Dynamic Neural Portraits (Supplementary Material)

Michail Christos Doukas[1,2] Stylianos Ploumpis[2] Stefanos Zafeiriou[1,2]
[1]Imperial College London, UK [2]Huawei Technologies, London, UK
{michail.christos.doukas, stylianos.ploumpis, stefanos.zafeiriou1}@huawei.com

1. Database

Here, we provide more details of the video portrait data we used to train and test our proposed model. We thank the authors of [6, 12, 9, 14] for sharing their video data. As can be seen in Table 1 and 2, we have used eleven different video portraits throughout use qualitative and quantitative experiments, at various resolutions: 256×256, 512×512, 1024×1024. We denote the number of frames we used to train each person-specific model by N_{train}, while N_{test} represents the number of test frames. Training frames come for the beginning of videos and test frames from the end, as we maintain the original training and test split of data. Please note that some portraits were used only for the task of cross-identity reenactment, either as source ($N_{train} = 0$) or as target identities ($N_{test} = 0$). The eleven identities are displayed in Fig. 1.

2. Face and Gaze Tracking

Our proposed method is primarily driven with head pose, facial expression and eye gaze information. We employ [11] to extract gaze angles from frames. We recover expression blendshapes using two different approaches, one for reconstruction (self-reenactment) and another one for cross-identity motion transfer (reenactment) experiments. We observed that GANFIT [7] operates very well when it comes to expression and shape (identity) disentanglement, thus we use this tracker throughout cross-identity reenactment. Nonetheless, during self-reenactment, we use the approach presented in [4], which relies on dense 3D facial landmarks regressed by RetinaFace [2] and 3DMM fitting on LSFM [1], as it appears to retain more detailed expression information from RGB frames in comparison to GANFIT. Moreover, we employ the same tracking method with [9] to retrieve head pose parameters.

3. Training

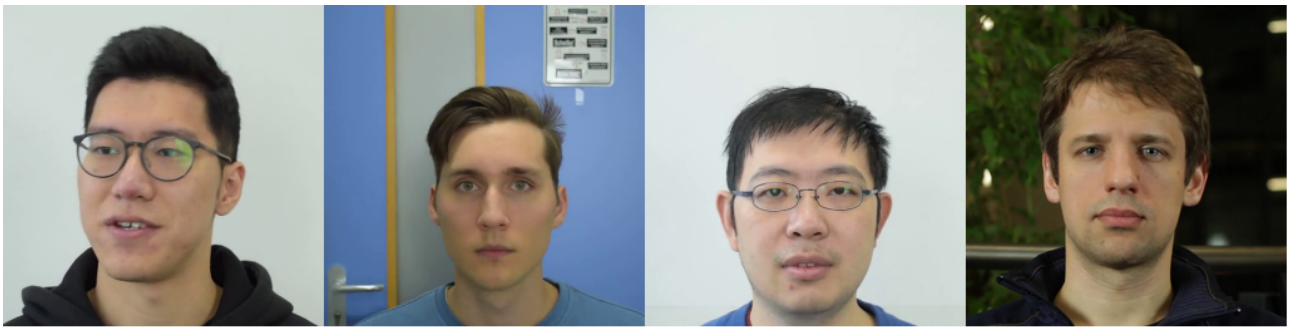
For the optimisation of our neural networks we utilise ADAM [13], with $\beta_1 = 0.9, \beta_2 = 0.999$ and learning rate $\eta = 0.0005$. We train a new model (set of networks) for each video portrait, which requires 100K to 150K iterations (approximately 18-27 hours) on a single NVIDIA Tesla V100 PCIe 32 GB, depending on the number of training frames. We use a batch size of 16. For audio-driven portrait synthesis, we train networks N_{aud} and N_{att} alongside the MLP and CNN-based decoder on the task of reconstruction.

4. Networks Architecture

Our model consists of 1) an MLP network F, 2) a CNN-based image decoder D, and optionally 3) two networks N_{aud} and N_{att} for audio feature prediction. For the architecture of audio-related networks, please refer to [9].

MLP network F. Our proposed MLP network consists of eight consecutive linear layers, each followed by a ReLU activation function. Similarly to NeRF [15], we use a skip connection in the fourth layer. The MLP's input vector emerges from the concatenation of the position $\gamma(\mathbf{x})$, head pose $\gamma(\mathbf{p}_i)$ and gaze $\gamma(\mathbf{g}_i)$ parameters, all of which are position encoded, as well as the expression parameters $\mathbf{e}_i$ and learnable latent code $\mathbf{v}_i$. In the output, a simple linear layer predicts colour $\mathbf{c}$ and feature vector $\mathbf{f}$. Please see Fig. 2 for more details.

CNN-based decoder D. Our adopted image decoder is based on the decoding part of the generator proposed originally in pix2pixHD [18]. It receives a feature map of resolution $H_f \times W_f$, assembled from the feature vectors created by the MLP, and uses it to synthesise a photo-realistic image. It is made up of six consecutive residual layers followed by two up-sampling layers and a final output layer that hallucinates the output image of resolution $H \times W$, where $H = 4H_f$ and $W = 4W_f$. The architecture of our decoding network is displayed in Fig. 3. Please note that for the ablation study with the CNN decoder only, we use a slightly different architecture. That is, we replace the residual layers with up-sampling layers, which receive a 4×4 tile of the pose, expression and gaze parameters (replicated along the spatial dimensions) and transform it to an image.



(a) ID. 1 (b) ID. 2 (c) ID. 3 (d) ID. 4

(e) ID. 5 (f) Obama (g) May (h) Biden

(i) Macron (j) Mitsotakis (k) Trudeau

Figure 1. Illustration of the different identities (video portraits) we used throughout our experiments.

Portrait	Resolution	N_{train}	N_{test}	Experiments	Data Source
ID. 1	512×512	5513	1000	reconstruction	NerFACE [6]
ID. 2	512×512	5518	1000	reconstruction	NerFACE [6]
ID. 3	512×512	5441	1000	reconstruction	NerFACE [6]
ID. 4	256×256	0	2380	reenactment	DVP [12]
ID. 5	256×256	18138	0	reenactment	DVP [12]
Obama	512×512	7272	728	audio-based reconstruction ablation study	AD-NeRF [9]
May	512×512	5500	550	audio-based reconstruction	Head2Head [14]
Biden	512×512	5301	454	ablation study	Head2Head [14]

Table 1. Details of the video portraits we used throughout the experiments in the main script.

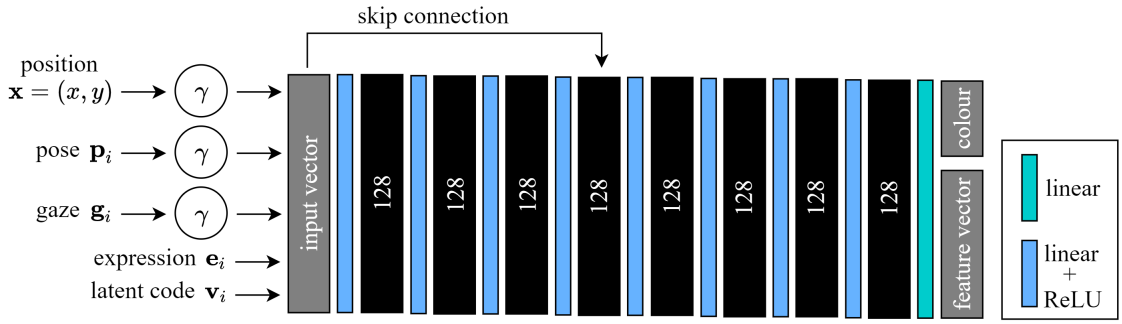


Figure 2. The architecture of our MLP network is based on the one originally proposed for NeRF [15], with the addition of a linear output layer that returns the RGB colour $\mathbf{c}$ and feature vector $\mathbf{f}$. Here, γ denotes the function of positional encoding.

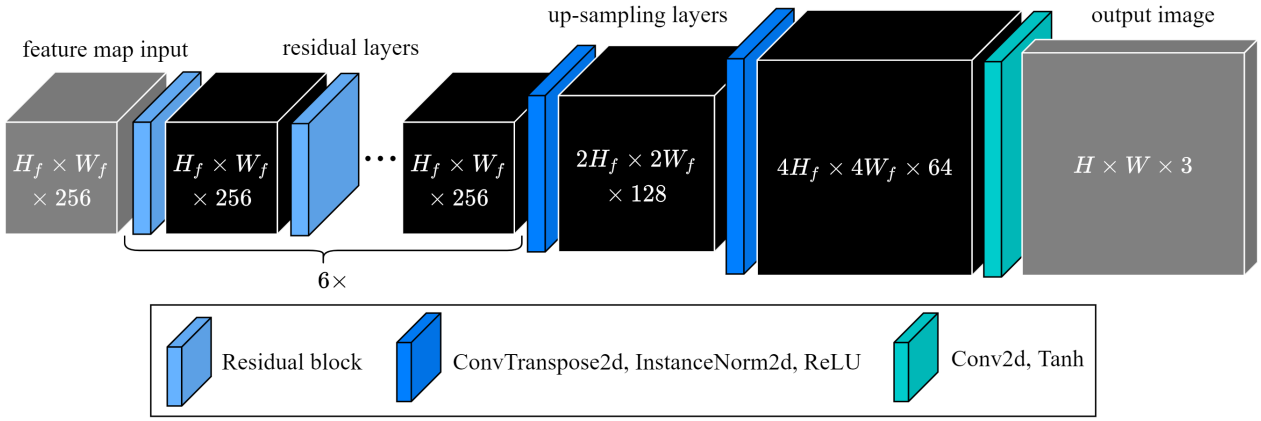


Figure 3. The architecture of our decoder is based on the generator of pix2pixHD [18], with six residual layers followed by two up-sampling layers and a final output layer. For more details on the structure of the residual block, please refer to [18].

Portrait	Resolution	N_{train}	Role	Data Source
Trudeau	512×512	0	driving	Head2Head [14]
Biden	512×512	5755	target	Head2Head [14]
Macron	1024×1024	0	driving	Head2Head [14]
Mitsotakis	1024×1024	10934	target	Head2Head [14]

Table 2. Details of the video portraits we used for the extra reenactment experiments that appear in the supplementary video.

5. Evaluation Metrics

Here, we provide more information on the evaluation metrics that we used in order to validate of our method's performance and compare it with related state-of-the-art methods.

L1-distance. This is the most basic means of evaluating the reconstructive performance of methods. Given a ground frame $\mathbf{I}_i \in [0,1]^{H \times W \times 3}$ and the corresponding generated frame $\tilde{\mathbf{I}}_i$, L1-distance is given by $||\mathbf{I}_i - \tilde{\mathbf{I}}_i||_1$.

LPIPS. By definition, L1 is a simple shallow function that assesses reconstruction only on pixel level. It has been shown that using feature maps for the evaluation of image reconstruction is more consistent with human perception. These feature maps are extracted from images with the assistance of pre-trained deep neural networks. Here, we employ the widely-used LPIPS [19] metric as a perceptual similarity score between the ground $\mathbf{I}_i$ and synthetic frames $\tilde{\mathbf{I}}_i$, which is calculated based on visual feature maps.

FID. We use FID score [10, 16] in order to measure the similarity between the dataset of ground truth frames and the dataset of frames generated by each method. This score provides an insight into the visual quality and photo-realism of synthesised frames, as this is an established means of measuring the quality of images created by generative models, such as GANs [8].

FVD. Since we actually generate video data, it is essential to evaluate the performance of different methods based on a metric that takes into account the temporal coherence among frames. For that, we assess the plausibility of generated videos with FVD [17], as this metric has shown to correlate well with human judgment on visual quality and photo-realism of video data.

CSIM. This is a metric for calculating identity preservation during reenactment. Given that cross-identity motion transfer entails the problem of transferring characteristics of the driving identity to the generated frames of the target person, it is of upmost importance to calculate how well different methods handle identity preservation. For that, we use an identity recognition network, namely ArcFace [3], for the computation of embedding vectors from the generated and real images of the target actor. Then, we compute the cosine similarity between embedding vectors coming from real and synthetic frames.

Expression Distance. One of the most important aspects of reenactment lies in successful expression transfer. In order to evaluate the generated expressions numerically, we employ DECA [5] for the extraction of expression parameters from driving and generated frames. Then, we compute the L1-distance between each pair of expression vectors, coming from the driving and synthetic data.

Pose Distance. Similarly to expression, we validate the accuracy of systems on head pose transfer. Again, we pass the driving and corresponding generated frames through DECA [5]. This yields a sequence of driving head rotation matrices $\mathbf{R}_t$ and synthetic head rotation matrices $\tilde{\mathbf{R}}_t$, $t = 1, \ldots, T$. Then, we compute head rotation distance as

$$\theta_{pose} = \arccos\Big(\frac{\text{tr}(\tilde{\mathbf{R}}_t \mathbf{R}_t^\top) - 1}{2}\Big) \tag{1}$$

and convert angle θ_{pose} to degrees.

Gaze Distance. As suggested by the uncanny valley effect, accurate gaze synthesis is very important when creating realistic human faces. We measure gaze transfer on the task of reenactment by first extracting driving gaze vectors $\mathbf{g}_t$ and synthetic gaze vectors $\tilde{\mathbf{g}}_t$, $t = 1, \ldots, T$ with the assistance of a state-of-the-art gaze detection system [11]. After that, we measure the distance between gaze vectors as the angle between them:

$$\theta_{gaze} = \arccos\Big(\frac{\mathbf{g}_t^\top \tilde{\mathbf{g}}_t}{||\mathbf{g}_t||_2 ||\tilde{\mathbf{g}}_t||_2}\Big). \tag{2}$$

Finally, we convert θ_{gaze} to degrees.

References

[1] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2):233–254, 2018.

[2] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[4] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021.

[5] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021.

[6] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021.

[7] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[9] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5784–5794, 2021.

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[11] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019.

[12] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 16–23. IEEE, 2020.

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[16] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1.

[17] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.