Supplementary Materials
CTrGAN: Cycle Transformers GAN for Gait Transfer

Shahar Mahpod Noam Gaash Hay Hoffman Gil Ben-Artzi
Ariel University
Ariel , Israel
http://gil-ba.com

A. Attention

A.1. Self and Cross-Attention

The goal of the attention layer is to discover relationships between a given query Q (e.g. an image) and pre-exist key data K (e.g. a set of images) and to represent these relationships using V. It is stated as follows:

$$Attention(Q,K,V) = Softmax(\frac{Q, K^T}{\sqrt{d_k}})V, \quad (1)$$

where d_k is a scaling factor [3].

Denote $\mathcal{A}(Q,K,V)$ as the attention block (Eq. 1, see [3] for additional details) and denote $\mathcal{F}_S$ and $\mathcal{F}_T$ as the feature encoders of the source and target domains, respectively. The self attention of the Transformers encoders $\mathcal{E}_S$ and $\mathcal{E}_T$ and decoders $\mathcal{D}_S$ and $\mathcal{D}_T$ is:

$$\begin{aligned}
\mathcal{E}_S\left(\mathbf{K}^t\right) &= \mathcal{A}\left(\mathcal{F}_T\left(\mathbf{K}^t\right), \mathcal{F}_T\left(\mathbf{K}^t\right), \mathcal{F}_T\left(\mathbf{K}^t\right)\right), \\
\mathcal{D}_S\left(\mathbf{P}^{s_i}\right) &= \mathcal{A}\left(\mathcal{F}_S\left(\mathbf{P}^{s_i}\right), \mathcal{F}_S\left(\mathbf{P}^{s_i}\right), \mathcal{F}_S\left(\mathbf{P}^{s_i}\right)\right), \\
\mathcal{E}_T\left(\mathbf{K}^{s_i}\right) &= \mathcal{A}\left(\mathcal{F}_T\left(\mathbf{K}^{s_i}\right), \mathcal{F}_T\left(\mathbf{K}^{s_i}\right), \mathcal{F}_T\left(\mathbf{K}^{s_i}\right)\right), \\
\mathcal{D}_T\left(\mathbf{P}^t\right) &= \mathcal{A}\left(\mathcal{F}_S\left(\mathbf{P}^t\right), \mathcal{F}_S\left(\mathbf{P}^t\right), \mathcal{F}_S\left(\mathbf{P}^t\right)\right).
\end{aligned}$$

Note that the target's Keys $\mathbf{K}^t$ Transformer encoders ($\mathcal{E}_S$) remain unchanged throughout the training process whereas the source's keys $\mathbf{K}^{s_i}$ (at $\mathcal{E}_T$) are updated with accordance of the specific source that is currently being used for the cyclic training.

The generator of the source and target are cross attention operations:

$$\mathcal{G}_{s\to t}\left(\mathbf{K}^t, \mathbf{P}^{s_i}\right) = \mathcal{H}_S\left(\mathcal{A}\left(\mathcal{D}_S\left(\mathbf{P}^{s_i}\right), \mathcal{E}_S\left(\mathbf{K}^t\right), \mathcal{E}_S\left(\mathbf{K}^t\right)\right)\right) \quad (2)$$

$$\mathcal{G}_{t\to s}\left(\mathbf{K}^{s_i}, \mathbf{P}^t\right) = \mathcal{H}_T\left(\mathcal{A}\left(\mathcal{D}_T(\mathbf{P}^t), \mathcal{E}_T\left(\mathbf{K}^{s_i}\right), \mathcal{E}_T\left(\mathbf{K}^{s_i}\right)\right)\right), \quad (3)$$

where $\mathcal{H}_S$ and $\mathcal{H}_T$ are the features decoders of $\mathcal{G}_{s\to t}$ and $\mathcal{G}_{t\to s}$, respectively.

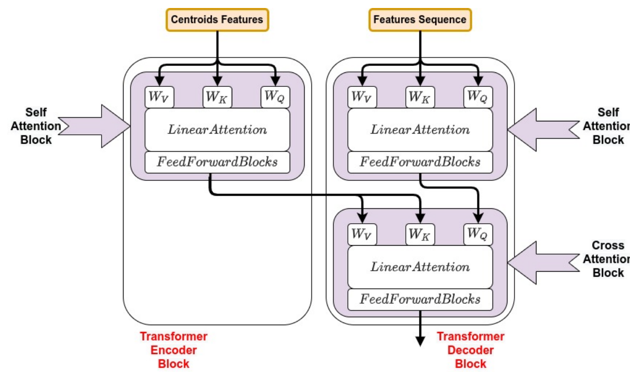


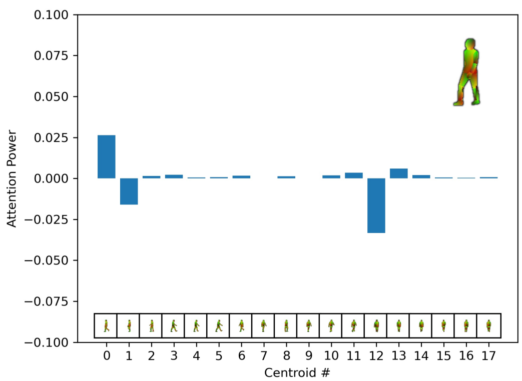
Figure 1: Attention Layers. Left side - Transformer encoder. Right side - Transformer decoder. The encoder performs self-attention while the decoder performs self-attention followed by cross attention with the encoder's outputs.

Figure 1 illustrates the process of attention between the Transformer encoder and decoder. In order to reduce processing power, we use linear attention [1, 2] instead of Transformer's original attention. Keys in the Transformer encoder undergo a process of attention among themselves and the output is served as the Keys and Values with the decoder's queries. A self-attention process takes place between the decoder queries Q and themselves, which reveals temporal relationships between consecutive images. In our model, the Transformer decoder includes both self- and cross- attention blocks. We tested our model with and without using self-attention in the decoder and reported the results for both.

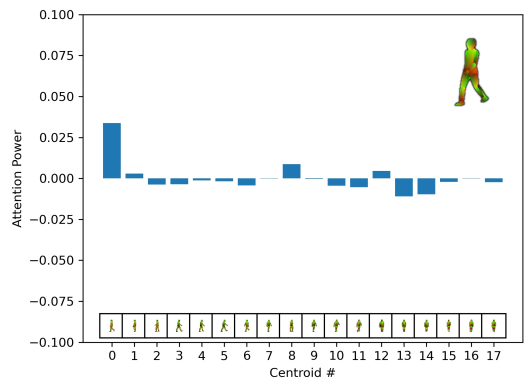
A.2. Attention histograms

Figure 2 shows a series of cross-attention histograms. The graphs show the value of attention between each key and the current image over time. The mean value of the attention of each key was subtracted separately. The attention shown corresponds to the first cross-attention block between the encoder and decoder.

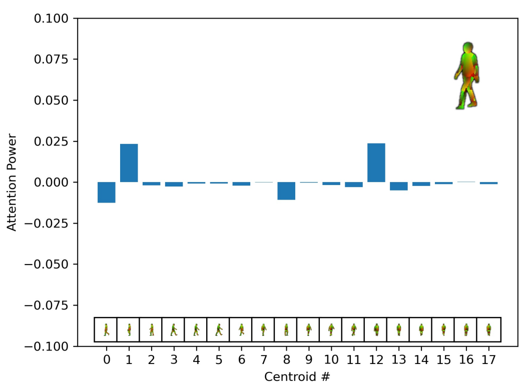
It is feasible to gain some insight into the attention mechanism's functioning. For example, we can see how Key 0 and Key 1's cross-attention evolved over time in relation to the position of the legs. In Figure 2.a, where the right leg is forward, and the left leg is backward, Key 0 makes a considerable contribution, whereas Key 1 makes a much smaller contribution. As the legs swap positions, with the left leg moving forward and the right leg moving backward (Figure 2.b,c,d), Key 0 and Key 1 switch roles as well, with Key 1 now contributing substantially and Key 0 contributing much less.



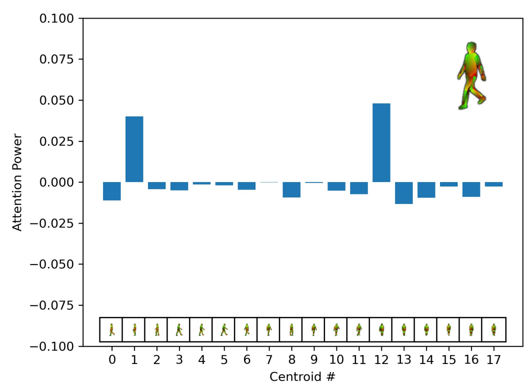
(a)



(b)



(c)



(d)

Figure 2: Attention histograms.

B. Ablation study

We evaluate the impact of the different components of the CTrGAN architecture on recognition accuracy (Table 1,2) and appearance quality (Table 3). V2V is the pose to appearance model.

- **Cycle Only** - refers to using only CycleGAN model.
- **Attention** - adds encoder self-attention and cross-attention between the image sequence and the keys using the decoder.
- **Time-Attention** - adds decoder self-attention which takes advantage of the temporal relations between the three frames within the sequence.
- **Time-Attention-5** - same as Time-Attention but increases the length of the temporal relation to five.
- **Target-Training** - same as Time-Attention but include the target in the training set.

Model	Features			Accuracy ↑		
	Attention mechanism	Encoder self-attention	Decoder self-attention	GaitPart	GaitSet	GaitGL
Cycle Only	✗	✗	✗	5.56	5.00	5.28
+ Attention	✓	✓	✗	78.06	51.39	69.17
+ Time-Attention	✓	✓	✓	84.72	56.67	68.06
Time-Attention-5	✓	✓	✓	79.17	55.83	66.11
Target-Training	✓	✓	✓	87.78	73.89	77.50

Table 1: Target recognition accuracy.

Model	Features			Accuracy ↓		
	Attention mechanism	Encoder self-attention	Decoder self-attention	GaitPart	GaitSet	GaitGL
Cycle Only	✗	✗	✗	97.22	86.11	95.83
+ Attention	✓	✓	✗	13.06	15.56	25.00
+ Time-Attention	✓	✓	✓	10.56	13.89	21.67
Time-Attention-5	✓	✓	✓	12.50	12.78	23.61
Target-Training	✓	✓	✓	8.89	10.00	15.83

Table 2: Source recognition accuracy.

Model	Features			Metrics			
	Attention mechanism	Encoder self-attention	Decoder self-attention	SSIM↑	LPIPS↓	FID↓	IS↓
Cycle Only	✗	✗	✗	0.9030	0.0593	47.3042	0.0011
+ Attention	✓	✓	✗	0.9084	0.0554	58.0742	0.0009
+ Time-Attention	✓	✓	✓	0.9093	0.0549	52.8933	0.0009
Time-Attention-5	✓	✓	✓	0.9093	0.0554	58.0353	0.0013
Target-Training	✓	✓	✓	0.9096	0.0538	51.7293	0.0011

Table 3: Appearance quality.

C. Sample results

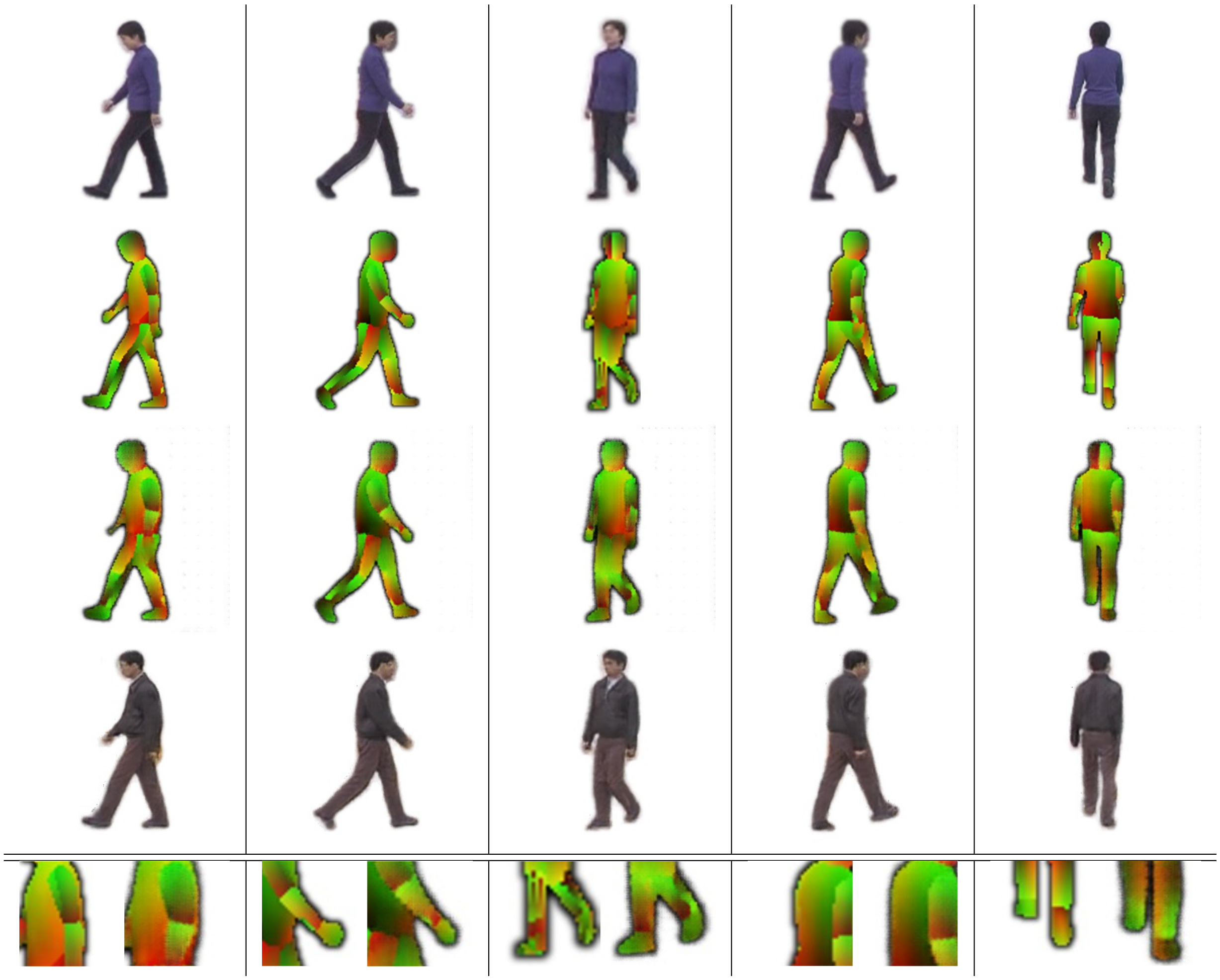


Figure 3: One of the sources is shown in the first line from the top. On the second line are the poses of the source. On the third line are the corresponding generated poses by CTrGAN. On the fourth line are the generated appearances of the target using V2V. The fifth line shows the differences between the source pose image (left image) and the target pose image (right image).

D. Motion transfer mimics the gait of the source.

Here, we examine the extent to which existing motion transfer methods retain the gait pattern of the source. Table 4 presents the source-accuracy. It is defined as the percentage of times the gait recognition model has identified the generated gait as the source's gait. The top of the table shows that the models correctly identified the sources after generating the target's appearance in about 87 percent of the test cases, illustrating the challenge of existing motion transfer methods to generate gait distinguishable from the source. The bottom of the table shows that, with CTrGAN, the gait recognition models' success rate drops significantly to approximately 15 percent of cases.

Method	Model	GaitPart	GaitSet	GaitGL
-	EDN	17.22	20.28	47.78
	V2V	93.06	76.94	93.33
ours	EDN	**7.78**	**7.22**	**12.78**
	V2V	9.44	15.00	20.56

Table 4: The source-accuracy $\downarrow$.

E. Detailed Implementation

E.1. CTrGAN internals

Both $\mathcal{G}_{t \to s}$ and $\mathcal{G}_{t \to s}$ have the exact same architecture. The input is a 256x256x4 image, the output of encoders $\mathcal{F}_T$ and $\mathcal{F}_S$ is a 16x16x256 tensor, and the output of the decoders $\mathcal{H}_T$ and $\mathcal{H}_S$ is an image of the same size (256x256x4).

The Transformers inputs from $\mathcal{F}_T$ and $\mathcal{F}_S$ are of size 16x16x256. We use spatial max pooling to reduce the tensor size to 4x4x256, which results in a flattened feature vector of 4096. The base dimension for the vectors in our Transformers is 1024. Therefore, the fully connected *FC* modules for Q, K, and V embed from 4096 to 1024 and vice versa. The Transformer encoders TE_S and TE_T receive 18 fixed key images (used as Q, K, and V) and is composed of two blocks of self-attention, while the Transformer decoders TD_S and TD_T consist of two blocks of one self-attention followed by one cross-attention block.

E.2. Discriminators structure

In this section, we describe (Table 5) the structure of the model that served as a discriminator block in CTrGAN.

Type	In	Out	Kernel	Stride	Activation
Conv	4	64	4	2	leaky relu(0.2)
Conv	64	128	4	2	instance norm + leaky relu(0.2)
Conv	128	256	4	2	instance norm + leaky relu(0.2)
Conv	256	512	4	1	instance norm + leaky relu(0.2)
Conv	512	1	4	1	-

Table 5: Basic discriminator

E.3. Features encoder structure

In this section, we describe (Table 6) the structure of the model that served as the features encoder blocks ($\mathcal{F}_T$ and $\mathcal{F}_S$) in CTrGAN. The $\mathcal{H}_T$ and $\mathcal{H}_S$ decoders are identical to the $\mathcal{F}_T$ encoder, except that they operate in the other direction

Type	In	Out	Kernel	Stride	Activation
Conv	4	16	3	1	norm+relu
Conv	16	32	3	2	norm+relu
Conv	32	64	3	2	norm+relu
Conv	64	128	3	2	norm+relu
Conv	128	256	3	2	norm+relu
ResBlock	256	256	3	1	norm+relu
	256	256	3	1	norm
ResBlock	256	256	3	1	norm+relu
	256	256	3	1	norm
ResBlock	256	256	3	1	norm+relu
	256	256	3	1	norm

Table 6: Feature encoders F_T and F_S structure

References

[1] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR, 2020.

[2] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8922–8931. Computer Vision Foundation / IEEE, 2021.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.