarajan, 2007) that loss functions in the form of (27) have a monotonically decreasing path to a unique minimum as $\beta \to 1$ and $\alpha \to 0$. However, given that the second log-det term is a monotonically decreasing function of α, it follows that the entire loss from (27) has a unique minimum as $\beta \to 1$ and $\alpha \to 0$. Consequently, it must be that at any local minimum of (26) $\boldsymbol{V}_x\boldsymbol{V}_x^\top = \boldsymbol{B}\boldsymbol{B}^\top$ in the limit as $\gamma \to 0$. Moreover, the feasibility of this limiting equality is guaranteed by our assumption that $r_z \geq r_c - r_y$ (i.e., if $r_z < r_c - r_y$, then $\boldsymbol{V}_x$ would not have sufficient dimensionality to allow $\boldsymbol{V}_x\boldsymbol{V}_x^\top = \boldsymbol{B}\boldsymbol{B}^\top$).

10.5 Final Pieces

We have already established that at any local minimizer $\{\theta^*, \phi^*\}$ it must be the case that $\boldsymbol{W}_x^* = \boldsymbol{A}$ and $\boldsymbol{W}_y^* = \widetilde{\boldsymbol{D}}$. Moreover, we also can infer from (17) and Section 10.3 that at any local minimum we have

$$\boldsymbol{b}_x^* = \left(\boldsymbol{I} - \boldsymbol{W}_x^*\boldsymbol{W}_y^*\right)\boldsymbol{c} - \boldsymbol{W}_x^*\boldsymbol{b}_y^* = \left(\boldsymbol{I} - \boldsymbol{W}_x^*\boldsymbol{W}_y^*\right)\boldsymbol{c} + \boldsymbol{W}_x^*\boldsymbol{W}_y^*\boldsymbol{b}_x^* = \left(\boldsymbol{I} - \boldsymbol{A}\widetilde{\boldsymbol{D}}\right)\boldsymbol{c} + \boldsymbol{A}\widetilde{\boldsymbol{D}}\boldsymbol{b}_x^* \tag{28}$$

from which it follows that $\left(\boldsymbol{I} - \boldsymbol{A}\widetilde{\boldsymbol{D}}\right)\boldsymbol{c} = \left(\boldsymbol{I} - \boldsymbol{A}\widetilde{\boldsymbol{D}}\right)\boldsymbol{b}_x^*$. This along is not sufficient to guarantee that $\boldsymbol{b}_x^* = \boldsymbol{c}$ is the unique solution; however, once we include the additional constraint $\rho_{gt}^y = \rho_\theta^{\hat{y}}$ per the Proposition 2 statement, then $\boldsymbol{b}_x^* = \boldsymbol{c}$ is uniquely determined (otherwise it would imply that $\hat{\boldsymbol{y}}$ has a nonzero mean). It then follows that $\boldsymbol{b}_y^* = -\boldsymbol{W}_y^*\boldsymbol{b}_x^* = -\widetilde{\boldsymbol{D}}\boldsymbol{c}$.

And finally, regarding $\boldsymbol{V}_x^*$, from Section 10.4 we have that $\boldsymbol{V}_x^*\left(\boldsymbol{V}_x^*\right)^\top = \boldsymbol{B}\boldsymbol{B}^\top$. Although this does *not* ensure that $\boldsymbol{V}_x^* = \boldsymbol{B}$, we can conclude that $\text{span}[\bar{\boldsymbol{U}}] = \text{span}[\boldsymbol{B}]$. Furthermore, we know from Lemma 1 and the attendant singular value decomposition that $\boldsymbol{V}_x^* = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Lambda}}\boldsymbol{P}^\top$ and $\left(\boldsymbol{V}_x^*\right)^\top\boldsymbol{V}_x^* = \boldsymbol{P}^\top\bar{\boldsymbol{\Lambda}}^\top\bar{\boldsymbol{\Lambda}}\boldsymbol{P}$. Therefore, up to an arbitrary permutation, each column of $\boldsymbol{V}_x^*$ satisfies

$$\|\boldsymbol{v}_{x,k}^*\|_2^2 = \begin{cases}\bar{\lambda}_k^2, & \forall\ k = 1,\ldots,\text{rank}[\boldsymbol{B}] \\ 0, & \forall\ k = \text{rank}[\boldsymbol{B}]+1,\ldots,r_z\end{cases} \tag{29}$$

where $\bar{\lambda}_k$ is an eigenvalue of $\bar{\boldsymbol{\Lambda}}$. Collectively then, these results imply that $\boldsymbol{V}_x^* = \left[\widetilde{\boldsymbol{B}}, \boldsymbol{0}\right]\boldsymbol{P}^\top$, where $\widetilde{\boldsymbol{B}} \in \mathbb{R}^{r_x \times \text{rank}[\boldsymbol{B}]}$ satisfies $\text{span}[\widetilde{\boldsymbol{B}}] = \text{span}[\boldsymbol{U}] = \text{span}[\boldsymbol{B}]$.

11 Proof of Corollary 3

From (15) in the proof of Proposition 2 and the derivations above, we have that at any optimal encoder solution $\phi^* = \{\boldsymbol{W}_z^*, \boldsymbol{b}_z^*\}$, both $\boldsymbol{W}_z^*$ and $\boldsymbol{b}_z^*$ are formed by left multiplication by $\left(\boldsymbol{V}_x^*\right)^\top$. Then based on Proposition 2 and

the stated structure of $\boldsymbol{V}_x^*$, it follows that $\boldsymbol{W}_z^* = \boldsymbol{P}\begin{bmatrix} \widetilde{\boldsymbol{W}}_z^* \\ \boldsymbol{0} \end{bmatrix}$ and $\boldsymbol{b}_z^* = \boldsymbol{P}\begin{bmatrix} \widetilde{\boldsymbol{b}}_z^* \\ \boldsymbol{0} \end{bmatrix}$, where $\widetilde{\boldsymbol{W}}_z^*$ has rank$[\boldsymbol{B}]$ rows and $\widetilde{\boldsymbol{b}}_z^* \in \mathbb{R}^{\text{rank}[\boldsymbol{B}]}$. Finally, there exists a bijection between $\boldsymbol{x}$ and $\{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\}$ given that

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{W}_y^* \boldsymbol{x} + \boldsymbol{b}_y^* \text{ and } \widetilde{\boldsymbol{\mu}}_z = \widetilde{\boldsymbol{W}}_z^* \boldsymbol{x} + \widetilde{\boldsymbol{b}}_z^* \quad (\text{for } \boldsymbol{x} \to \{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\} \text{ direction}) \\ \boldsymbol{x} &= \boldsymbol{W}_x^* \boldsymbol{y} + \boldsymbol{V}_x^* \boldsymbol{P}\begin{bmatrix} \widetilde{\boldsymbol{\mu}}_z \\ \boldsymbol{0} \end{bmatrix} + \boldsymbol{c} \quad (\text{for } \{\boldsymbol{y}, \widetilde{\boldsymbol{\mu}}_z\} \to \boldsymbol{x} \text{ direction}), \end{aligned} \tag{30}$$

completing the proof.