# Coherent and Consistent Relational Transfer Learning with Auto-encoders

Harald Strömfelt[1,3], Luke Dickens[2], Artur d'Avila Garcez[3] and Alessandra Russo[1]

[1]Imperial College London, Exhibition Rd, South Kensington, London SW7 2BX, UK
[2]University College London, Gower St, London WC1E 6BT, UK
[3]City, University of London, Northampton Square, London EC1V 0HB, UK

**Abstract**

Human defined concepts are inherently transferable, but it is not clear under what conditions they can be modelled effectively by non-symbolic artificial learners. This paper argues that for a transferable concept to be learned, the system of relations that define it must be coherent across domains and properties. That is, they should be consistent with respect to relational constraints, and this consistency must extend beyond the representations encountered in the source domain. Further, where relations are modelled by differentiable functions, their gradients must conform – the functions must at times move together to preserve consistency. We propose a Partial Relation Transfer (PRT) task which exposes how well relation-decoders model these properties, and exemplify this with ordinality prediction transfer task, including a new data set for the transfer domain. We evaluate this on existing relation-decoder models, as well as a novel model designed around the principles of consistency and gradient conformity. Results show that consistency across broad regions of input space indicates good transfer performance, and that good gradient conformity facilitates consistency.

**Keywords**

Representation Learning, Relation Learning, Variational AutoEncoders, Concept Learning

## 1. Introduction

In many situations, concepts that pertain to one set of data can also be relevant to another [1, 2]. Take, for instance, the general concept of ordinality, whose semantics are defined by relations: isSuccessor, isPredecessor, isGreater, isLess and isEqual; together with their constraints. Successfully capturing this concept involves learning the corresponding relations such that they maintain data set and property independence, with no retraining. This is to say that they have been abstracted from the specific property and act instead as a generic set of characterizing relations for the semantics of ordinality. For this, we argue that the relations must be *consistent* with their expected constraints and coherent across ordinal properties spanning different data sets, which means their consistency is maintained regardless of data set or particular ordinal property.

As a concrete example, suppose that we have successfully learned to order images of numbers by their abstract digit identity, and are presented with a new data set containing images of individual stacks of blocks. Suppose then that we wish to obtain an ordering over them, such

*Submitted to the 15th International Workshop On Neural-Symbolic Learning and Reasoning (NeSy '21)*
h.stromfelt17@imperial.ac.uk (H. Strömfelt)

that we can compare arbitrary data instances using the above relations. Provided that the learned relations are consistent with their expected constraints, it should be possible to obtain an encoding that establishes each successor, via our isSuccessor relation, and immediately be able to compare data instance over the remaining relations. Following this logic, the primary purpose of this paper is to evaluate under which conditions a relation-decoder model is able to obtain the ordinality concept. We do this by taking a set of popular relation-decoder models, including a proposed Dynamic Comparator (DC) model, and assess 1. their consistencies as measured in the source data set, and 2. their ability to perform a Partial Relation Transfer (PRT) task to a novel target data set, which measures the robustness of their consistencies across domains. The evaluation takes place in two steps. In the first, we learn the above set of ordinality relations by ordering *MNIST* images based on their abstract digit identity and report each model's consistency profile. In the next step, we take the now pretrained isSuccessor relation-decoder and apply it to a proposed *BlockStacks* data set, which consists of images of multicolored block stacks. Each stack contains a single red block at various heights, which we use to test the degree to which ordering the encodings of each block stack image, subject to the pretrained isSuccessor relation, leads to transferred prediction accuracy across the remaining relations. In summary, the contributions of our work are:

- We devise an experimental setup that can expose the degree to which learning relations leads to concept abstraction, together with a new *BlockStacks* data set that presents a challenging ordering task based on a complex property.
- We introduce a set of data set agnostic characteristic measures for relation-decoders which can help determine their ability to perform PRT.
- We present a Dynamic Comparator model that achieves excellent PRT.
- Finally, we present a comprehensive analysis of model characteristics against corresponding PRT performance, for a set of popular relation-decoders.

The rest of the paper is presented as follows. Section 2 firstly positions our paper with respect to related work. Section 3 formalises the PRT task and outlines the architecture we employ to solve it, including the proposed DC relation-decoder model. We then define how we compute model consistency and gradient-conformity in Section 4. Finally, we provide results and analysis in Section 5, with concluding remarks in Section 6.

## 2. Related Work

Relational representations play a prominent role in Knowledge Graph Embedding (KGE), wherein sets of relation-decoders are jointly learned, through triplet link prediction, in order to obtain a semantic latent factor representation for entities [3, 4, 5, 6, 7, 8, 9, 10, 11]. In principle any KGE link prediction model can be employed in this work, but we focus on those that assume a Euclidean representation space and do not require any additional per-triplet engineering. Although KGE methods typically do not use a shared auto-encoder as we do in this paper, Schlichtkrull et al. [12] did adopt an auto-encoding framework, where a graph neural network is used as the encoder, however they did not work with visual data and the model was not applied to transfer. Disentanglement, which also aims to learn semantic representations

for data is of relevance to this work [13, 2], wherein multiple methods have been proposed, for example using Generative Adversarial Networks [14] and VAEs [15, 1, 16, 17, 18, 19, 20]. Of particular relevance to our work are investigations looking at the transferability of disentangled representations [21, 22, 23], but these did not include relation learning. A bridge between relation learning and disentanglement, wherein relation-decoders are employed as a semi-supervision to VAEs, can be found in [24, 25, 26]. Lastly, we note that our experimental setup is most remnant of domain adaptation [27]. To the best of our knowledge, no work has compared relation-decoders in their ability to abstract concepts, as measured by their consistency and its transfer across domains.

## 3. The Partial Relation Transfer Task and Model

Partial Relation Transfer (PRT) is at its core a domain adaptation task [27], wherein we have a source and target data domain, consisting of a set of images, $X_s$ and $X_t$, respectively, and a set of shared relation prediction tasks, $\mathscr{R} = \{r_1, \dots, r_n\}$. We approximate each relation using a relation-decoder $\phi_r^M : Z \times Z \rightarrow [0, 1]$, where $Z$ denotes a latent space that contains all image encodings $\boldsymbol{z}_i \in Z$. The superscript $M$ denotes a specific relation-decoder model, as we test multiple variants. To obtain embeddings we use a domain-specific auto-encoder, consisting of an encoder $\psi_{enc}^{s/t} : X \rightarrow Z$ and decoder $\psi_{dec}^{s/t} : Z \rightarrow X$, which helps to minimise information loss through reconstruction of the input image[1].

The evaluation takes place as a two-step procedure. In the first, all relation decoders are trained in the source domain, as a semi-supervision to the auto-encoder, using available labels, $\boldsymbol{y}^s \in \mathbb{R}^{|\mathscr{R}|\times|X_s|\times|X_s|}$, that specify whether a relation $r \in \mathscr{R}$ holds between image $\boldsymbol{x}_i, \boldsymbol{x}_j \in X_s$. Here, $|\cdot|$ denotes the cardinality of the operand set, but in practice we only use a small fraction of the available labels. In the second evaluation step, we initialise a new auto-encoder to be applied to the target dataset and use a subset of the pretrained relation-decoders, with labels $\boldsymbol{y}^t \in \mathbb{R}^{|\mathscr{R}|\times|X_t|\times|X_t|}$, to act as fixed-parameter 'guides' for the encoder.

To obtain informative data encodings, we use a Variational AutoEncoder (VAE), specifically the $\beta$-VAE, given its simplicity and demonstrated ability to separate distinct factors in the latent representation [1, 15, 28]. The $\beta$-VAE achieves this by optimising the ELBO objective, which for the purposes of this paper we express as a loss over both encoder and decoder:

$$\mathscr{L}_{\beta\text{-VAE}}^{ELBO} = \mathscr{L}(\psi_{enc}^{s/t}, \psi_{dec}^{s/t}) + \beta\mathscr{L}(\psi_{enc}^{s/t}, \mathcal{N}(\mathbf{0}, \mathbb{1})), \tag{1}$$

where an additional $\beta$ scalar hyperparameter is used to influence disentanglement through stronger distribution matching pressure to an isotropic zero-mean Gaussian prior, $\mathcal{N}(\mathbf{0}, \mathbb{1})$. When $\beta = 1$ we obtain the original VAE objective [28]. We provide the full ELBO loss, with a detailed explanation, in Appendix B. Each experiment involves taking embeddings from a corresponding encoder and passing them through to sets of relation-decoders (either the full set in the case in the source domain, or only a subset in the target domain). We can treat each relation-decoder as producing a prediction $\hat{y}_{rij}$ for whether relation $r$ holds between data

[1] Further analysis on the performance of BlockStacks embeddings for domain-dependent task can be found in Appendix E

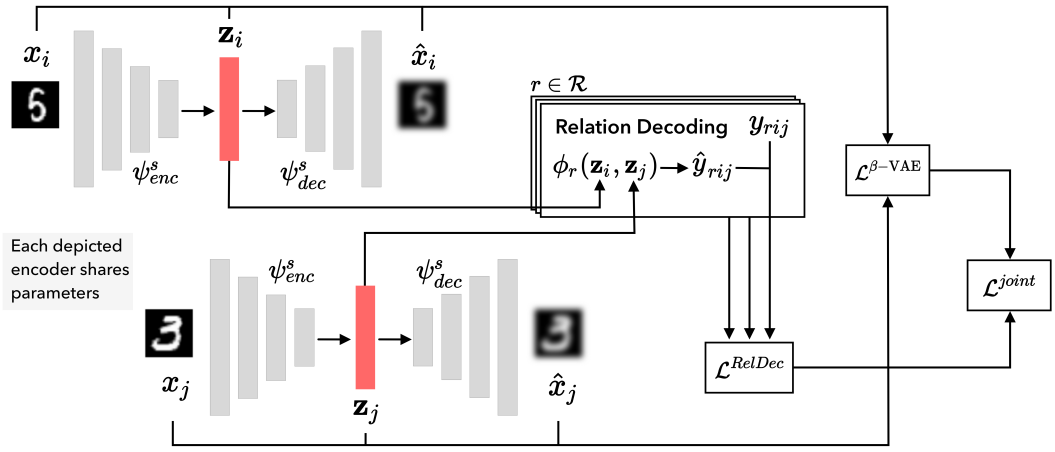


**Figure 1:** Depiction of the architecture we use for PRT. In this diagram, we show how the initial relation learning is performed on the source *MNIST* dataset. Moving to the target domain involves using $\psi^t_{enc/dec}$ and fixing parameters for each included $\phi_r$ relation-decoder.

instances $i$ and $j$ [5]. Using the ground truth $y_{rij}$, we can then compute the loss over all relation-decoders, $\mathscr{L}^{RelDec}$, as the binary cross-entropy of prevision versus ground truth. This gives us the final joint objective between VAE and relation-decoders:

$$\mathscr{L}^{joint} = \mathscr{L}^{ELBO}_{\beta\text{-VAE}} - \lambda \underbrace{\mathbb{E}_{r,y_{rij},\boldsymbol{z}_i,\boldsymbol{z}_j}[y_{rij}\log(\hat{y}_{rij}) + (1-y_{rij})\log(1-\hat{y}_{rij})]}_{\mathscr{L}^{RelDec}}, \tag{2}$$

where $\lambda$ is a scalar weighting parameter.

### 3.1. Dynamic Comparator

In our analysis, we include a proposed low-complexity, but nonetheless expressive, "Dynamic Comparator" (DC) model, which is designed to model systems of relations, whilst encouraging desirable properties for PRT. The overall DC model is composed of two modes, a distance-based measure, $\phi^{\dagger}_r$, that can compute how close the vector difference between two inputs is to a positive or negative valued reference vector, and a step-like function, $\phi^{\ddagger}_r$, that determines the sign of the difference between two points, optionally with an offset. The overall DC model is given by[2]:

$$\phi^{DC}_r(\boldsymbol{z}_i, \boldsymbol{z}_j) = a_0 \cdot \underbrace{\sigma_0\big(\eta_0(\|\boldsymbol{u} \odot (\boldsymbol{z}_i - \boldsymbol{z}_j + \boldsymbol{b}_{\dagger})\|^2_2)\big)}_{\phi^{\dagger}_r} + a_1 \cdot \underbrace{\sigma_1\big((\eta_1 \cdot \boldsymbol{u}^{\top}(\boldsymbol{z}_i - \boldsymbol{z}_j + \boldsymbol{b}_{\ddagger}))\big)}_{\phi^{\ddagger}_r}. \tag{3}$$

[2]In the main text we report results for this DC model, but we can use any function that has the required characteristics for $\phi^{\dagger}$ and $\phi^{\ddagger}$. We include results for other versions in Appendix D.

where $\boldsymbol{a} = \texttt{Softmax}(\boldsymbol{A}) \in \mathbb{R}^2$ is an attention weighting between the two modes, and ensures that $\phi^{DC}$ is bound to [0,1]. $\sigma_0, \sigma_1$ are an exponential and sigmoid function, respectively; $\boldsymbol{u} = \text{Softmax}(\boldsymbol{U}) \in \mathbb{R}^m$ is an attention mask which is applied to $m$-dimensional latent embeddings; $\boldsymbol{b}_\dagger, \boldsymbol{b}_\ddagger \in \mathbb{R}^m$ are learnable bias terms that enables an offset to each mode; and $\eta_0 \in \mathbb{R}^+$ are non-negative and $\eta_1 \in \mathbb{R}$ any-valued scalar terms, respectively. Lastly, $\odot$ denotes the Hadamard product (elementwise multiplication) and $\|\cdot\|_2$ is the $L2$-norm. Due to a convergence issue when using a pretrained DC with fixed parameters, we needed to use a flexible fitting procedure in which we enable the DC parameters to train in the target domain, but with the additional loss term $\|\boldsymbol{\rho}^* - \boldsymbol{\rho}\|$, between pretrained $\boldsymbol{\rho}^*$ and untrained parameters $\boldsymbol{\rho}$, respectively. In all cases we evaluated the final parameter values in the target domain and found them to be approximately equivalent to the $\boldsymbol{\rho}^*$. We did not apply this method to the other models as they were all able to fit the isSuccessor relation in the target domain.

## 4. Measuring relation-decoder characteristics

In this section we describe a series of measures that we use to understand more about the intrinsic characteristics of each relation-decoder, which together help identify the behaviour of each relation-decoder model and provide insight regarding their respective PRT performance.

For any system of relations, we can write down a truth-table that defines the valid truth-states that they may collectively take, which we expect our relation-decoders to model. For example, we know that any time isGreater is true, isLess must not be. By assuming that each relation-decoder output is pairwise conditionally independent given $\boldsymbol{z}_i, \boldsymbol{z}_j$, for instance,

$$p(\text{isGreater}, \text{isLess}|\boldsymbol{z}_i, \boldsymbol{z}_j) = p(\text{isGreater}|\boldsymbol{z}_i, \boldsymbol{z}_j)p(\text{isLess}|\boldsymbol{z}_i, \boldsymbol{z}_j),$$

we can produce a probability statement for whether the relations are consistent with valid entries to the truth-table. Taking $r_1 = \text{isGreater}$ and $r_2 = \text{isLess}$ as our entire system of relations, we can produce the following truth-table conversion, where invalid entries are omitted:

| $r_1(x_i, x_j)$ | $r_2(x_i, x_j)$ | $\mathscr{F}(r_1, r_2)$ |
|---|---|---|
| $T$ | $F$ | $T$ |
| $F$ | $T$ | $T$ |
| $F$ | $F$ | $T$ |

$$\implies \begin{aligned} \mathscr{F}(r_1, r_2) = \forall x_i, x_j \big( & (r_1(x_i, x_j) \wedge r_2(x_i, x_j)) \\ & \vee (\neg r_1(x_i, x_j) \wedge r_2(x_i, x_j)) \\ & \vee (\neg r_1(x_i, x_j) \wedge \neg r_2(x_i, x_j)) \big) \end{aligned} \tag{4}$$

which, using our relation-decoders for each relation and with $\boldsymbol{z}_{i,j} = \psi_{enc}(\boldsymbol{x}_{i,j})$ and $\neg\phi_r(\boldsymbol{z}_i, \boldsymbol{z}_j) = 1 - \phi_r(\boldsymbol{z}_i, \boldsymbol{z}_j)$, we express the probability of $\mathscr{F}$ being true as:

$$\begin{aligned} p(\mathscr{F}|\boldsymbol{z}_i, \boldsymbol{z}_j) = \big( & (\phi_{r_1}(\boldsymbol{z}_i, \boldsymbol{z}_j) \cdot \phi_{r_2}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \\ & + ((1 - \phi_{r_1}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \cdot \phi_{r_2}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \\ & + ((1 - \phi_{r_1}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \cdot (1 - \phi_{r_2}(\boldsymbol{z}_i, \boldsymbol{z}_j))\big). \end{aligned} \tag{5}$$

Finally, since $\mathscr{F}$ should hold for all input combinations, we heavily penalise violations by using a binary cross-entropy loss between $\mathscr{F}$ and the expected outcome:

$$H_{True}(p(\mathscr{F})) = -\frac{1}{N} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in Z} 1 \cdot \log p(\mathscr{F}|\boldsymbol{z}_i, \boldsymbol{z}_j), \tag{6}$$

where $Z$ is the latent space, as we can compute this score for any samples from this space[3] and $N$ is a normalising constant, equal to the number of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ sample pairs used in the calculation. We refer to this measure as Con-A referring to the fact that we use it to measure consistency across multiple relations.

To provide a deeper understanding about how relation-decoders collectively interact with their inputs, we use a gradient evaluation to see whether models respond similarly to changes in their input. For a set of relations, we define the gradient-conformity (GC) of relation $r_i$ against all others by the following cosine-similarity:

$$GC = \left| \frac{\boldsymbol{d}_i^T \boldsymbol{d}_j}{||\boldsymbol{d}_i||_2 \, ||\boldsymbol{d}_j||_2} \right| \quad \text{where } \boldsymbol{d}_i = \left. \frac{d\phi_{r_i}}{d\boldsymbol{z}^c} \right|_{\boldsymbol{z}^c = \boldsymbol{z}_s^c} \text{ and } \boldsymbol{d}_j = \left. \frac{d\phi_{r_j}}{d\boldsymbol{z}^c} \right|_{\boldsymbol{z}^c = \boldsymbol{z}_s^c}, \quad \forall i \neq j \tag{7}$$

where $|\cdot|$ denotes the absolute of the operand and $\boldsymbol{z}^c$ is the concatenation of each relation-decoder's inputs, with gradients evaluated at reference inputs $\boldsymbol{z}_s^c$. GC will be 1 if gradients are aligned and zero if orthogonal[4].

## 5. Results

This section presents results for the PRT task on a range of relation-decoder models. In the *source domain*, we learn a system of binary relations: $\mathscr{R}$ ={isSuccessor (S), isPredecessor (P), isGreater (G), isEqual (E), isLess (L)}, on digits represented in MNIST images, alongside a $\beta$-VAE. In the *target domain*, we take the pretrained S relation as a fixed-parameter guide for a new $\beta$-VAE applied to *BlockStacks* images (see Appendix A for BlockStacks image examples), and then evaluate PRT accuracy on the held-out G, E, L and P relations. Relation-decoder models compared here are: TransR [29], HolE [30], NTN [3], our proposed DC and a basic neural-network baseline, NN. NN is a simple four-layer ($l_{\text{in}}, l_1, l_2, l_{\text{out}}$) neural-network with layer sizes $l_{\text{in}} = 2d_z, l_1 = 2d_z$ and $l_2 = d_z$, with ReLU activations. The final output layer $l_{\text{out}}$ is a single value passed through a sigmoid function, to bound the output to [0,1]. Further model details are provided in Appendix C.

We vary $\beta$ only in the source domain, ranging across values $\{1, 4, 8, 12\}$, but fix it in the target domain. $\lambda$ is fixed in both domains (see Appendix C.3 for further details on hyperparameter settings). For Con-A and GC measures, we produce encodings for three data splits: data-embeddings, where all inputs are encodings of a domain's test data; interpolation, where we obtain an empirical mean and variance for the domain's data-embeddings and sample from a corresponding Gaussian distribution; and extrapolation, where we sample from regions strictly outside the data-embeddings region.

Figure 2-top provides relation-decoder prediction accuracy in both the source *MNIST* (left), and target *BlockStacks* (right), domains. Key observations are that DC produces excellent PRT performance, whilst NN, NTN and HolE all see some degradation from their source accuracies. TransR seems to maintain a similar accuracy profile. We include $\beta$'s impact on these performances in Figure 2-bottom. Barring DC which has little discernible change in either

[3] in practice as we cannot include every encoding combination, we provide an estimate.

[4] We can evaluate against this measure for arbitrary samples from $Z$.

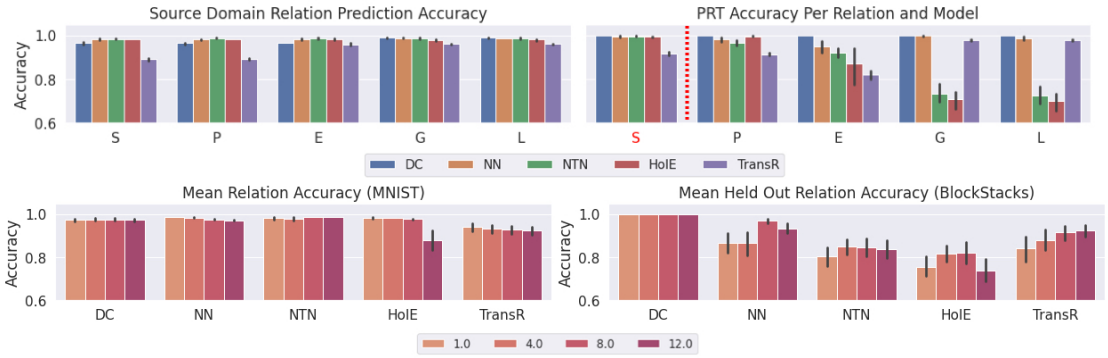


**Figure 2: [Top]** Relation-decoder prediction accuracy per relation and model, in the source (left) and target domains. Relations are abbreviated on the $x$-axis by { S: isSuccessor, P: isPredecessor, E: isEqual, G: isGreater, L: isLess }, with a red highlight identifying which relation is included as a guide for $\psi^t_{enc}$. **[Bottom]** $\beta$ impact profiles for each relation-decoder model, aggregated across all relations in the source (left) domain and aggregated only for held-out relations in the target (right) domain. In all cases, higher values are better.

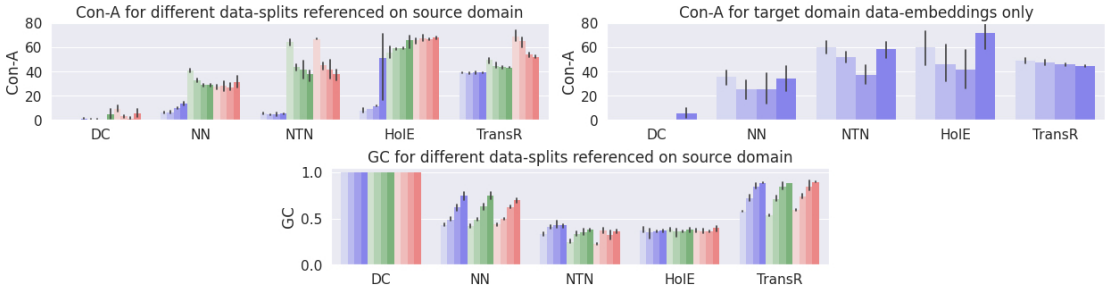


**Figure 3: [Top]** Con-A values for each relation-decoder model, referenced to source (left) and target (right) domains (lower values better). **[Bottom]** GC values for each relation decoder (higher values better). In all plots, darker color shades denote higher values of $\beta$, corresponding to greater disentanglement pressure from the $\beta$-VAE. In top-left and bottom plots, blue, green and red groups show results for data-embeddings, interpolation and extrapolation embeddings respectively (see main text for details).

domain, PRT performance is significantly impacted by $\beta$ in all models, but has little effect in the source domain. Additionally, TransR has a strong positive correlation with $\beta$, whereas NN, NTN and HolE produce the best PRT performance with intermediate disentanglement pressure. To interrogate further how $\beta$ affects each model, we provide: (Figure 3-top) mean relation Con-A referenced to both source (left) and target (right) domain embeddings; and (Figure 3-bottom) source domain referenced GC measures for each model. In the left and bottom plots, blue (left group), green (middle group) and red (right group) show results for the data-embeddings, interpolation and extrapolation regions of latent space, in respective order. From the source domain Con-A results, we note that DC shows excellent consistency across relations in all regions. Most other models have worse interpolation and extrapolation

consistency. Increasing $\beta$ appears to give some improvement for all but HolE, but there are indications that this trend does not persist into the largest $\beta = 12$ value. Interestingly, Con-A values for target data-embeddings (right) are notably worse than for source data-embeddings, with values closer to those for interpolation or extrapolation in the source domain. For GC, DC performance is close to 1 for all $\beta$ with no discernible change. All other models show a weaker GC with positive correlation between GC and $\beta$. TransR and NN achieve significantly higher GC than NTN and HolE.

## 5.1. Key Experimental Results

### 5.1.1. Does good source task accuracy lead to successful PRT?

Since we transfer pretrained models from source to target domain and ensure that the target encoder, $\psi^t_{enc}$, fits its encodings to S, we might expect that relation-decoding performance will be the same in both domains. However, despite DC, NN, NTN and HolE all performing close to 100% accuracy, and TransR achieving above 80%, across all relations, and with all relations able to achieve similar prediction accuracy (or better in the case of DC) on the guide relation S, PRT performance varies significantly across models. It is firstly evident that DC is successful at PRT, sustaining approximately 100% accuracy across all held-out relations. NN achieves mostly good performance, with greater degradation across P and E relations. Although HolE and NTN both achieve good PRT for P, there is increasing degradation across E and G, L relations. TransR is able to achieve strong relative performance where PRT accuracy per relation is comparable to what was possible in the source domain. These results indicate that source accuracy alone is not enough to determine whether models will be successful at PRT.

### 5.1.2. How does $\beta$ affect Con-A and GC and how does this impact model coherence?

To provide an overview of how increased disentanglement pressure affects each model we can firstly compare how $\beta$ affects model performance in both source and target domain. Figure 2-bottom demonstrates that, although relation prediction accuracies for most models either do not respond, or respond negatively, to increases in $\beta$ in the source domain, their PRT behaviour differs significantly across models: DC shows no discernible change, whilst NN, NTN and HolE all show a parabolic response with a maximum PRT around $\beta = 8$; TransR shows a general positive correlation but with diminishing returns above $\beta = 8$. To gain further insight into the role of disentanglement pressure, it is necessary to look at how each model's intrinsic behaviour responds to $\beta$ changes.

First, we attempt to expose the relationship between $\beta$ and consistency and whether this has any effect on PRT performance. By Figure 3-top, DC clearly outperforms all other models on Con-A and this coincides with better PRT performance. The next best performing model on Con-A in the source domain is also the next best on PRT performance. In most cases Con-A degrades for all models when moving from data-embeddings to interpolation and extrapolation, but the degree of degradation changes depending on the model. Interestingly, across all models, their target Con-A is notably close to that of interpolation or extrapolation in the source domain analysis. This suggests that guiding $\psi^t_{enc}$ to fit relation S produces data embeddings that lie in the interpolation or extrapolation regions with respect to *MNIST* embeddings. This

suggests that a relation-decoder model's ability to retain consistency over regions of latent space beyond where *MNIST* embeddings are found leads to improved PRT. These findings provide compelling evidence in support of our claim that consistency across relations is important for PRT performance.

Secondly, we examine how gradient-conformity affects PRT performance. To achieve successful PRT, fitting the target encoder to a single pretrained relation should lead to embeddings that are structured correctly with respect to the other pretrained relations. For this to be possible there must be a degree of conformity between how each model computes its system of relations. As an extreme case, suppose we have a two-dimensional latent representation, with two relations that are each calculated using entirely different dimensions of latent space. By fitting an encoder to one of these relations, there is no guarantee that the latent dimension, that the other relation requires, receives the necessary guidance. DC shows excellent and stable GC values (near 1) across all conditions. This is by design as the use of masks per relation ensures that if masks match for any two relations, then their gradients will be either parallel or anti-parallel. Excluding HolE, all remaining models show a positive correlation between GC and $\beta$, and it appears that models with either higher GC values, or $\beta$ response, typically perform better at PRT. Together this provides tentative evidence to suggest that GC is important to model coherence, as measured by their PRT performance. It is possible that we do not see a monotonic benefit of GC against PRT, due to no further extrapolation or interpolation Con-A gains with $\beta > 8$.

## 6. Conclusion

We provide a comprehensive analysis of relation-decoder characteristics when learning the system of relations that together define the semantics of a concept. We then compare these characteristics with a Partial Relation Transfer task setting, which determines whether, given logical constraints between relations, fitting embeddings to one relation-decoder leads to embeddings that satisfy all other relations in terms of their logical consistency and accuracy. Our results demonstrate that model consistency, and possibly gradient-conformity, across different regions of input space together determine whether a set of relation-decoders have learned a consistent and coherent notion of a given concept, in this case ordinality. These measures make it possible to check whether a set of relation-decoders have indeed learned a transferable concept, or if they are limited to a single data domain and property.

## References


[1] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, in: 5th International Conference on Learning Representations, {ICLR}, Toulon, France, 2017.

[2] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, A. Lerchner, Towards a Definition of Disentangled Representations, arXiv preprint arXiv:1812.02230 (2018). URL: http://arxiv.org/abs/1812.02230. doi:arXiv:1812.02230v1. arXiv:1812.02230.

[3] R. Socher, D. Chen, C. Manning, D. Chen, A. Ng, Reasoning With Neural Tensor Networks for Knowledge Base Completion, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 926–934. arXiv:arXiv:1301.3618v2.

[4] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex Embeddings for Simple Link Prediction, in: Proceedings of the 33nd International Conference on Machine Learning, {ICML}, New York, NY, USA, 2016, pp. 2071–2080. arXiv:1606.06357.

[5] T. Trouillon, É. Gaussier, C. R. Dance, G. Bouchard, On inductive abilities of latent factor models for relational learning, Journal of Artificial Intelligence Research 64 (2019) 21–53. doi:10.1613/jair.1.11305. arXiv:1709.05666.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, Curran Associates, Inc., Lake Tahoe, USA, 2013, pp. 2787–2795.

[7] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proceedings of the IEEE 104 (2016) 11–33. doi:10.1109/JPROC.2015.2483592. arXiv:1503.00759.

[8] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2724—-2743. doi:10.1109/TKDE.2017.2754499.

[9] Y. Dai, S. Wang, N. N. Xiong, W. Guo, A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks, Electronics 9 (2020) 1–29. doi:10.3390/electronics9050750.

[10] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, Advances in Neural Information Processing Systems 2018-December (2018) 4284–4295. arXiv:1802.04868.

[11] R. Abboud, İ. İ. Ceylan, T. Lukasiewicz, T. Salvatori, Boxe: A box embedding model for knowledge base completion, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6dbbe6abe5f14af882ff977fc3f35501-Abstract.html.

[12] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling Relational Data with Graph Convolutional Networks, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10843 LNCS (2018) 593–607. doi:10.1007/978-3-319-93417-4_38. arXiv:1703.06103.

[13] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1798–1828. doi:10.1109/TPAMI.2013.50. arXiv:1206.5538.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems

2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 2172–2180. URL: https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html.

[15] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in $\beta$-VAE, in: Advances in Neural Information Processing Systems 30, Nips, Long Beach, CA, USA, 2017. URL: http://arxiv.org/abs/1804.03599. arXiv:1804.03599.

[16] R. T. Q. Chen, X. Li, R. B. Grosse, D. Duvenaud, Isolating Sources of Disentanglement in Variational Autoencoders, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada, 2018, pp. 2615—-2625. arXiv:1802.04942.

[17] K. Ridgeway, M. C. Mozer, Learning Deep Disentangled Embeddings With the F-Statistic Loss, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada, 2018, pp. 185—-194.

[18] C. Eastwood, C. K. I. Williams, A framework for the quantitative evaluation of disentangled representations, in: 6th International Conference on Learning Representations, {ICLR}, Vancouver, BC, Canada, 2018.

[19] A. Kumar, P. Sattigeri, A. Balakrishnan, Variational inference of disentangled latent concepts from unlabeled observations, in: 6th International Conference on Learning Representations, {ICLR}, Vancouver, BC, Canada, 2018. arXiv:1711.00848.

[20] F. Locatello, S. Bauer, M. Lucic, G. R{\"{a}}tsch, S. Gelly, B. Sch{\"{o}}lkopf, O. Bachem, Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, in: Proceedings of the 36th International Conference on Machine Learning,{ICML}, Long Beach, California, USA, 2019, pp. 4114—-4124. arXiv:arXiv:1811.12359v4.

[21] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, M. Tschannen, Weakly-Supervised Disentanglement Without Compromises, CoRR abs/2002.0 (2020). arXiv:2002.02886.

[22] X. Steenbrugge, S. Leroux, T. Verbelen, B. Dhoedt, Improving Generalization for Abstract Reasoning Tasks Using Disentangled Feature Representations, in: Neural Information Processing Systems (NeurIPS) Workshop on Relational Representation Learning, Montreal, Canada, 2018. doi:http://arxiv.org/abs/1811.04784. arXiv:arXiv:1811.04784v1.

[23] S. van Steenkiste, F. Locatello, J. Schmidhuber, O. Bachem, Are Disentangled Representations Helpful for Abstract Visual Reasoning?, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2019, pp. 14222—-14235. arXiv:1905.12506.

[24] T. Karaletsos, S. Belongie, G. Rätsch, When crowds hold privileges: Bayesian unsupervised representation learning with oracle constraints, in: 4th International Conference on Learning Representations, {ICLR}, San Juan, Puerto Rico, 2016, pp. 1–16. arXiv:1506.05011.

[25] J. Chen, K. Batmanghelich, Weakly Supervised Disentanglement by Pairwise Similarities, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI, New York, NY, USA, 2020. arXiv:1906.01044.

[26] J. Chen, K. Batmanghelich, Robust ordinal VAE: employing noisy pairwise comparisons for disentanglement, CoRR abs/1910.05898 (2019). URL: http://arxiv.org/abs/1910.05898. arXiv:1910.05898.

[27] I. Redko, A. Habrard, E. Morvant, M. Sebban, Y. Bennani, Advances in Domain Adaptation

Theory, Elsevier, 2019.
[28] D. P. Kingma, M. Welling, Auto-Encoding Variational Bayes, in: Proceedings of the 2nd International Conference on Learning Representations, Banff, Alberta, Canada, 2014. doi:10.1051/0004-6361/201527329. arXiv:1312.6114.
[29] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: B. Bonet, S. Koenig (Eds.), Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 2181–2187. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571.
[30] M. Nickel, L. Rosasco, T. A. Poggio, Holographic embeddings of knowledge graphs, in: D. Schuurmans, M. P. Wellman (Eds.), Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, AAAI Press, 2016, pp. 1955–1961. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484.
[31] M. Asai, Photo-Realistic Blocksworld Dataset, arXiv preprint arXiv:1812.01818 (2018).
[32] I. Donadello, L. Serafini, A. d'Avila Garcez, Logic Tensor Networks for Semantic Image Interpretation, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 1596—-1602. arXiv:1705.08968.
[33] L. Serafini, A. D. Garcez, Logic tensor networks: Deep learning and logical reasoning from data and knowledge, in: Proceedings of the 11th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy'16) co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence {(HLAI} 2016), New York, NY, USA, 2016. arXiv:1606.04422.
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

## A. BlockStacks dataset description

The *BlockStacks* dataset consists of 12,000 images (200×200 pixels but resized in code to 128×128) of individual block stacks, of varying height (between 1-10 blocks), block colors (uniformly sampled from options: { gray, blue, green, brown, purple, cyan, yellow} ) and position (uniformly sampled from $x, y$ range (-3,-3) to (3,3)), but with the requirement that each instance consists of a single red block at a random height (see Figure 4 for example images). These were rendered using the CLEVR rendering agent with the help of code from [31]. The dataset is divided into 9000:1500:1500 train, validation and test splits.

## B. Explanation of the $\beta$-VAE

The VAE is derived by introducing an approximate posterior $q_\alpha(\boldsymbol{Z}|\boldsymbol{X})$, from which a lower bound (commonly referred to as the Evidence LOwer Bound (ELBO)) on the true marginal log $p_\theta(\boldsymbol{X})$ can be obtained by using Jensen's inequality [28]. The VAE maximises the log-probability by

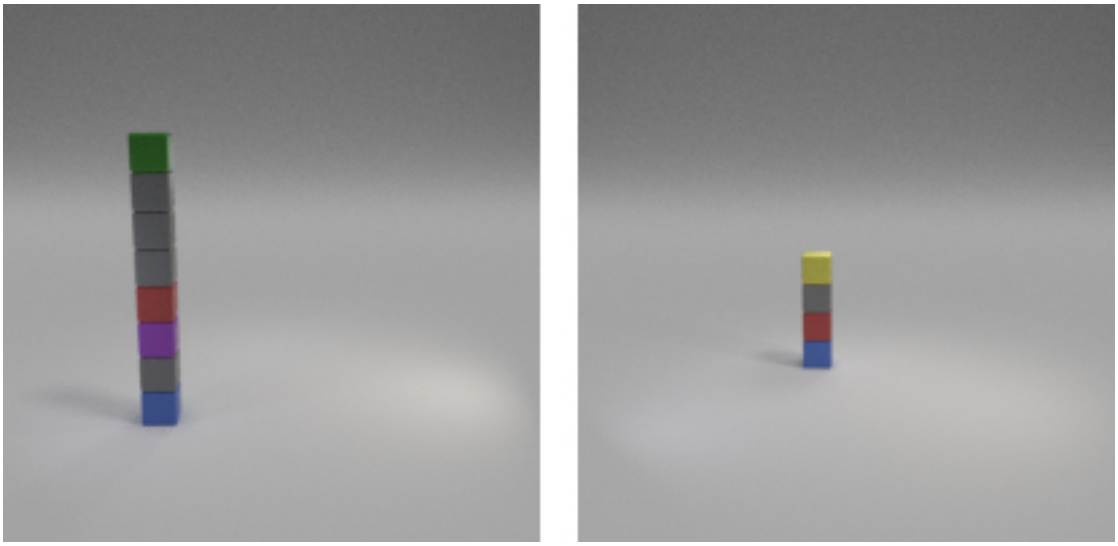

**Figure 4:** Example of two *BlockStacks* data set images.

maximising this lower bound, given by:

$$\mathscr{L}^{ELBO}_{\beta\text{-VAE}} = \mathbb{E}_{q_\alpha(\boldsymbol{Z}|\boldsymbol{X})}[\log p_\theta(\boldsymbol{X}|\boldsymbol{Z})] - \beta D_{KL}(q_\alpha(\boldsymbol{Z}|\boldsymbol{X})\|p_\theta(\boldsymbol{Z})), \tag{8}$$

where $q_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ is the approximate posterior, typically modelled as a neural network encoder with parameters $\alpha$. Similarly $p_\theta(\boldsymbol{X}|\boldsymbol{Z})$ is modelled as a decoder with parameters $\theta$ and is calculated as a Monte Carlo estimation. A reparameterization trick is used to enable differentiation through an otherwise undifferentiable sampling from $q_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ (see [28]). In the $\beta$-VAE [1, 15], an additional $\beta$ scalar hyperparameter was added as it was found to influence disentanglement through stronger distribution matching pressure with respect to the prior $p_\theta(\boldsymbol{Z})$, where this prior is typically set to an isotropic zero-mean Gaussian $\mathcal{N}(\mathbf{0}, \mathbb{1})$). When $\beta = 1$ we obtain the standard VAE objective [28].

## C. Model Descriptions

In this section we provide model details for each relation-decoder that we use and the VAE architecture that we employ for each data set.

### C.1. Relation Decoder implementations

**TransR**:

$$\phi_r^{\text{TransR}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \|\boldsymbol{h}_r + \boldsymbol{r} - \boldsymbol{t}_r\|_2$$

with,

$$\boldsymbol{h}_r = \boldsymbol{M}_r\boldsymbol{z}_i \quad \text{and} \quad \boldsymbol{t}_r = \boldsymbol{M}_r\boldsymbol{z}_j.$$

As we want to obtain a [0,1] output, we modify TransR through $\phi_r^{\text{TransR}^+} = \sigma(c - \phi_r^{\text{TransR}})$, where $\sigma$ is a sigmoid function and c is a scalar that ensures that at $\phi_r^{\text{TransR}^+}(\boldsymbol{z}_i, \boldsymbol{z}_j) = 0$, then $\phi_r^{\text{TransR}^+}(\boldsymbol{z}_i, \boldsymbol{z}_j) \approx 0$. In all experiments we set $c = 10$.
**NTN** (modified version from [32, 33]):

$$\phi_r(\boldsymbol{z}_0, \dots, \boldsymbol{z}_n) = \sigma\big(\boldsymbol{u}_r^\top[\tanh(\boldsymbol{z}^{c\top}\boldsymbol{M}_r\boldsymbol{z}^c + \boldsymbol{V}_r\boldsymbol{z}^c + \boldsymbol{b}_r)]\big) \tag{9}$$

where $\boldsymbol{u}_r \in \mathbb{R}^k$, $\boldsymbol{M}_r \in \mathbb{R}^{(n-1)\cdot d_z \times (n-1)\cdot d_z \times k}$, $\boldsymbol{V}_r \in \mathbb{R}^{k\times(n-1)\cdot d_z)}$ and $\boldsymbol{b}_r \in \mathbb{R}^k$. The only hyperparameter to consider is $k$, which controls the NTN's capacity - in all experiments, we set this to 1. Here $\boldsymbol{z}^c$ is a concatenation of the inputs $\boldsymbol{z}_0, \dots, \boldsymbol{z}_n$, which was introduced in [32, 33]. In contrast the original NTN (see [3]) is only applicable to binary relations and does not include the outer sigmoid.
**HolE**:

$$\phi_r^{\text{HolE}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sigma(\boldsymbol{r}^\top(\boldsymbol{z}_i \star \boldsymbol{z}_j))$$

where $\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denotes the circular correlation operator and is given by,

$$[\boldsymbol{z}_i \star \boldsymbol{z}_j]_k = \sum_{m=0}^{d-1} z_{i,m} z_{j,k+m} \mod d$$

**NN**: a simple four-layer neural-network with hidden layer sizes $l_{\text{in}} = 2d_z, l_1 = 2d_z$ and $l_2 = d_z$, with ReLu activations, for latent representations with size $d_z$. The final output layer, $l_{\text{out}}$, is a single value passed through a sigmoid function, to cap the output within [0,1].

### C.2. VAE configuration

In all representation learning experiments, we use a $\beta$-VAE trained for 300,000 steps, following accepted practice from [20, 22].

The encoder-decoder model parameters are given in Table 1 - we include the model configurations used for both *MNIST* and *BlockStacks* datasets.

### C.3. $\mathscr{L}^{joint}$ configuration

In the source domain, we vary $\beta$ values between $\{1, 4, 8, 12\}$ and fix $\lambda = 10^3$. In the target domain, we fix $\beta$ to $10^{-4}$ and $\lambda = 10^{-2}$ and normalise the $\mathscr{L}^{ELBO}_{\beta\text{-VAE}}$ reconstruction term by dividing by a factor $\frac{1}{\sqrt{H\cdot W\cdot C}}$, for height $H$, width $W$ and color channels $C$, and normalize $\mathscr{L}(\psi^t_{enc}, \mathcal{N}(\mathbf{0}, \mathbb{1}))$ by a factor $\frac{1}{d_z}$, for latent representation size $d_z$.

## D. Supplementary Results

Figure 5 and Figure 6 provide additional results for Con-I (individual consistency scores for individual relation properties covering transitivity, asymmetry and reflexivity) and Con-A, configured on the same data splits as described in the main text. These results cover variants of

**Table 1**
Specification of our $\beta$-VAE encoder and decoder model parameters, for both 28×28 (top) and 128×128 (bottom) size input data. I: Input channels, O: Output channels, K: Kernel size, S: Stride, P: Padding, A: Activation

| **Encoder** |
|---|
| Input: $28 \times 28 \times N_C = 1$ |
| **Layer_ID ; I ; O ; K ; S ; P ; A** |
| Conv2d_1 ; $N_C$ ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_2 ; 32 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_3 ; 32 ; 64 ; $3 \times 3$ ; 2 ; 1 ; ReLU |
| Conv2d_4 ; 64 ; 64 ; $2 \times 2$ ; 2 ; 1 ; ReLU |
| **Layer_ID ; Num Nodes : In - Out ; A** |
| FC_z ; 576 - 144 ; ReLU |
| FC_z_mu ; 144 - 10 ; None |
| FC_z_logvar ; 144 - 10 ; None |

| **Decoder** |
|---|
| Input: $\mathbb{R}^{10}$ |
| **Layer_ID ; Num Nodes : In - Out ; A** |
| FC_z ; 10 - 144 ; ReLU |
| FC_z_mu ; 144 - 576 ; ReLU |
| **Layer_ID ; I ; O ; K ; S ; P ; A** |
| UpConv2d_1 ; 64 ; 64 ; $2 \times 2$ ; 2 ; 1 ; ReLU |
| UpConv2d_2 ; 64 ; 32 ; $3 \times 3$ ; 2 ; 1 ; ReLU |
| UpConv2d_3 ; 32 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| UpConv2d_4 ; 32 ; $N_C$ ; $4 \times 4$ ; 2 ; 1 ; Sigmoid |

| **Encoder** |
|---|
| Input: $128 \times 128 \times N_C = 3$ |
| **Layer_ID ; I ; O ; K ; S ; P ; A** |
| Conv2d_1 ; $N_C$ ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_2 ; 32 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_3 ; 32 ; 64 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_4 ; 32 ; 64 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| Conv2d_5 ; 64 ; 64 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| **Layer_ID ; Num Nodes : In - Out ; A** |
| FC_z ; 1024 - 256 ; ReLU |
| FC_z_mu ; 256 - 10 ; None |
| FC_z_logvar ; 256 - 10 ; None |

| **Decoder** |
|---|
| Input: $\mathbb{R}^{10}$ |
| **Layer_ID ; Num Nodes : In - Out ; A** |
| FC_z ; 10 - 256 ; ReLU |
| FC_z_mu ; 256 - 1024 ; ReLU |
| **Layer_ID ; I ; O ; K ; S ; P ; A** |
| UpConv2d_1 ; 64 ; 64 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| UpConv2d_2 ; 64 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| UpConv2d_3 ; 32 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| UpConv2d_4 ; 32 ; 32 ; $4 \times 4$ ; 2 ; 1 ; ReLU |
| UpConv2d_5 ; 32 ; $N_C$ ; $4 \times 4$ ; 2 ; 1 ; Sigmoid |

the DC and NN models. DC variants include: DC-Basic, uses the same $\phi_r^{\ddagger}$ as DC, but uses a similar $\phi_r^{\dagger}$ to that of [25] but includes the dynamic $\boldsymbol{u}$ mask and $\boldsymbol{b}_{\dagger}$ offset; DC-Gaus, again same $\phi_r^{\ddagger}$ but uses a Gaussian function for $\phi_r^{\dagger}$; DC-Cauchy, uses a Cauchy distribution form for $\phi_r^{\dagger}$ and a Cauchy cumulative distribution function for $\phi_r^{\ddagger}$; and finally DC-CCS which employs a modified Cauchy distribution for $\phi_r^{\dagger}$, via

$$\sigma\Big(\eta(2 \cdot \phi_r^{\text{DC-Cauchy},\dagger} - 1)\Big)$$

where $\sigma$ is the sigmoid function and $\eta$ is a scalar value. This modification enables a cliff-like shape for $\phi_r^{\dagger}$, such that it can output close to 1 for a wider vector difference range. Note all distribution forms are unnormalized so that they cover the interval [0,1].

The NN variants vary layer depth and size, but all use a common input layer of size $l_{\text{in}} = 2 * d_z$. NN2 is a three-layer neural network with hidden layer size $d_z$ and NN3 is a four-layer neural network which is the same as NN, but in contrast has a $d_z$ pre-final layer size, thereby omitting

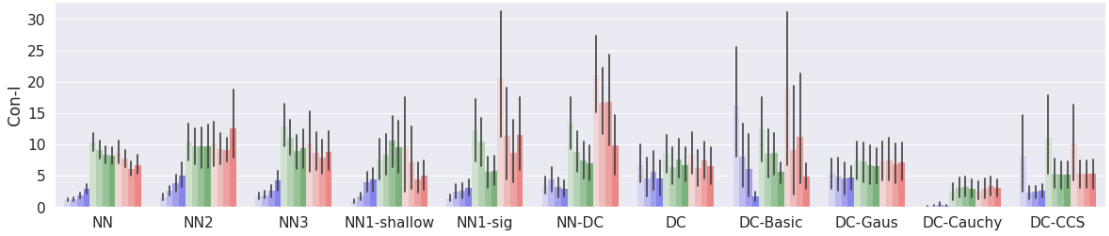


**Figure 5:** Consistency values for individual relation properties (Con-I), covering: transitivity, reflexivity and asymmetry. Values are for variants of DC and NN relation-decoder models, referenced to source (MNIST) domain (lower values better). In all plots, darker color shades denote higher values of $\beta$ (in range $\{1, 4, 8, 12\}$), corresponding to greater disentanglement pressure from the $\beta$-VAE. Blue, green and red groups show results for data-embeddings, interpolation and extrapolation embeddings respectively (see main text for details on these data splits).

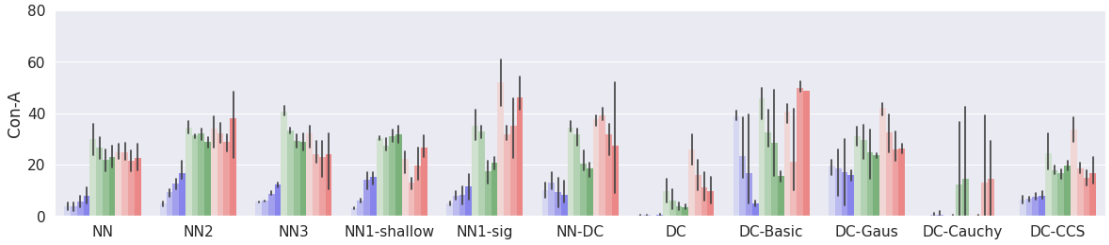


**Figure 6:** Con-A values for variants of DC and NN relation-decoder models, referenced to source (MNIST) domain (lower values better). In all plots, darker color shades denote higher values of $\beta$ (in range $\{1, 4, 8, 12\}$), corresponding to greater disentanglement pressure from the $\beta$-VAE. Blue, green and red groups show results for data-embeddings, interpolation and extrapolation embeddings respectively (see main text for details on these data splits).

the bottleneck dimension reduction of NN. NN1-shallow includes only one hidden layer, like NN2, of size $\frac{d_z(d_z-1)}{2}$ which enables a pairwise comparison between each input dimension. NN1-sig is the same as NN but employs sigmoid activations, instead of ReLUs. NN-DC is the again same as the NN from the main text, but includes an additional $\phi_r^{\dagger}$-type node that can compute relative differences between inputs in the same way as DC.

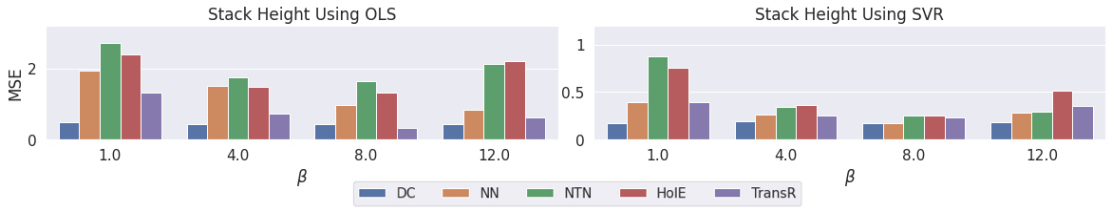


**Figure 7:** Analysis of domain-specific information retention by the $\beta$-VAE when using different relation-decoders for ordinality relation decoding. We attempt to predict the overall BlockStacks stack height on the final fixed embeddings obtained after isSuccessor relation-decoder alignment.

## E. How does each model impact the retention of domain-dependent information

Figure 7 shows results for BlockStacks overall block height prediction accuracy when training on fixed encodings of each block stack, after isSuccessor relation-decoder alignment as been applied. Note $\beta$ is fixed in the target domain, so the only moving part are the pretrained models which are trained with varied source $\beta$ values. Note also that dc has an unfair advantage here, as the steered fitting approach allows more flexibility to the VAE learning phase - for this reason the result is only included in the appendix. Since we are interested in capturing general representations that encode both domain-dependent and -independent information, we use each target encoder $\psi^t_{enc}$ obtained from each PRT experiment and produce encodings for the full *BlockStacks test* set. The resulting encodings are then divided into a new train and test subset, used to train both a *Sci-Kit Learn* Linear regressor and Support Vector Machine regressor with a RBF $^\circ$ kernel [34]. We present the resulting Mean Squared Errors (MSE) in Figure 7, with Ordinary Least Squares (OLS) (a) and Support Vector Regression (SVR) (b).

There are a number of noteworthy details: firstly, DC shows no dependence on $\beta$ and leads to a lower MSE across all settings; second, excluding DC, for all models we observe an optimum MSE at $\beta = 8$, with TransR reaching DC MSE performance for OLS and NN doing the same for SVR. These results indicate that lower MSE can be obtained by using non-linear regression, which indicates that to some degree, the block stack height factor is not encoded linearly, regardless of selected model. Next, by contrasting with Figure 3-bottom, these results suggest that models with higher GC lead to embeddings that are more amenable to domain-specific factor prediction. However, the parabolic trend, where increasing $\beta$ to 12 leads to an increase in error, is in agreement with Figure 2-bottom-right, which showed that most models do not improve at PRT for the largest $\beta$.