DeepLandscape: Adversarial Modeling of Landscape Videos
Supplementary material

Elizaveta Logacheva[1], Roman Suvorov[1], Oleg Khomenko[1], Anton Mashikhin[1], and Victor Lempitsky[1,2]

[1] Samsung AI Center, Moscow
[2] Skolkovo Institute of Science and Technology, Moscow

1 Training the main model

Training Configuration Our final models follows original StyleGAN training schedule. We alternate between two phases: a resolution transition phase for 600k samples then a stabilization phase for 600k samples. After final reslution is reached we train model until the number of batches reaches 450k. The proportion of pairwise discriminator changes linearly from 0.5 to 0.1 during the resolution transition phase. We use crops instead of generated frames when update pairwise discriminator with probabily 0.5. For inference we used accumulated exponential moving average with $\alpha = 0.999$ to generate samples. Our final model was trained using Adam optimizer with parameters $\beta_1 = 0, \beta_2 = 0.99$.

As in the original StyleGAN, we change batch size parameter depending on resolution: (4px, 512), (8px, 256), (16px, 128), (32px, 64), (64px, 32), (128px, 32), (256px, 16), (512px, 8), (1024px, 8). Learning rates are: (up to 128px, 1e-3), (128px, 1.5e-3), (256px, 2e-3), (eq. or bigger than 256px, 3e-3)

Pairwise Discriminator The pairwise discriminator differs from the original StyleGAN discriminator only in the input Conv 1x1 layer which has half the number of output channels of the original StyleGAN discriminator and is applied to each frame independently. After that both feature maps are concatenated.

Balancing discriminators To choose the most effective way of balancing two discriminators we evaluated four different experiments (image resolution is 128px). While freq = 0.3 and freq = 0.5 suffer from much worse image quality, *decay* and freq = 0.1 behaves similarly but *decay* works slightly better on moving objects and generates more compelling dynamics.

2 Inference Details

Our overall inference procedure consists of the following steps.

1. Training encoder **E** on a dataset of samples from a pretrained **G**.

Setup	FID↓	SSIM↑	LPIPS↓	R↑
decay	56.13	0.884	0.062	**0.00**
freq = 0.1	**54.15**	0.880	**0.064**	-0.03
freq = 0.3	63.48	0.887	0.058	-0.12
freq = 0.5	82.10	**0.893**	0.055	-0.15

Table 1. Different techniques to balance discriminators. The column R in the table is obtained from the side-by-side user study. It shows the change in frequency when assessors prefer this variant to the default one (*decay*). Although decaying the relative frequency doesn't give the best results when comparing against any quantitative metric, it balances image quality with motion plausibility and wins user preference.

2. Given a real image x to be animated, obtain a set of style vectors $\mathcal{W}'$ using $\mathbf{E}$.
3. Starting from $\mathcal{W}'$, find $\widehat{\mathcal{W}}$ and $\widehat{\mathcal{S}}$ with gradient descent, to improve reconstruction and preserve ability to animate.
4. Having $x, \widehat{\mathcal{W}}, \widehat{\mathcal{S}}$ fixed, optimize $\mathbf{G}$ to improve reconstruction even more.

Steps 1, 2, 4 are pretty simple, so their description in the paper is fairly detailed. Thus, here we present only the extended definition of the step 3 (latents optimization). We present two variants: one without using a segmentation mask (Algorithm 1, a part of EOI and EOIF); and another one relying on a segmentation mask to route information between $\mathcal{S}^{\mathrm{st}}$ and $\mathcal{S}^{\mathrm{dyn}}$ (Algorithm 2, a part of EOIFS). Other variants of inference can be obtained by changing EOI, specifically:

- turning off the $\mathcal{W}$ penalty L^{O}_{init} (EO, MO, I2S);
- changing $\mathcal{W}$ initialization: mean style instead of $\mathbf{E}$ predictions (MO, I2S);
- changing $\mathcal{S}$ initialization: random instead of zero (I2S, E);
- turning off optimization of $\mathcal{S}$ (I2S);
- not optimizing latents at all (E).

In I2S, we also tried using $\mathbf{E}$-based initialization for $\mathcal{W}$, with no success. Initialization of $\mathcal{S}$ is not very important in EOI, EOIF, EOIFS, but starting from zeros slightly helps stability.

In order to regularize $\mathcal{S}$ and to prevent too much details to be described by spatial inputs, we tried both L2-regularization and gradient scaling. While L2 helps, we found gradient scaling much more efficient: it leads to better convergence (more accurate reconstruction) and still allows to push information from $\mathcal{S}$ to $\mathcal{W}$. We found experimentally that during latents optimization $\frac{\partial L^O}{\partial \mathcal{S}}$ should be divided by 1000 for best results. This effectively changes relative learning rate for $\mathcal{S}$, comparing to the learning rate of $\mathcal{W}$.

2.1 What $\mathbf{z}^{\mathrm{dyn}}$ Actually Describe?

In order to manipulate lighting on a real image, we train a dedicated neural network $\mathbf{A}$, which approximates local dynamics of a multilayer perceptron $\mathbf{M}$,

Algorithm 1 EOI: initialize with **E**ncoder, **O**ptimize, tie $\mathcal{W}$ to **I**nitial values

Inputs: generator **G**, style initialization $\mathcal{W}'$, input image x.
Outputs: optimized $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
Hyperparameters: number of iterations N, perceptual loss coefficient $\lambda_{PL} = 0.01$, gradient scale for $\mathcal{S}$ $\lambda_{\mathcal{S}} = 0.001$, Adam learning rate $lr = 0.1$.

```
$\widehat{\mathcal{W}} \leftarrow \mathcal{W}'$
$\widehat{\mathcal{S}} \leftarrow 0$
$UpdateRule \leftarrow$ initialize Adam optimizer for $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
$iter \leftarrow 0$
while $iter < N$ do
    $y \leftarrow \mathbf{G}(\widehat{\mathcal{W}}, \widehat{\mathcal{S}})$    ▷ Obtain reconstructed image
    $L^O_{rec} \leftarrow MAE(y, x) + \lambda_{PL} PL(y, x)$    ▷ Reconstruction loss
    $L^O_{init} \leftarrow MSE(\widehat{\mathcal{W}}, \mathcal{W}')$    ▷ Style regularization
    $L^O \leftarrow L^O_{rec} + L^O_{init}$    ▷ Total latents loss
    Calculate $\frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}}}$    ▷ loss.backward()
    $\frac{\partial L^O}{\partial \widehat{\mathcal{S}}} \leftarrow \lambda_{\mathcal{S}} \frac{\partial L^O}{\partial \widehat{\mathcal{S}}}$    ▷ Scale gradients for $\widehat{\mathcal{S}}$
    $\widehat{\mathcal{W}}, \widehat{\mathcal{S}} \leftarrow UpdateRule(\widehat{\mathcal{W}}, \widehat{\mathcal{S}}, \frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}}})$
    If $L^O$ does not improve over 20 iterations, halve $lr$
    If $L^O$ does not improve over 100 iterations, stop early
    $iter \leftarrow iter + 1$
end while
return $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
```

which maps $\mathbf{z}$ to $\mathbf{w}$. During training of **G** and **M**, $\mathbf{z}^{\text{dyn}} \in \mathbb{R}^3$ is sampled from standard normal distribution. However, it is not practical to sample styles for real images, because we usually want to get something concrete (e.g. day to evening or evening to night conversion).

Thus, we needed a technique to build an "interpretation" of 3 numbers which make up $\mathbf{z}^{\text{dyn}}$. A well established approach for that is to (a) sample a set of synthetic images from **G**, (b) manually assign them class labels (e.g. day, evening, night); (c) obtain "direction vectors", which correspond to the shortest path from one class to another in the latent space. Having direction vectors, one can modify $\mathbf{z}^{\text{dyn}}$ along them in order to change image style accordingly. This approach can help to build an interpretation of a complex high-dimensional model.

However, in our case we have only 3 components to interpret, thus we decided to take a more simple way: manually change $\mathbf{z}^{\text{dyn}}$ coordinates one-by-one and try to describe the way the image changes. For each coordinate we tried values from $\{-3, -2, -1, 0, 1, 2, 3\}$ while keeping other coordinates zero. We also tried changing pairs and triplets of coordinates the same way.

We found that as a result of multiple **G** training sessions on the same dataset, $\mathbf{z}^{\text{dyn}}$ consistently received approximately the following semantics:

1. The first coordinate changes brightness without altering color temperature (day-to-night). Thus, when moving from day to night we do not arrive to a warm yellow sunset.

Algorithm 2 EOIFS: initialize with **E**ncoder, **O**ptimize, tie $\mathcal{W}$ to **I**nitial values, guide $\mathcal{S}$ with **S**egmentation

Inputs: generator $\mathbf{G}$, style initialization $\mathcal{W}'$, input image x, static regions mask m (1 for static regions, 0 for sky and water).
Outputs: optimized $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
Hyperparameters: number of iterations N, perceptual loss coefficient $\lambda_{PL} = 0.01$, gradient scale for $\mathcal{S}$ $\lambda_{\mathcal{S}} = 0.001$, Adam learning rate $lr = 0.1$.

```
$\widehat{\mathcal{W}} \leftarrow \mathcal{W}'$
$\widehat{\mathcal{S}^{\mathrm{st}}}, \widehat{\mathcal{S}^{\mathrm{dyn}}} \leftarrow 0$    ▷ Initialize $\widehat{\mathcal{S}}$ with zeros
$UpdateRule \leftarrow$ initialize Adam optimizer for $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
$iter \leftarrow 0$
while $iter < N$ do
    $y \leftarrow \mathbf{G}(\widehat{\mathcal{W}}, \widehat{\mathcal{S}})$    ▷ Obtain reconstructed image
    $L^O_{init} \leftarrow MSE(\widehat{\mathcal{W}}, \mathcal{W}')$    ▷ Style regularization
    if iter % 2 == 0 then    ▷ Even iterations are for static regions
        $y_m \leftarrow y \circ m$    ▷ Zero out dynamic regions
        $x_m \leftarrow x \circ m$
        $L^O_{rec} \leftarrow MAE(y_m, x_m) + \lambda_{PL} PL(y_m, x_m)$    ▷ Reconstruction loss
        $L^O \leftarrow L^O_{rec} + L^O_{init}$    ▷ Total latents loss
        Calculate $\frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{st}}}}$    ▷ Calculate grad only w.r.t $\widehat{\mathcal{S}^{\mathrm{st}}}$
        $\frac{\partial L^O}{\partial \widehat{\mathcal{S}^{\mathrm{st}}}} \leftarrow \lambda_{\mathcal{S}} \frac{\partial L^O}{\partial \widehat{\mathcal{S}^{\mathrm{st}}}}$    ▷ Scale gradients for $\widehat{\mathcal{S}^{\mathrm{st}}}$
        $\widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{st}}} \leftarrow UpdateRule(\widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{st}}}, \frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{st}}}})$
    else
        $y_m \leftarrow y \circ (1 - m)$    ▷ Zero out static regions
        $x_m \leftarrow x \circ (1 - m)$
        $L^O_{rec} \leftarrow MAE(y_m, x_m) + \lambda_{PL} PL(y_m, x_m)$    ▷ Reconstruction loss
        $L^O \leftarrow L^O_{rec} + L^O_{init}$    ▷ Total latents loss
        Calculate $\frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{dyn}}}}$    ▷ Calculate grad only w.r.t $\widehat{\mathcal{S}^{\mathrm{dyn}}}$
        $\frac{\partial L^O}{\partial \widehat{\mathcal{S}^{\mathrm{dyn}}}} \leftarrow \lambda_{\mathcal{S}} \frac{\partial L^O}{\partial \widehat{\mathcal{S}^{\mathrm{dyn}}}}$    ▷ Scale gradients for $\widehat{\mathcal{S}^{\mathrm{dyn}}}$
        $\widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{dyn}}} \leftarrow UpdateRule(\widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{dyn}}}, \frac{\partial L^O}{\partial \widehat{\mathcal{W}}, \widehat{\mathcal{S}^{\mathrm{dyn}}}})$
    end if
    If $L^O$ does not improve over 20 iterations, halve $lr$
    $iter \leftarrow iter + 1$
end while
return $\widehat{\mathcal{W}}, \widehat{\mathcal{S}}$
```

2. The second coordinate changes brightness and color temperature together: negative values lead to darker images with all lights (city, sunset) getting more saturated and warm; positive values lead to lighter images with colder colors. One can get day-to-evening conversion with that coordinate alone.
3. The third coordinate does almost the same as the first one does. We found no significant difference between them.
4. By changing the first and the second coordinates, one can obtain dark night, warm sunset, blue hour, bright day with clouds, bright day with clear sky.

Our experiments show that $\mathbf{z}^{\text{dyn}}$ affects image style in a fairly monotonic way.

Using the described methodology, we constructed a vocabulary of 9 styles, which correspond to different time of day and weather. We use only these styles for all videos where we animate real images for our quantitative and qualitative experiments. We use styles randomly sample from normal distribution for fully synthetic videos.

3 Inference procedure ablation study.

Where we quantify the impact of different elements of our inference algorithm on the reconstruction accuracy, image quality, static consistency and motion amount. Image quality and static consistency for best inference variants (EOIF and EOIFS) are discussed in the paper, Section 4 (Experiments). The reconstruction quality is evaluated via LPIPS and SSIM measured between the input and the reconstructed images. The amount of motion is quantified as mean optical flow [1] in the sky region, according to semantic segmentation. We generate videos in the same way as for other experiments. The results of this ablation study (Table 2) verify that all steps of our inference procedure are needed to obtain animations that both have plausible motion and fit the input images well.

Algorithm	SSIM↑	LPIPS↓	Flow$\times 10^{-3}$ ↑
I2S [2]	0.80	0.18	1.3
MO	0.95	0.07	1.3
E	0.54	0.43	2.2
EO	0.95	0.07	1.5
EOI	0.92	0.11	2.5
EOIF	**0.96**	**0.04**	2.3
EOIFS	0.94	0.05	**2.6**

Table 2. Quantitative evaluation of inference procedure: reconstruction quality and motion amount. While sevral variants result in good reconstruction, only EOIF and EOIFS variants yield both good reconstruction and motion.

4 Animated Landscape finetuning

In order to ensure fair comparison, we tried to reproduce results from AL paper using only our video dataset. We tried both training from scratch and finetuning the publicly available model for 50 to 200 epochs. Training from scratch did not converge, so we present here only metrics obtained with finetuning. Table 3 shows that finetuning damages the model. This is most probably due to the very small dataset, which contains motions of very different speed. Authors of AL somehow equalized speed of different videos, but the exact methodology for that is unknown. Training on unequalized videos is harmful. On contrary, our model does not require equalization of motion speed, which allows to use more dirty data without degradation of performance.

	FVD↓	LPIPS↓	SSIM↑	FID↓	User preference↑
$\mathrm{AL}_{\mathrm{noint}}$	162	**0.063**	0.92	**51.9**	**0.68**
$\mathrm{AL}_{\mathrm{finetuned}}$	**159**	0.065	0.92	53.4	0.32

Table 3. Comparison of pretrained and finetuned AL

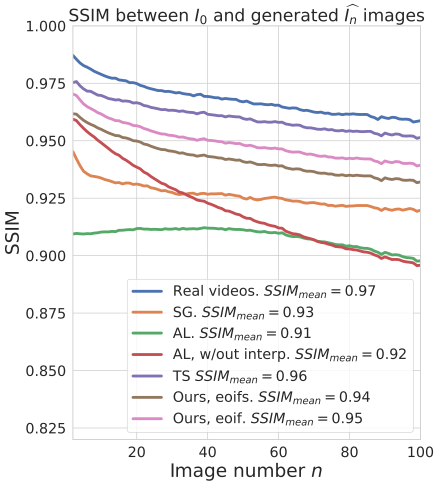

Fig. 1. Continuation of Figure 7. Quantitative comparison of image quality, static consistency and motion plausibility

5 The Structural Similarity Index (SSIM)

Figure 1 presents masked SSIM between I_0 and $\widehat{I}_n$, which mostly measure image quality and static consistency. Note that TS baseline, which uses segmentation and simply copies static parts, outperforms other methods (but losses the game when it comes to perceptual quality and motion plausibility).

6 Side-by-side Comparison on the speed of real videos

Method	**EOIF**	**EOIFS**
SG	0.28	0.27
AL (no int)	0.33	0.36
AL (+ style)	0.14	0.12
TS	0.11	0.12
Real	0.68	0.70
Ours (EOIF)	–	0.52
Ours (EOIFS)	0.48	–

Fig. 2. Ratio of wins row-over-column for side-by-side setting B, synthetic video speed aligned to that of real ones (faster videos).

On Figure 2 we present side-by-side user study, setup B, i.e. with speed of synthetic videos aligned to that of real ones. Note that real videos win more often, but advantage of our method against competitors is even more evident.

References

1. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 9000–9008
2. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 4432–4441