Supplementary Material for:

ImaGINator: Conditional Spatio-Temporal GAN for Video Generation

Appendix

This Appendix is organized as follows. We firstly present details on the architecture of our proposed ImaGINator in Appendix A. Secondly in Appendix B, we report numerous results of additional experiments related to

B.1. the effectiveness of both *reconstruction loss* and *adversarial loss* utilized in our loss function,

B.2. the contribution of proposed *spatio-temporal fusion*, illustrated by deactivating it in different layers of the Decoder,

B.3. the performance of ImaGINator, as well as state-of-the-art algorithms on 6 datasets including 3 facial expression datasets MUG [1], UvA-NEMO [2], BU-4DFE [3], 2 action datasets NATOPS [4] and Weizmann [5] and BAIR robot push [6]. Evaluation is performed by 3 evaluation metrics.

Finally, we disclose in Appendix C numerous example frames of generated video sequences.

In summary, based on results in the main paper and supplementary material, our approach significantly outperforms existing video generation/prediction methods on all 6 tested datasets and evaluation metrics.

A. Network Architecture

We proceed to introduce details of our model in this section.

A.1. Generator

We proceed to describe the network architecture of the Generator, illustrated in Figure 1. It consists of two parts, (a) an image Encoder, containing five 2D convolutional layers (*Conv1* - *Conv5*) and (b) a video Decoder with five transposed (1+2)D convolutions (*Deconv 6-1* - *Deconv10-2*). Each transposed (1+2)D convolution has two separate and successive operations, M transposed 1D temporal convolutional filters followed by a transposed 2D spatial convolution. In all layers of the Generator, we use the Batch Normalization [7], followed by the *LeakyReLU* after each convolution and transposed convolution, except for the last layer, where we directly use the *Tanh* activation function after the transposed convolution.

Towards generating a video, the Encoder firstly encodes an input image of size $64 \times 64 \times 3$ into a latent vector of size 100, proceeds to combine it with a noise vector of size 512, as well as with a one-hot category vector towards formulating a representation of video in a latent space. Then, the Decoder generates a video based on this representation. Each transposed 1D convolutional layer (except *Deconv6-1*) in the Decoder merges three different types of feature maps as input through *spatio-temporal fusion*, (i) a motion map from its last 2D layer, (ii) an appearance map from the corresponding layer in the Encoder through skip connections, as well as (iii) a one-hot category map replicated from the one-hot category vector. All feature maps share the same spatial size. In particular, we capture the feature maps from layers *Conv1*, *Conv2*, *Conv3*, *Conv4*, in order to fuse with the outputs from layers *Deconv9-2*, *Deconv8-2*, *Deconv7-2* and *Deconv6-2*, respectively. Details of the Generator are exhibited in Table 1.

A.2. Discriminators

Our ImaGINator includes two Discriminators, an *image* Discriminator D_I, as well as a *video* Discriminator D_V. The input of D_I entails N randomly sampled frames, either from real or generated videos. In our experiments, we set $N = 16$. D_I provides as output a scalar value, indicating whether the frames are real or fake. D_I is represented by a network of five 2D convolutional layers. The kernel size in all layers is 4×4, see Figure 2.

D_V discriminates videos based on the related realistic appearance and motion. It is represented by a network containing five 3D convolutional layers, see Figure 3. While $4 \times 4 \times 4$ kernels have been applied in the first four layers, one $2 \times 4 \times 4$ kernel is featured in the last layer ($T \times H \times W$ denotes time step, height and width of a kernel respectively). A one-hot category vector is replicated into a one-hot category map of the same spatial size of the output feature map of *Conv1*. Then, *Conv2* takes the concatenation of

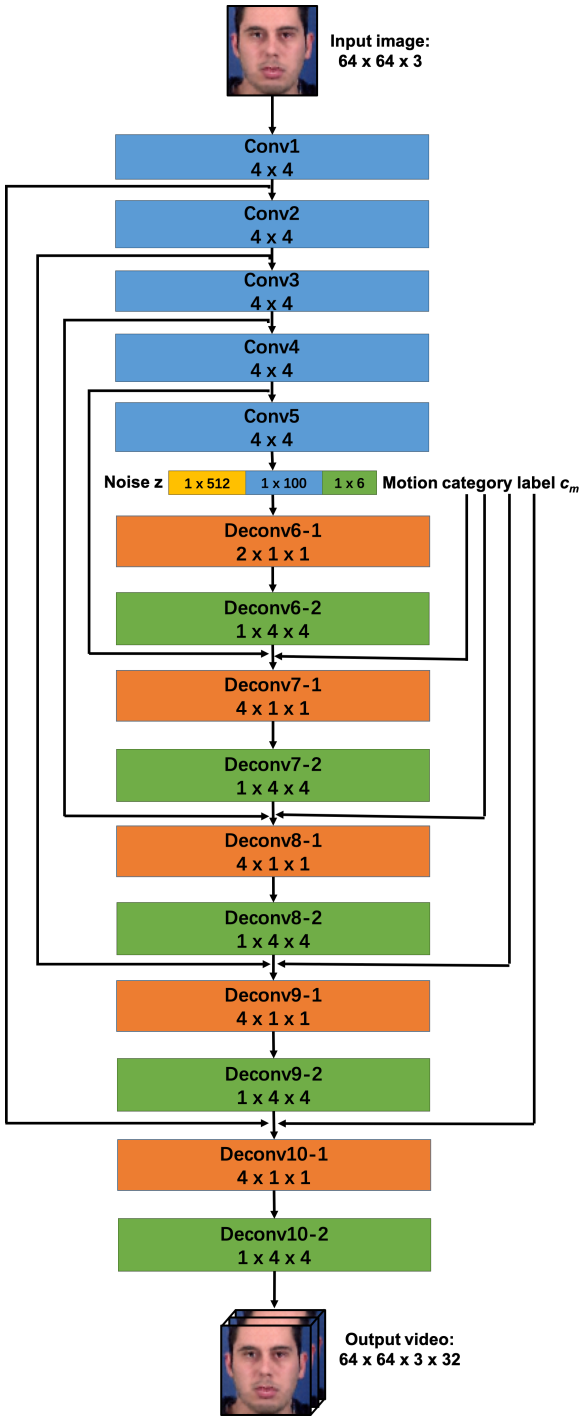

Figure 1: **Network architecture of the Generator.** Our Generator G accepts an image of size $64 \times 64 \times 3$ as input and generates a 32-frame long video. G incorporates an image Encoder (*Conv1* - *Conv5*) and a video Decoder (*Deconv6-1* - *Deconv10-2*). Skip connections link Encoder and Decoder, with the goal of enforcing the Decoder to reuse appearance features directly. A motion category vector is replicated into feature maps and concatenated with each feature map in the Decoder (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset).

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv2D	64	4x4	2x2	1x1
Conv2	Conv2D	128	4x4	2x2	1x1
Conv3	Conv2D	256	4x4	2x2	1x1
Conv4	Conv2D	512	4x4	2x2	1x1
Conv5	Conv2D	100	4x4	1x1	No
Deconv6-1	TransConv1D	4096	2x1x1	1x1x1	No
Deconv6-2	TransConv2D	512	1x4x4	1x1x1	No
Deconv7-1	TransConv1D	3072	4x1x1	2x1x1	1x0x0
Deconv7-2	TransConv2D	256	1x4x4	1x2x2	0x1x1
Deconv8-1	TransConv1D	1536	4x1x1	2x1x1	1x0x0
Deconv8-2	TransConv2D	128	1x4x4	1x2x2	0x1x1
Deconv9-1	TransConv1D	768	4x1x1	2x1x1	1x0x0
Deconv9-2	TransConv2D	64	1x4x4	1x2x2	0x1x1
Deconv10-1	TransConv1D	36	4x1x1	2x1x1	1x0x0
Deconv10-2	TransConv2D	3	1x4x4	1x2x2	0x1x1

Table 1: **Network architecture of the Generator.** Our Generator incorporates an image Encoder (*Conv1* - *Conv5*), as well as a video Decoder (*Deconv6-1* - *Deconv10-2*). KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

both feature maps as input.

In all layers in both Discriminators, we use the Spectral Normalization (SN) [8], followed by the *LeakyReLU* after each convolution, except for the the last layer, where we use *Sigmoid* activation function after the normalization. Details pertained to the network architecture of the Discriminators are presented in Table 2 (image Discriminator) and Table 3 (video Discriminator), respectively.

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv2D	64	4x4	2x2	1x1
Conv2	Conv2D	128	4x4	2x2	1x1
Conv3	Conv2D	256	4x4	2x2	1x1
Conv4	Conv2D	512	4x4	2x2	1x1
Conv5	Conv2D	1	4x4	1x1	No

Table 2: **Network architecture of the image Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

Figure 2: **Network architecture of the image Discriminator.** It contains five 2D convolutional layers of kernel size 4×4.

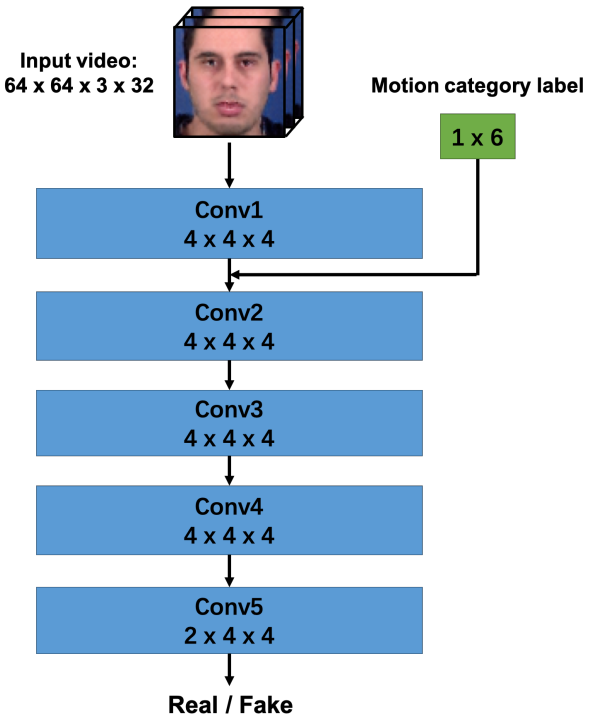

Figure 3: **Network architecture of the video Discriminator.** It includes five 3D convolutional layers, a motion category vector is firstly replicated and then concatenated with the feature map of the first layer (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset).

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv3D	64	4x4x4	2x2x2	1x1x1
Conv2	Conv3D	128	4x4x4	2x2x2	1x1x1
Conv3	Conv3D	256	4x4x4	2x2x2	1x1x1
Conv4	Conv3D	512	4x4x4	2x2x2	1x1x1
Conv5	Conv3D	1	2x4x4	1x1x1	No

Table 3: **Network architecture of the video Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

	Adv. Loss	**Recon. Loss**	**Two losses**
MUG	35.62	45.43	**29.02**
NATOPS	33.97	61.32	**26.86**
Weizmann	150.48	217.58	**99.80**
UvA-NEMO	19.29	30.72	**16.16**

Table 4: **Evaluation results for models using different losses** on four datasets represented by video FID. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

B. Additional Experimental Results

We here resume experiments, which support our choice in architecture-design for ImaGINator. Specifically in B.1, we conduct an ablation study to analyze the pertinence of the joint use of *adversarial loss* and *reconstruction loss* in our loss function. Subsequently, in B.2 we conduct experiments, showcasing the impact of the proposed *spatio-temporal fusion* on effectively decomposing motion and appearance. Finally, in B.3 we demonstrate experimental results on an additional facial expression dataset, BU-4DFE, again comparing our method with two state-of-art methods.

B.1. Reconstruction loss vs. Adversarial loss

Towards evaluating the pertinence of both components in our loss function, we conduct two experiments. While the first experiment integrates merely the *adversarial loss* in ImaGINator, omitting the reconstruction loss; the second experiment merely integrates *reconstruction loss*, omitting the adversarial loss. Generated videos are evaluated based on video FID, SSIM, as well as PSNR for the datasets MUG, NATOPS, Weizmann and UvA-NEMO.

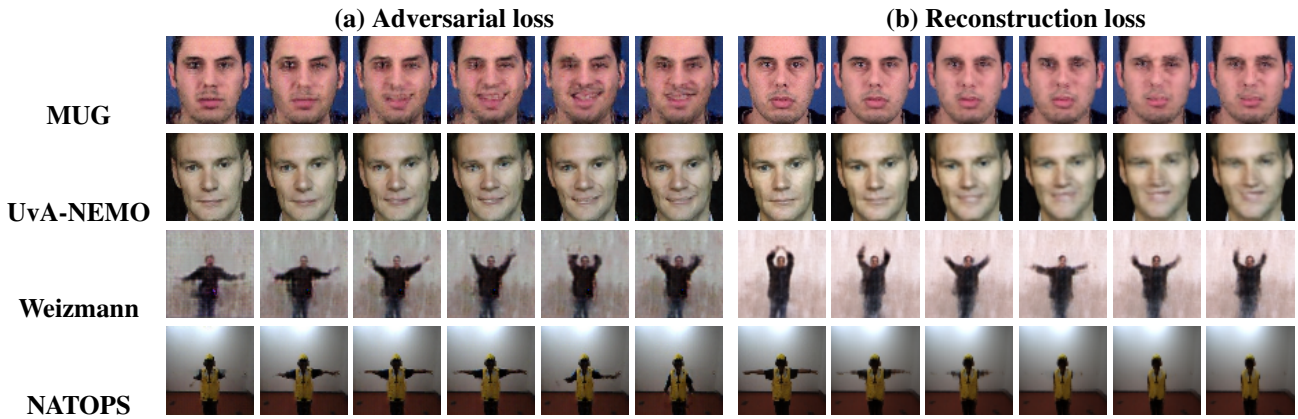

Figure 4: **Comparison of use of merely (a) Adversarial loss and (b) Reconstruction loss.** We illustrate generated frames for (a) and (b) on four datasets. We observe that frames in (a) are sharper than (b), but (b) retains overall structures better than (a). Frames are sampled with time step 4.

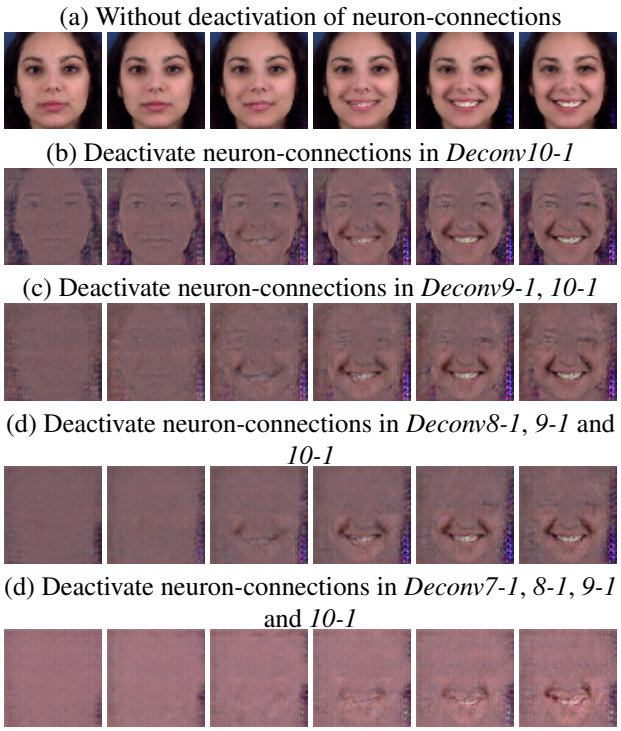

Figure 5: **Motion, appearance decomposition.** We illustrate generated examples by deactivating neuron-connections corresponding to appearance features in each layer one by one ((a) - (d)). Frames are sampled with time step 4.

As shown in Table 4, models only using adversarial loss achieve lower video FID than those only using reconstruction loss. However, results in Table 5 and Table 6 indicate that the use of reconstruction loss manifests in significantly higher SSIM and PSNR than models only using adversarial loss. We conclude that adversarial loss is instrumental in improving the perceptual quality of videos, as it enforces the Generator to create videos, matching the distribution of the training data. At the same time the reconstruction loss encourages the Generator to produce frames, resembling the ground truth by reducing the pixel-wise distance, see Figure 4.

In contrast to both single loss experiments, the ImaGINator (using both losses), w.r.t. both evaluation metrics

	Adv. Loss	Recon. Loss	Two Losses
MUG	0.54	0.74	**0.75**
NATOPS	0.87	0.88	**0.88**
Weizmann	0.50	0.54	**0.73**
UvA-NEMO	0.64	0.66	**0.66**

Table 5: **Evaluation of frame quality** between generated frames and ground truth on four datasets using SSIM. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

	Adv. Loss	Recon. Loss	Two Losses
MUG	19.24	22.60	**22.63**
NATOPS	26.72	27.10	**27.39**
Weizmann	17.01	18.03	**19.67**
UvA-NEMO	19.87	20.02	**20.04**

Table 6: **Evaluation of frame quality** between generated frames and ground truth on four datasets using PSNR. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

	MUG	NATOPS	Weizmann	UvA-NEMO	BU-4DFE
VGAN [9]	74.72	167.71	127.31	30.01	273.94
MoCoGAN [10]	45.67	49.46	116.08	29.81	62.99
ImaGINator	**29.02**	**26.86**	**99.80**	**16.16**	**32.64**

Table 7: **Evaluation of video quality** on five datasets using video FID, pertaining to VGAN, MoCoGAN and proposed ImaGINator. Lower video FID relates to better video quality.

	MUG	NATOPS	Weizmann	UvA-NEMO	BU-4DFE
VGAN [9]	0.28	0.72	0.29	0.21	0.24
MoCoGAN [10]	0.58	0.74	0.42	0.45	0.45
ImaGINator	**0.75**	**0.88**	**0.73**	**0.66**	**0.76**

Table 8: **Evaluation of image quality** on five datasets using SSIM, pertaining to VGAN, MoCoGAN and proposed ImaGINator. Higher SSIM indicates better structure similarity between generated frames and ground truth.

	MUG	NATOPS	Weizmann	UvA-NEMO	BU-4DFE
VGAN [9]	14.54	20.99	15.78	13.43	14.56
MoCoGAN [10]	18.16	21.82	17.58	16.58	17.64
ImaGINator	**22.63**	**27.39**	**19.67**	**20.04**	**22.53**

Table 9: **Evaluation of image quality** on five datasets using PSNR, pertaining to VGAN, MoCoGAN and proposed ImaGINator. Higher PSNR indicates better frame quality.

achieves the best results. Hence, both types of losses are complementary and pertinent for the performance of the ImaGINator.

B.2. Motion and Appearance decomposition

The proposed *spatio-temporal fusion* encourages the Decoder to focus on generating motion features, by reusing appearance features from the Encoder. Towards demonstrating this, we conduct an experiment, in which we visualize generated results, while controlling neuron-connections in different layers of the Decoder. We use the model, pretrained on the **MUG** dataset, and proceed to *deactivate* neuron-connections corresponding to appearance features in layers *Deconv10-1*, *9-1*, *8-1* and *7-1*, incrementally and generate thereby four groups of results (each layer corresponds to one group results). Figure 5 showcases the impact of the deactivation of neuron-connections in each layer from top to bottom, resulting in an exhibited lowered appearance information. While in the last layer, appearance is nearby vanished, a remaining motion can still be observed. This indicates the effect of our *spatio-temporal fusion*, which successfully encourages the Decoder to focus on generating motion by reusing the appearance.

B.3. Experiments on BU-4DFE dataset

The third experiment provides results on the BU-4DFE dataset related to ImaGINator, as well as two state-of-the-art methods, VGAN and MoCoGAN.

The **BU-4DFE** dataset consists of 606 facial expression videos of 101 subjects. The subjects exhibit expressions associated to the categories *anger*, *disgust*, *happiness*, *fear*, *sadness*, and *surprise*.

Table 7, Table 8 and Table 9 summarize the results of the third experiment. Notably, ImaGINator achieves the lowest video FID (32.64), as well as the highest SSIM (0.76) and PSNR (22.53), outperforming other state-of-the-art methods. The same table summarizes results for all datasets, and we note that the proposed ImaGINator consistently and systematically outperforms VGAN and MoCoGAN w.r.t. all three evaluation metrics and on all five datasets. Related generated video frames from the three methods on BU-4DFE dataset are depicted in Figure 6.

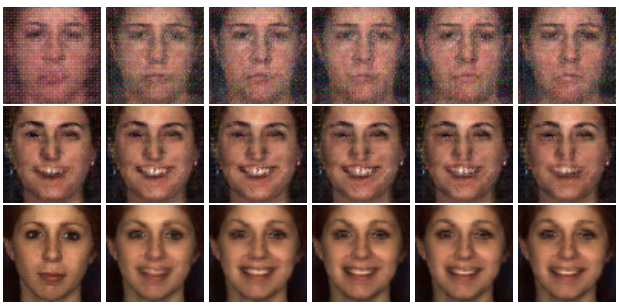

Figure 6: **Example generated video frames on BU-4DFE.** We illustrate generated video frames from VGAN (top), MoCoGAN (middle) and our ImaGINator (down). Frames are sampled with time step 3.

B.4. Experiments on BAIR robot push dataset

Finally we compare our ImaGINator with SV2P [11] on the BAIR robot push dataset in order to test the generalization of our model. We present quantitative evaluation results in Table 10, which showcase that our method significantly outperforms SV2P w.r.t. all 3 evaluation metrics: SSIM, PSNR and video FID. Moreover, we present generated examples from both methods in Figure 7, where the robot arm disappears in videos generated by SV2P, while it remains visible in videos generated by our approach (as in the original dataset). In summary of the new experiments, our approach consistently outperforms SV2P.

	SSIM	**PSNR**	**FID**
SV2P	0.78	18.35	25.12
ImaGINator	**0.89**	**21.47**	**9.89**

Table 10: Comparision of ImaGINator and SV2P.

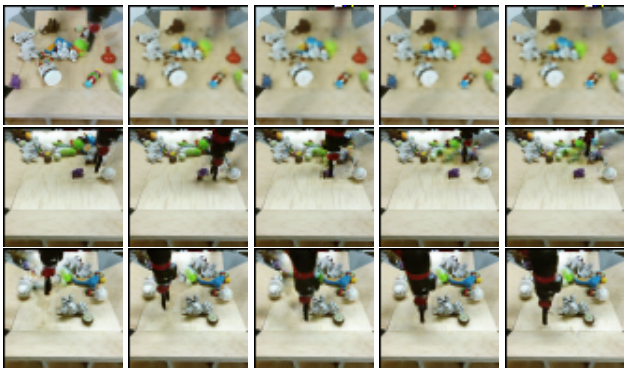

Figure 7: **Generated frames on BAIR robot push dataset.** Top: SV2P, middle and bottom: proposed ImaGINator. Time step of 3 sampling.

C. Generated examples

Due to page limitation in the main paper, we here provide additional examples, generated by ImaGINator on the six datasets MUG, NATOPS, Weizmann, UvA-NEMO, BU-4DFE and BAIR robot push. We randomly choose results from the generated data. Frames from different datasets are shown in Figure 8 (**UvA-NEMO**), Figure 9 (**MUG**), Figure 10 (**NATOPS**), Figure 11 (**Weizmann**), Figure 12 (**BU-4DFE**) and Figure 13 (**BAIR robot push**). Each line represents a video sequence generated based on the input image, shown at the first column.

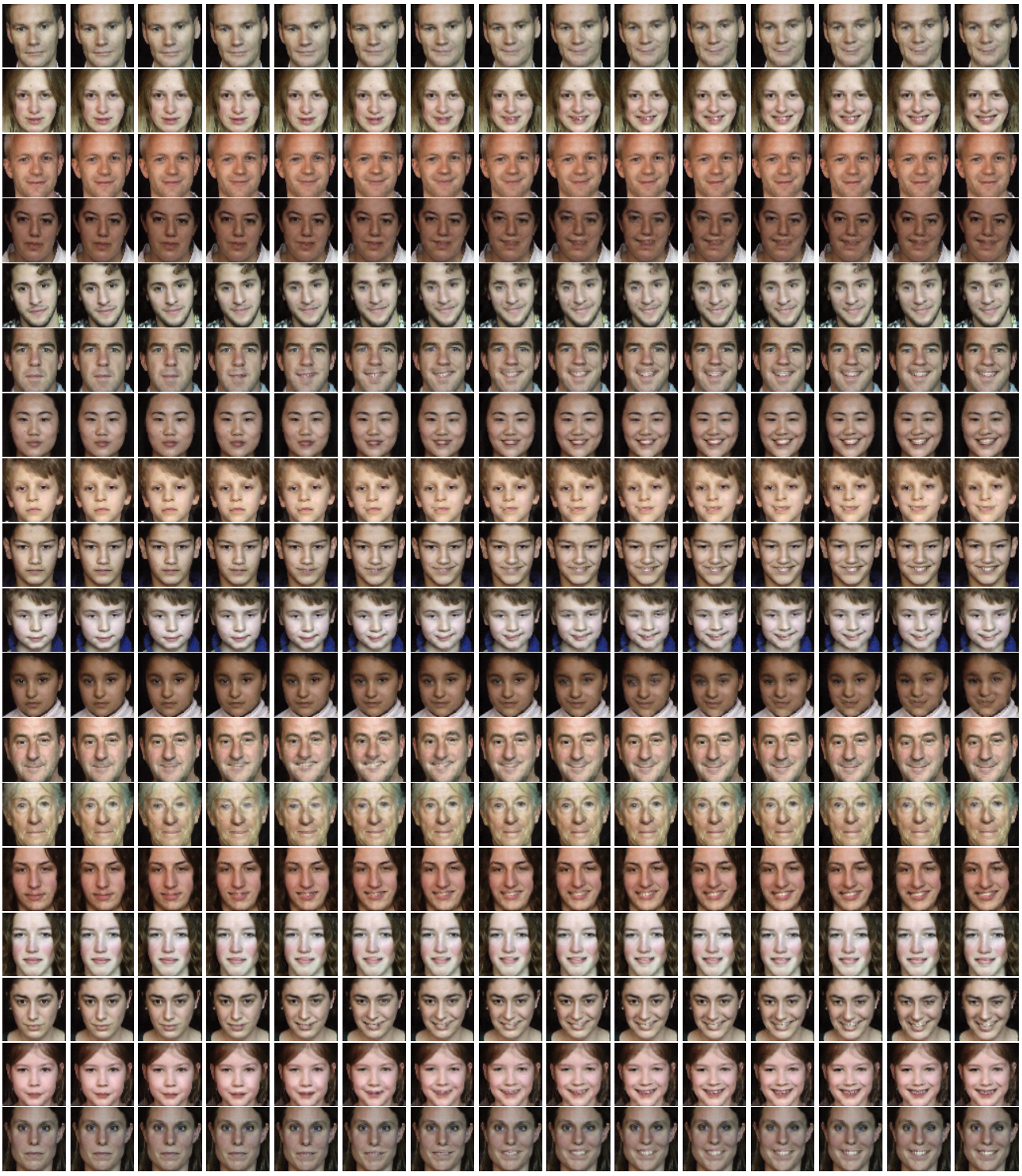

Figure 8: Generated examples from **UvA-NEMO**.

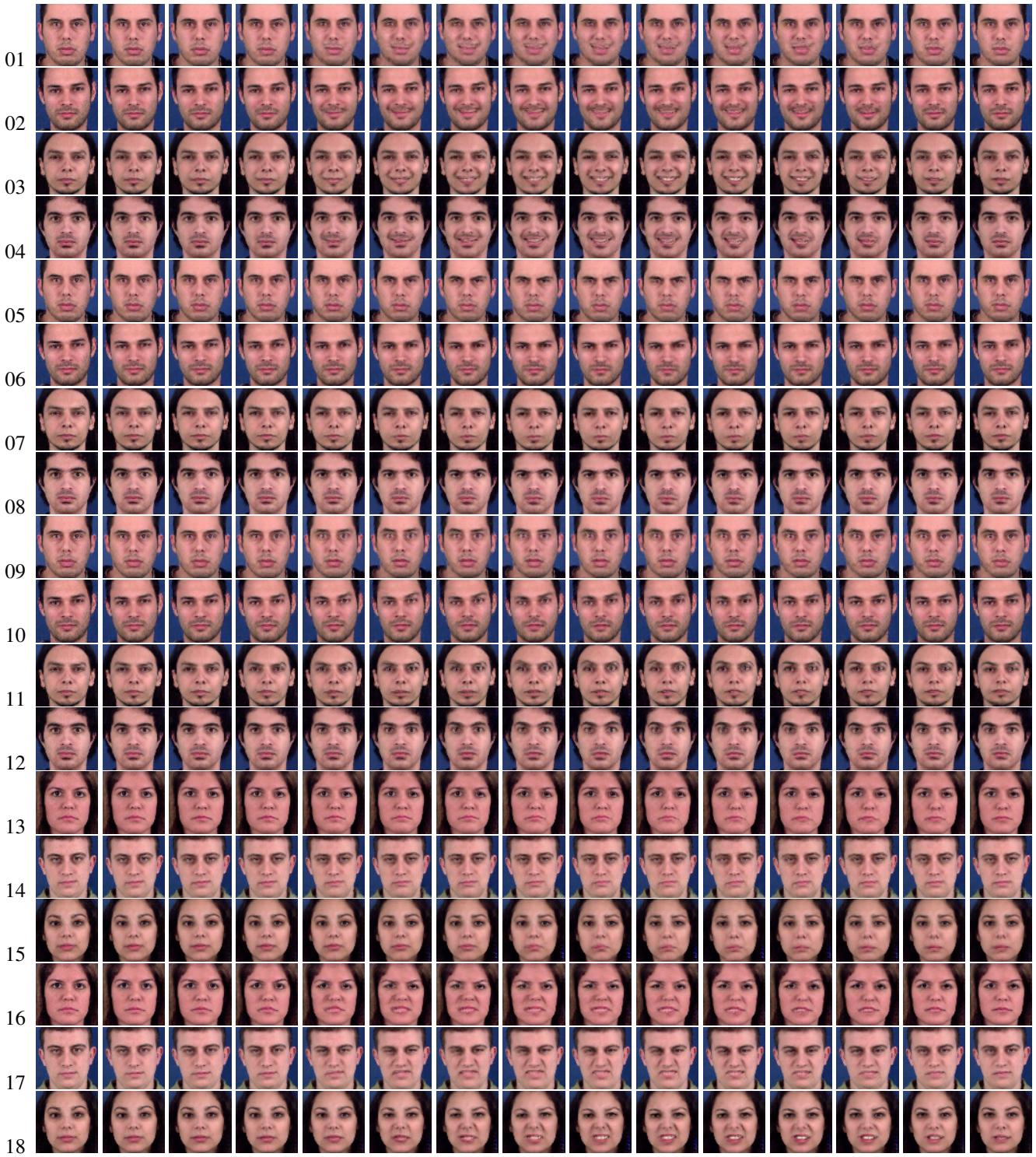

Figure 9: Generated examples from **MUG**. Labels are *happiness* (01,02,03,04), *anger* (05,06,07,08), *fear* (09,10,11,12), *sadness* (13,14,15) and *disgust* (16,17,18).

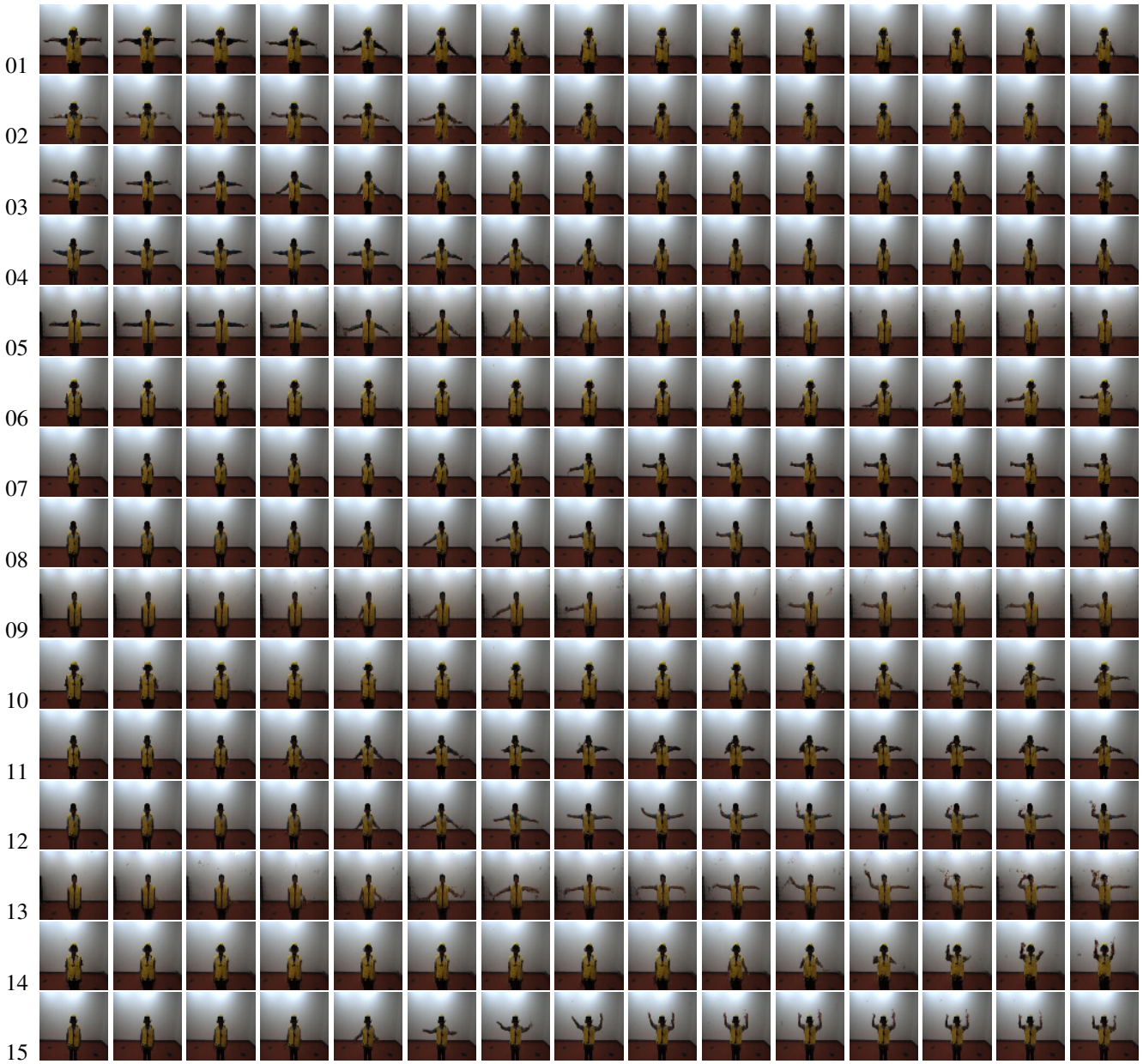

Figure 10: Generated examples from **NATOPS**. Labels are *Fold Wings* (01,02,03,04,05), *All Clear* (06,07,08,09), *Nosegear Steering* (10,11), *Turn Right* (12,13) and *Move Ahead* (14,15).

Figure 11: Generated examples from **Weizmann**. Labels are *One hand wave* (01,02,05,06), *Two hands wave* (03,04,11,12), *Bend* (07,08,13,14) and *Jack* (09,10).

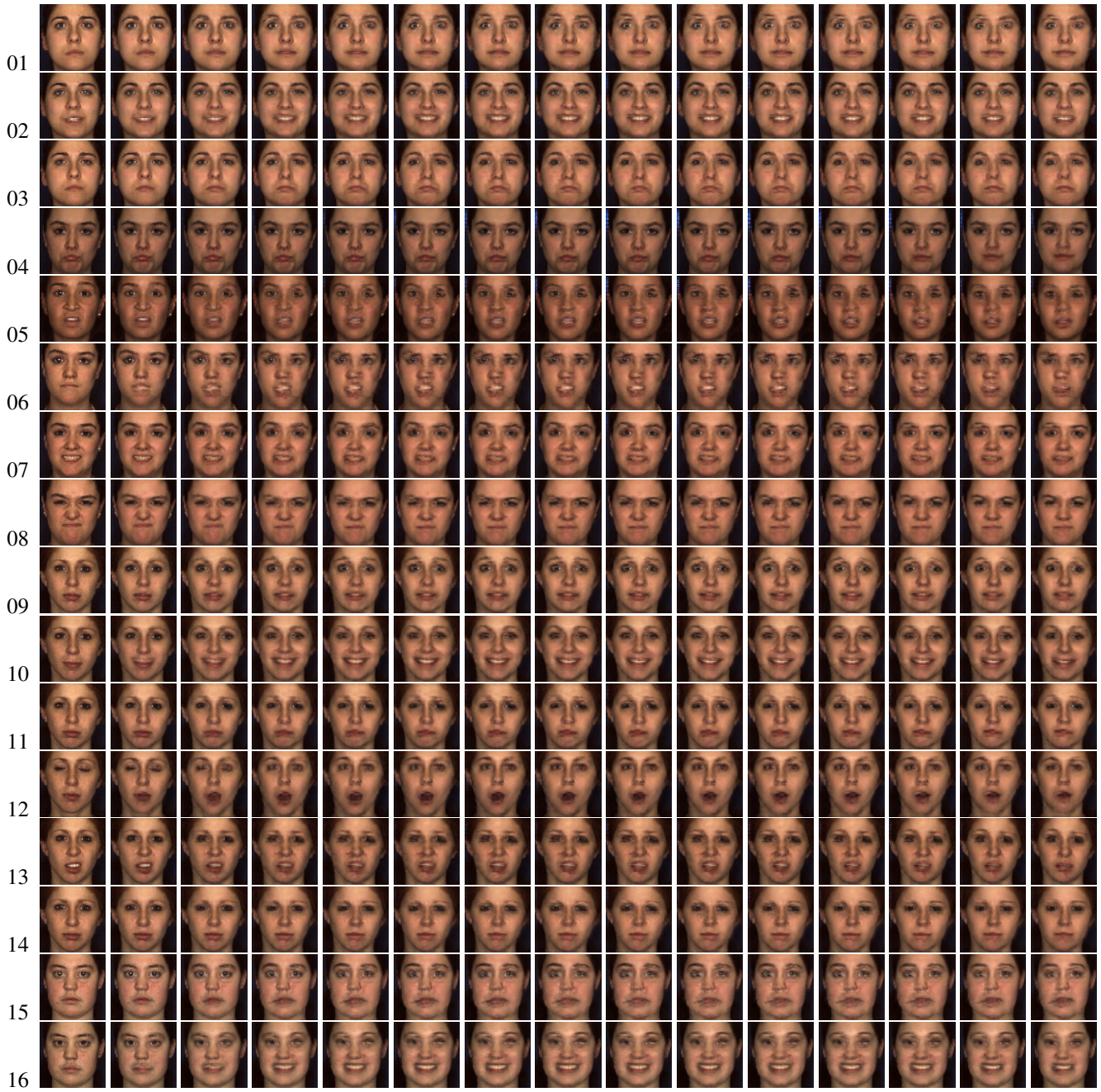

Figure 12: Generated examples from **BU-4DFE**. Labels are *happiness* (02,07,10,16), *fear* (01,09,15), *sadness* (03,04,11), *anger* (08,14), *surprise* (06,12) and *disgust* (05,13).

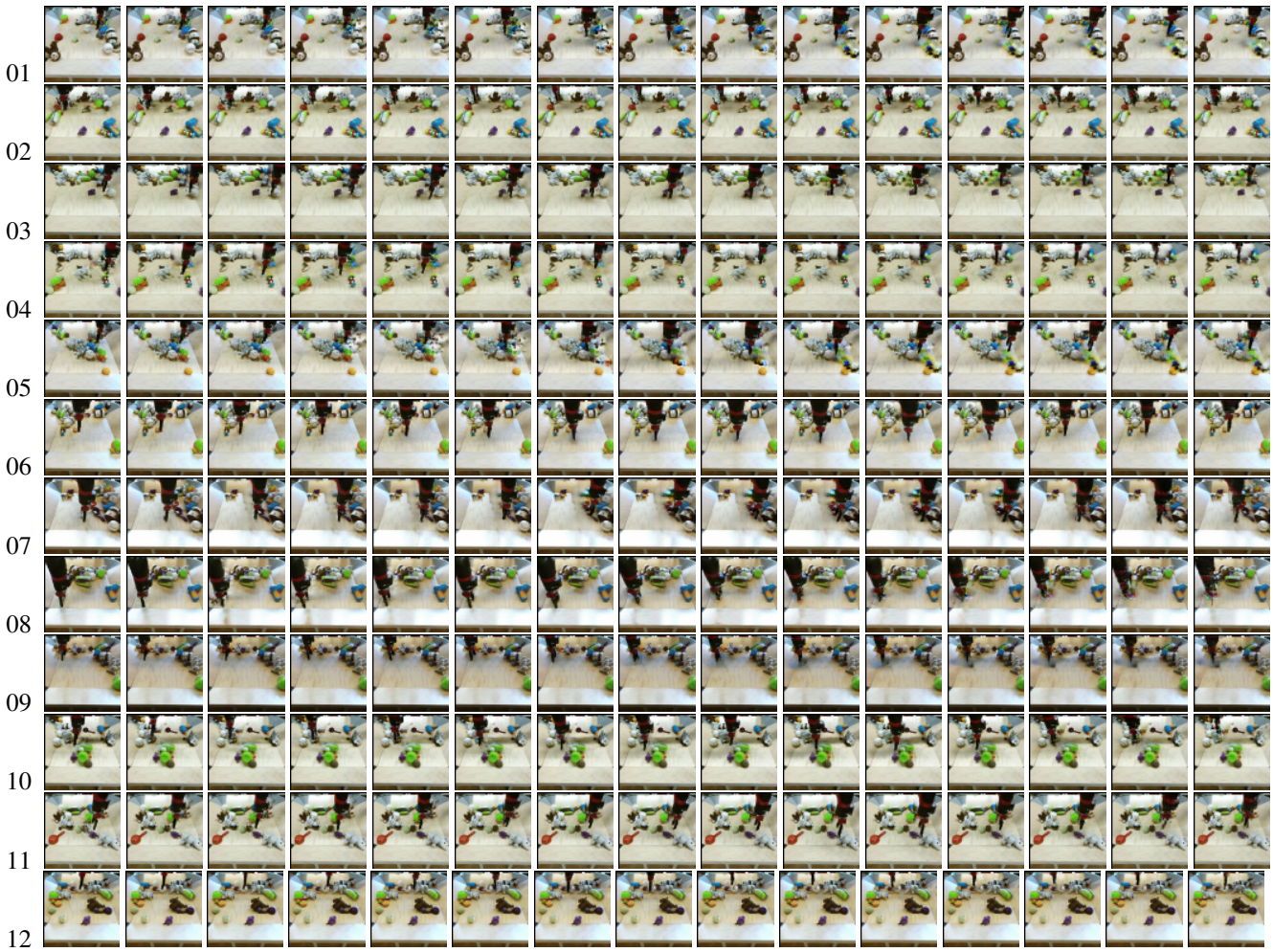

Figure 13: Generated samples from **BAIR robot push**.

References

[1] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *Image analysis for multimedia interactive services (WIAMIS), 2010 11th international workshop on*, pp. 1–4, IEEE, 2010.

[2] H. Dibeklioğlu, A. A. Salah, and T. Gevers, "Are you really smiling at me? spontaneous versus posed enjoyment smiles," in *ECCV*, 2012.

[3] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu, "A high-resolution spontaneous 3d dynamic facial expression database," in *FG*, 2013.

[4] Y. Song, D. Demirdjian, and R. Davis, "Tracking Body and Hands For Gesture Recognition: NATOPS Aircraft Handling Signals Database," in *FG*, 2011.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *TPAMI*, vol. 29, pp. 2247–2253, December 2007.

[6] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," *CoRL*, 2017.

[7] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *ICML*, 2015.

[8] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *ICLR*, 2018.

[9] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *NIPS*, 2016.

[10] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *CVPR*, 2018.

[11] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *ICLR*, 2018.