Quantised Transforming Auto-Encoders: Achieving Equivariance to Arbitrary Transformations in Deep Networks (Supplementary Material)

Jianbo Jiao
jianbo@robots.ox.ac.uk
João F. Henriques
joao@robots.ox.ac.uk

Visual Geometry Group
University of Oxford

In this supplementary material we provide more implementation details of the proposed quantised transforming auto-encoders (Section 1). We also include additional qualitative results on the RGBD-Object dataset [2] with comparison to alternative solutions (Section 2), qualitative performance with the additive-space solution (Section 3), qualitative performance for the out-of-distribution (compositional) extrapolation (Section 4), additional qualitative performance on the datasets mentioned in the main paper (Section 5), and also some dynamic results (Section 6) as attached videos.

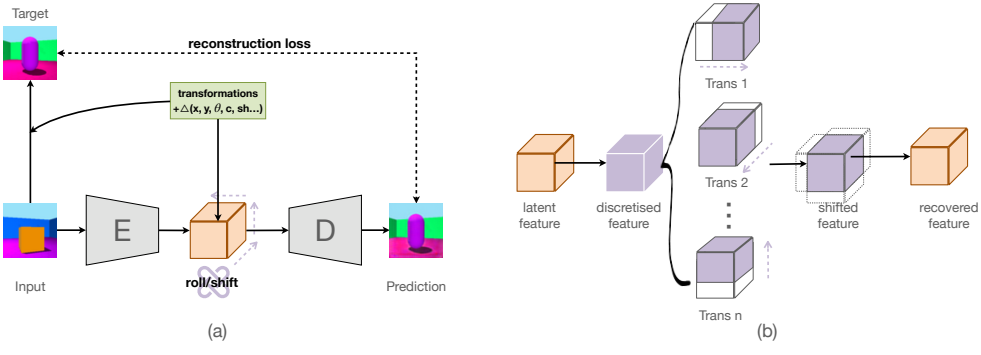

Figure 1: (a) The basic framework of the proposed approach; (b) Illustration of the discretised rolling embeddings. For simplicity, here the rolling embeddings are illustrated by a 3-dimensional tensor, while in practice they can be of higher dimensions.

1 Network Architecture Details

In this section, we provide more details of the network design and the pipeline of the proposed approaches. The network architecture of the proposed Quantised Transforming Auto-Encoders (QTAE) is shown in Figure 1, from which we can see that the input image is

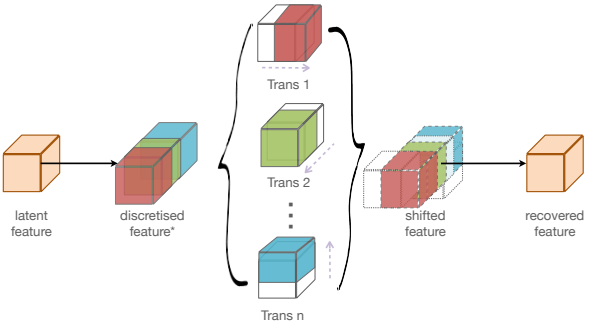

Figure 2: Illustration of the efficient additive-space shifting, for large transformation spaces. The quantity marked with an asterisk indicates the additive product combination.

re-rendered according to the transformations that apply to the embeddings (orange block in Figure 1 (a)). The overall architecture follows an encoder-decoder fashion, and the network is trained with the reconstruction objective. We use simple convolutional-deconvolutional auto-encoders with several convolutional blocks in the encoder, consisting of a 3×3 or 5×5 convolution, max-pooling (with stride 2) and ReLU activation. The number of channels is 16-32-64-64. As for the deconvolutional layers (transposed convolutions), they follow the same layout but without max-pooling.

The rolling embeddings are illustrated in Figure 1 (b), where we can see that the input embedding (orange block) is first discretised to an orthogonal representation followed by the rolling operations (which can be interpreted as a tensor product space of the dimensions corresponding to the different transformations). Note that for simplicity, the discretised representation is showcased as three-dimensional, while it can be higher dimensional depending on the transformation space. The rolled (or shifted) embeddings are recovered to the original three-dimensional feature space, and finally fed into the decoder.

As mentioned in the main paper, in order to address the memory consumption issue cased by the increasing transformation space, we propose an efficient combination of transformations in an additive tensor space (as opposed a the tensor product space). An illustration of this efficient solution is shown in Figure 2, where the input embedding is discretised by unstacking independent matrices, each corresponding to a different transformation. The embeddings after rolling/shifting are stacked to the original feature dimension and fed back to the decoder to re-render the result.

2 Qualitative Performance on RGBD-Object

Due to the space limitations, we did not include qualitative performance on the RGBD-Object [2] in the main paper. Here, we show a qualitative comparison between our approach and other alternative solutions in Figure 3. From the results we can see that our method performs much better than the compared methods, though the re-rendered results for all the methods are less detailed than on other datasets. This is mainly cause by the low diversity of available views for each object in the RGBD-Object dataset and the complexity of the appearance. In addition, as mentioned in the main paper, the object instances in the re-rendered test images are never seen in the training set, though the same category might

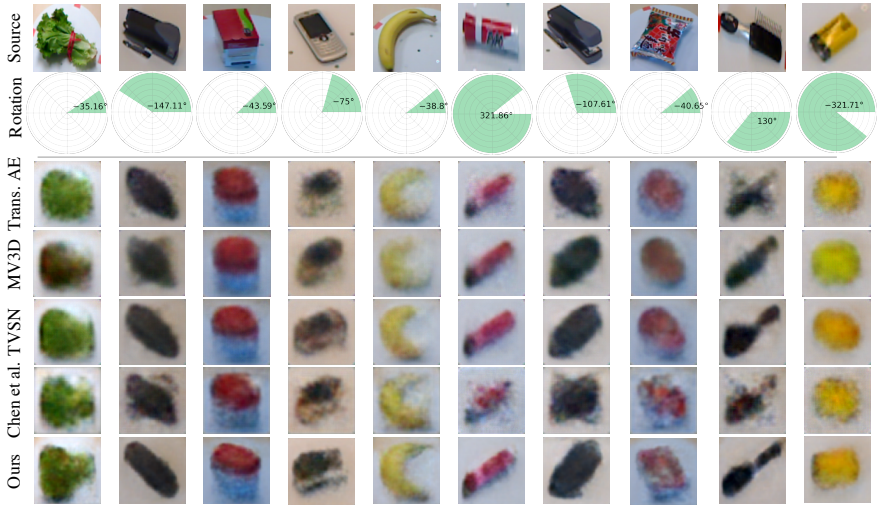

Figure 3: Qualitative performance on the RGBD-Object dataset, with comparison to state-of-the-art methods. The *Source* images are re-rendered with the given angles in the second row. Note that this dataset has very few distinct object instances (300), so it is a difficult task for deep networks to fit it.

appear. This further increases the difficulty of this challenging task, as accurate filters with the particular patterns of these test-set images cannot be learned by the decoder.

3 Qualitative Performance of Additive vs. Product-space

In addition to the quantitative results presented in Table 2 in the main paper, here we showcase the qualitative performance of the efficient additive-space solution, as shown in Figure 4. We can see that despite having much fewer parameters and memory consumption, the efficient solution does not result in a significant performance drop visually, and even shows better quality for some cases (*e.g.* the third last sample) in the 3D Shapes dataset [].

4 Qualitative Performance for Out-of-distribution Extrapolation

In this section, we show the qualitative performance of our approach on compositionality and out-of-distribution extrapolation, as described in Section 4.4 in the main paper. Following the main paper, here we show the three concepts (*i.e.* pairs of attributes) that were never observed during the network training: *blue sphere, large cylinder*, and *cube on red wall*, in Figure 5. We can see that although the above-mentioned concepts are never seen by the model during training, our approach is able to compose and extrapolate outside the training distribution and the re-rendered results are of good quality. This again validates the effectiveness of the proposed method.

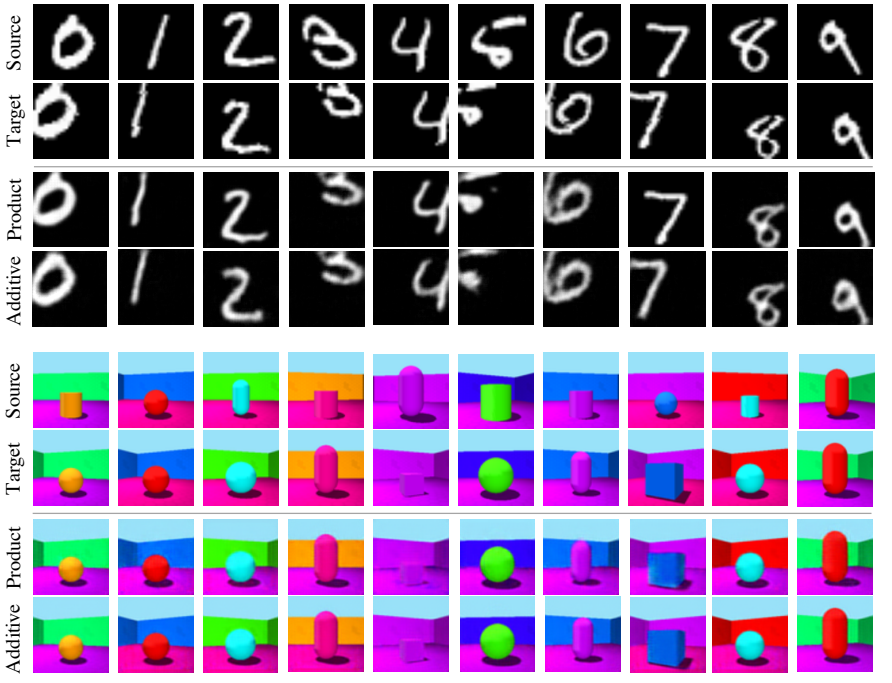

Figure 4: Qualitative performance comparison between the product-space, and the additive-space (more efficient).

5 Additional Qualitative Results

Here in this supplementary material, we provide additional qualitative results in Figure 6, Figure 7, Figure 8, and Figure 9, for the datasets used in the main paper.

6 Dynamic Video Performance

In addition to the results shown in the main paper, in the supplementary material we further present dynamic results in videos showing more transitions. Please refer to the video files "NORB.m4v" and "3DShape.m4v" for more details.

References

[1] Chris Burgess and Hyunjik Kim. 3D Shapes Dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[2] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

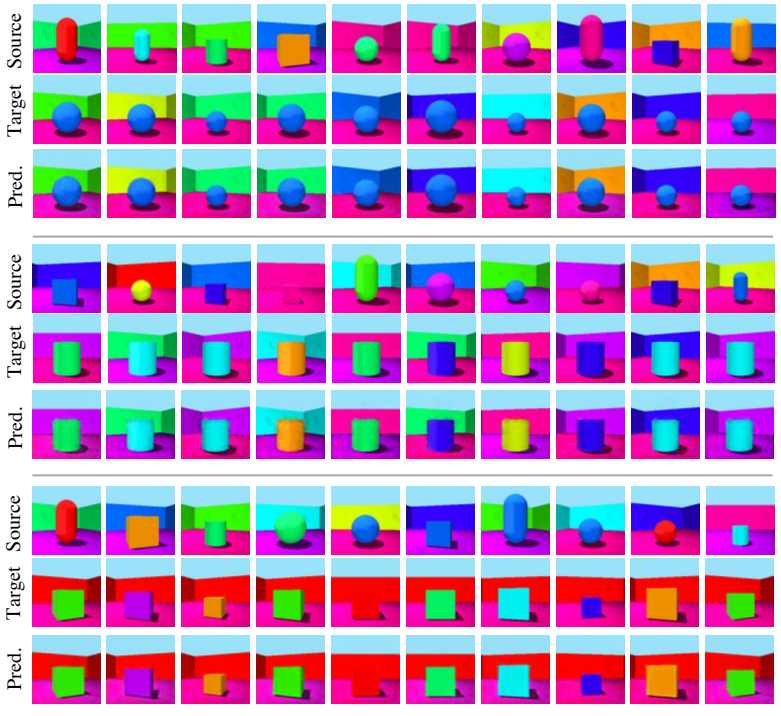

Figure 5: Qualitative performance for the out-of-distribution extrapolation. From top to bottom are the different settings of unseen concepts: blue sphere, large cylinder, and cube on red wall.

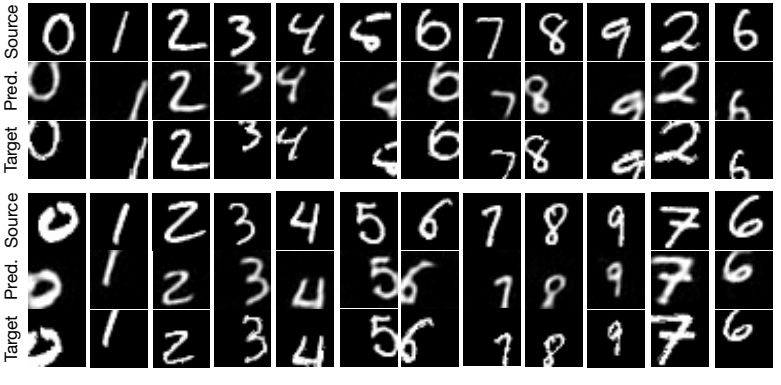

Figure 6: Additional results for the proposed approach on MNIST [13] data.

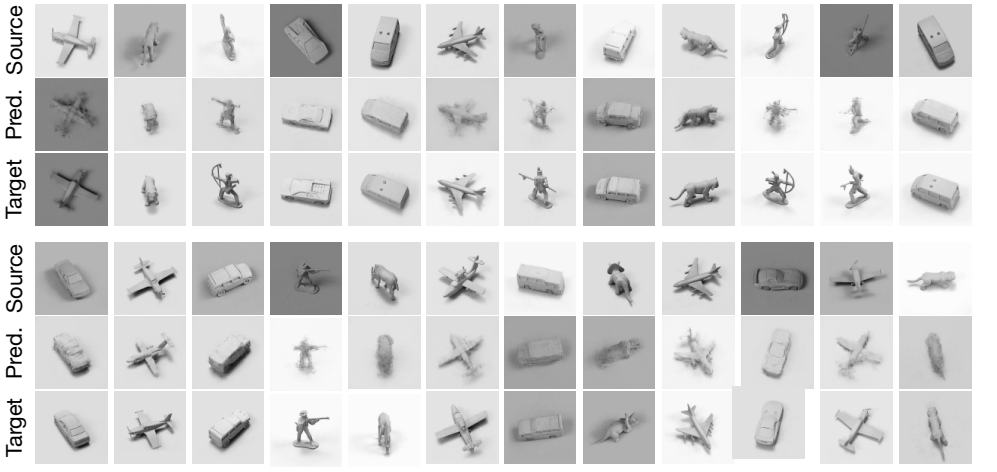

Figure 7: Additional results for the proposed approach on SmallNORB [4] data.

[4] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.

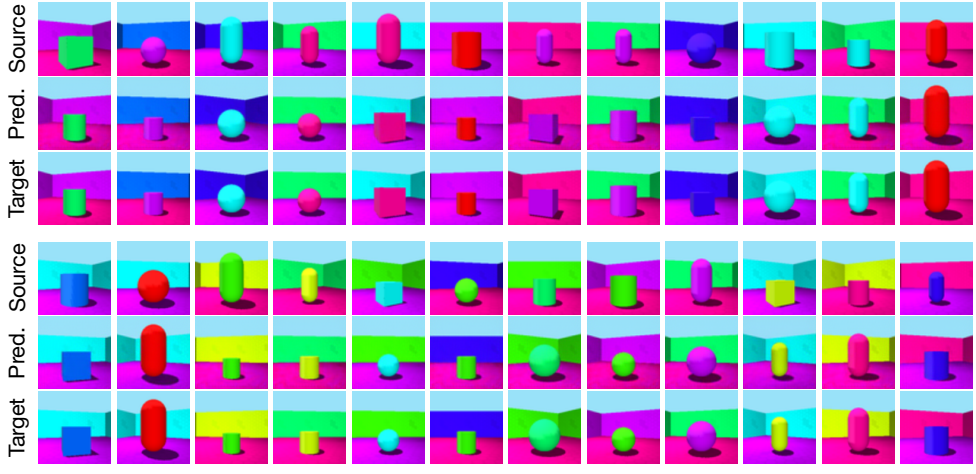

Figure 8: Additional results for the proposed approach on 3DShape [] data.

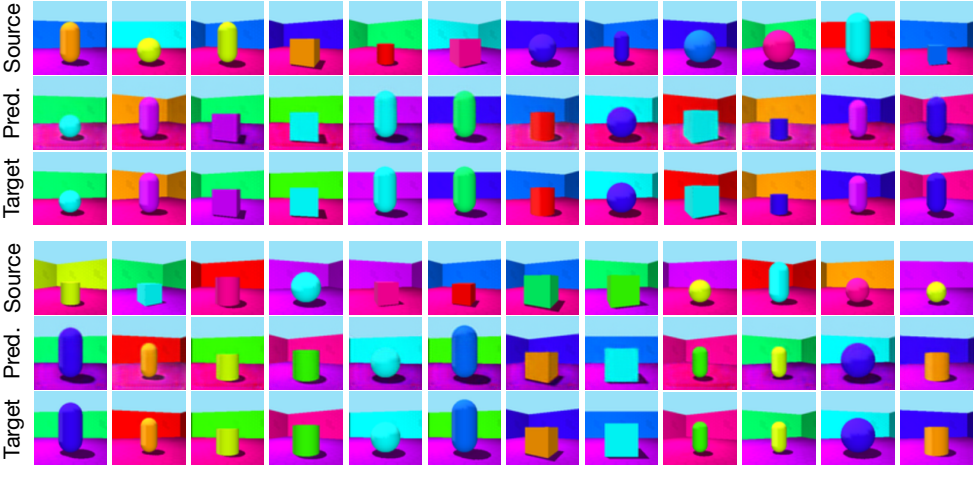

Figure 9: Additional results for the proposed approach on 3DShape [] data when considering the colour transformations.