Frame Interpolation Transformer and Uncertainty Guidance
Supplementary Material

Markus Plack[1*] Karlis Martins Briedis[2,3] Abdelaziz Djelouah[3]
Matthias B. Hullin[1] Markus Gross[2,3] Christopher Schroers[3]
[1]University of Bonn, Bonn, Germany
[2]Department of Computer Science, ETH Zürich, Switzerland
[3]DisneyResearch|Studios, Zürich, Switzerland
mplack@cs.uni-bonn.de, karlis.briedis@inf.ethz.ch

1. Overview

We give more details on our user study together with additional results in Sec. 2. Sec. 3 contains more evaluation results of our uncertainty guidance approach. Finally, we show a full table of our ablation study in Sec. 4, interpolation of arbitrary times in Sec. 5, and give more details on our network architecture and implementation in Sec. 6.

2. User Study

We conducted an extensive user study to evaluate our method on both live-action and rendered content.

Methodology. Similar to [9] we asked users to compare the interpolation results of our approach against other methods side by side through a web interface as shown in Fig. 2. The left-right order is sampled randomly to avoid bias and we extended their methodology by adding an option for strong preference. We asked users to contribute 40 comparisons to the study, but we gave them the opportunity to rate up to 120 samples and stop at any point. The samples shown to the users were taken randomly, but we ensured that all votes were distributed equally among all samples.

Input. We used 30 frame pairs from each of the animated movies [1, 2, 4, 5] for the comparisons yielding a total of 120 pairs. For live-action, we randomly selected one pair of each scene from the validation set of DAVIS [11] (20 pairs in total) and one pair of each video from the SNU-FILM [3] categories medium, hard and extreme, *i.e.* 31 per difficulty level, for a total of 113 frame pairs for live-action content. To get smooth animations for the comparison, we recursively apply each method until we get a sequence of 17 frames, which we show to the users in a forward/backward loop, *i.e.* a boomerang.

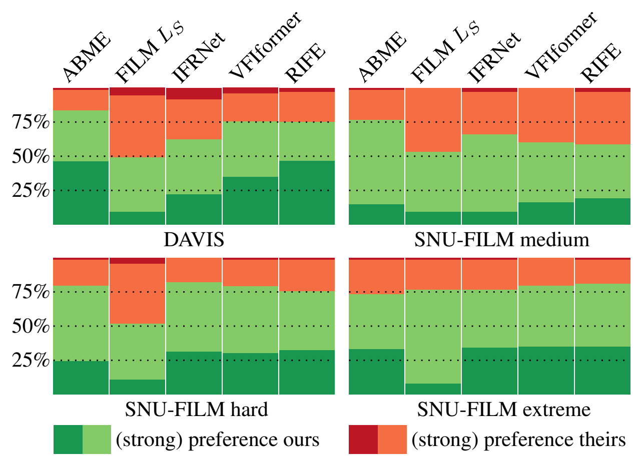


Figure 1. User study on live-action datasets. On average, users had a normal/strong preference for our method for 45/25% of all votes.

Methods. On each sample we compared our L_S approach against the following methods. ABME [10], FILM L_S [12], IFRNet (Large) [7], RIFE [6], VFIformer [8].

Results. We collected a total of 3158 AB comparisons from 69 participants for the animated movie data and 1463 votes from 33 participants for live action data. We show the results on live-action data in Fig. 1.

Visual examples. We give examples of the data used in our user study in Figs. 5 to 8.

3. Uncertainty Guidance

We have shown the PSNR improvement of our L_1 variant when replacing patches based on the color error prediction. We show the same plot in Fig. 9, but also include LPIPS and repeat the study for our L_S variant and when

[*]Work done during an internship at Disney Research

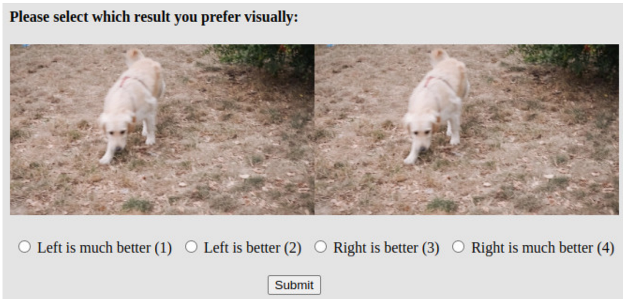


Figure 2. Screenshot of the interface of our user study showing an interpolation of [11].

using the perceptual error in terms of LPIPS for patch selection. In Figs. 10 to 13 we show the full analysis of the capabilities of our model to handle additional inputs compared to a replacement of the output patches with highest error. Fig. 3 shows a failure case of our error estimation.

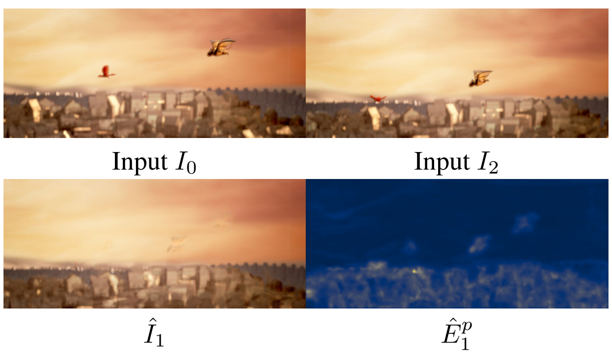


Figure 3. Failure case of the error prediction on [5]. When attempting to bridge a large 7 frame gap on full resolution, the model is no longer able to correlate the positions and only predicts an error around the original locations of the dragon/bird.

4. Ablation Study

We give a full listing of PSNR and SSIM values on all datasets for our ablation study in Tab. 1.

5. Arbitrary Time Interpolation

Our method is capable of interpolating frames at times other than $t = 1$ by rescaling the flow vectors in the cross-backward warping and the flow residual module. We show PSNR and LPIPS results for intermediate values in Fig. 4 on data from X4K1000FPS [13]. For the evaluation we use non-overlapping sequences of 9 frames, where the first and last frames are the input, and downsample the resolution to 512×270. Note, that the network was not trained on such data and does currently not take the value of t into account other than for rescaling the flow vectors. This likely leads to instabilities and a diminishing quality for values of t other than 1.

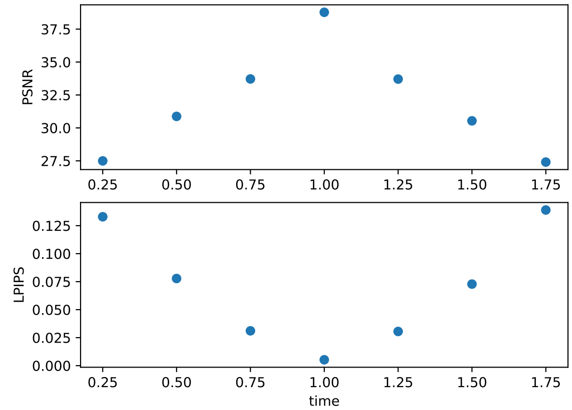


Figure 4. Interpolation results for arbitrary times between input frames at $t = 0$ and $t = 2$ on X4K1000FPS [13].

6. Implementation Details

Here, we give more details on our implementation and network architecture. We used Pytorch 1.11 for our implementation and follow their nomenclature here. All 2d convolutions use kernel size 3, unless denoted otherwise and D_l denotes the number of channels of the latent feature representation at level l ($D_0 := 67$, $D_1 := 163$, $C_{i\in\{2..6\}} := 355$). We show more details of the network architecture in Figs. 14 to 16.

References

[1] (c) copyright 2006, Blender Foundation / Netherlands Media Art Institute / www.elephantsdream.org. Elephants dream, 2006. *Licensed under Creative Commons Attribution 2.5* (https://creativecommons.org/licenses/by/2.5/). 1, 6

[2] (CC) Blender Foundation | gooseberry.blender.org. Cosmos laundromat - first cycle - 2k, 2015. *Licensed under Creative Commons Attribution-ShareAlike 4.0* (https://creativecommons.org/licenses/by-sa/4.0/). 1, 5

[3] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, Number 07, pages 10663–10671, 2020. 1

[4] Copyright (C) 2008 Blender Foundation — peach.blender.org. Big buck bunny, 2008. *Licensed under Creative Commons Attribution 3.0* (http://creativecommons.org/licenses/by/3.0/). 1, 4

Error Est.	Deep Features	Shared Frames	Vimeo90k PSNR	Vimeo90k SSIM	Big Buck Bunny PSNR	Big Buck Bunny SSIM	Cosmos Laundromat PSNR	Cosmos Laundromat SSIM	Elephants Dream PSNR	Elephants Dream SSIM	Sintel PSNR	Sintel SSIM
✔	✔	✔	36.34	0.9814	35.98	0.9815	34.55	0.9407	35.25	0.9680	37.25	0.9697
✔	✘	✔	36.28	0.9812	35.18	0.9800	34.08	0.9398	34.71	0.9656	36.27	0.9679
✘	✔	✔	36.31	0.9813	35.89	0.9813	34.56	0.9418	35.19	0.9678	37.23	0.9699
✘	✔	✘	35.82	0.9796	35.20	0.9794	34.49	0.9409	34.74	0.9656	36.86	0.9677
✘	✘	✘	35.76	0.9793	34.98	0.9789	34.38	0.9405	34.62	0.9645	34.59	0.9675

Table 1. Full listing of our ablation study of our network design.

[5] Copyright (c) Blender Foundation — durian.blender.org. Sintel, 2010. *Licensed under Creative Commons Attribution 3.0* (`http://creativecommons.org/licenses/by/3.0/`). 1, 2, 7

[6] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2021. 1

[7] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. IFRNet: Intermediate feature refine network for efficient frame interpolation. *arXiv preprint arXiv:2205.14620*, 2022. 1

[8] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022. 1

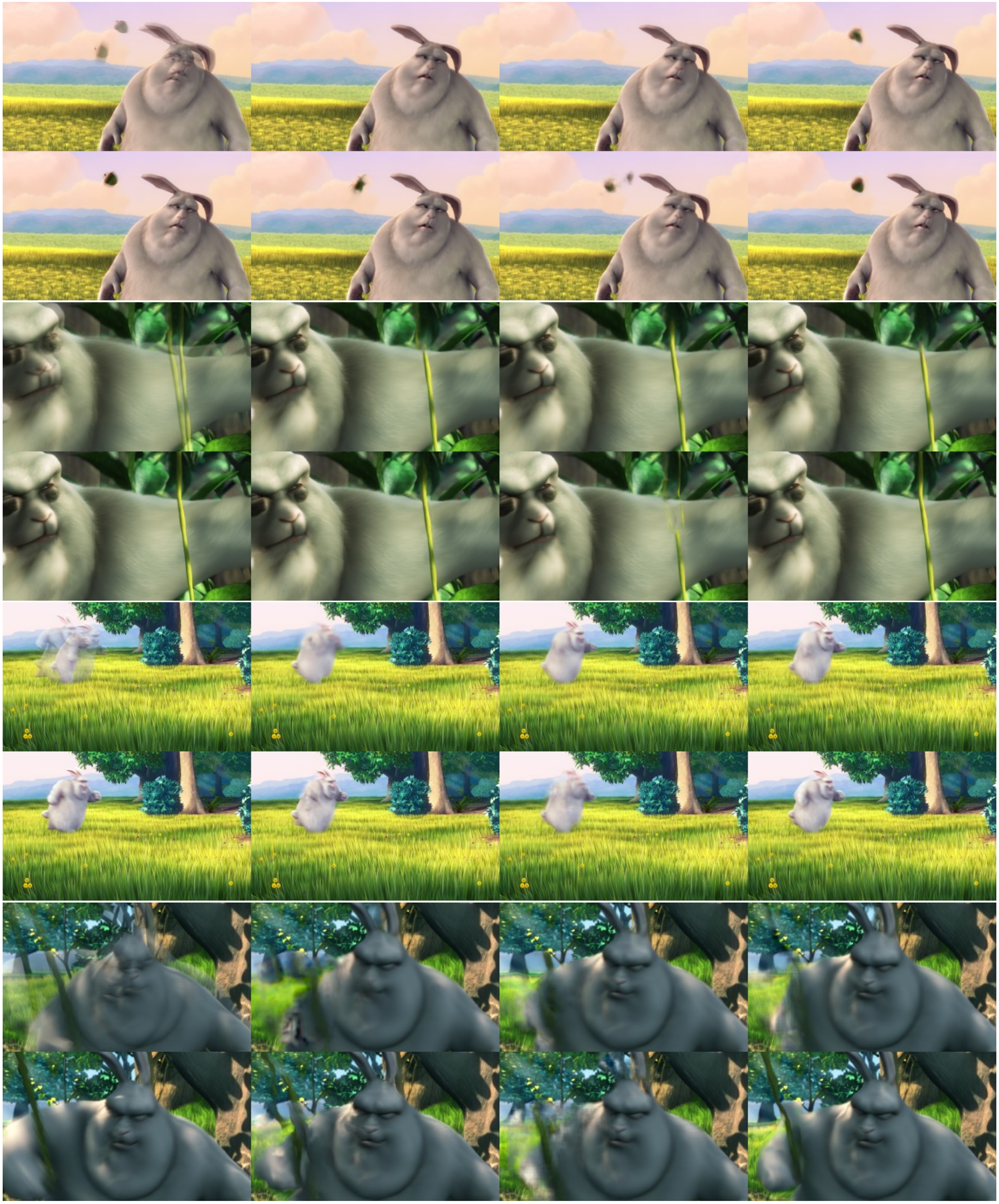
[9] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 1

[10] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14539–14548, 2021. 1

[11] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 1, 2

[12] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *European Conference on Computer Vision*, 2022. 1

[13] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14489–14498, 2021. 2



Inputs
GTruth

ABME
FILM L_S

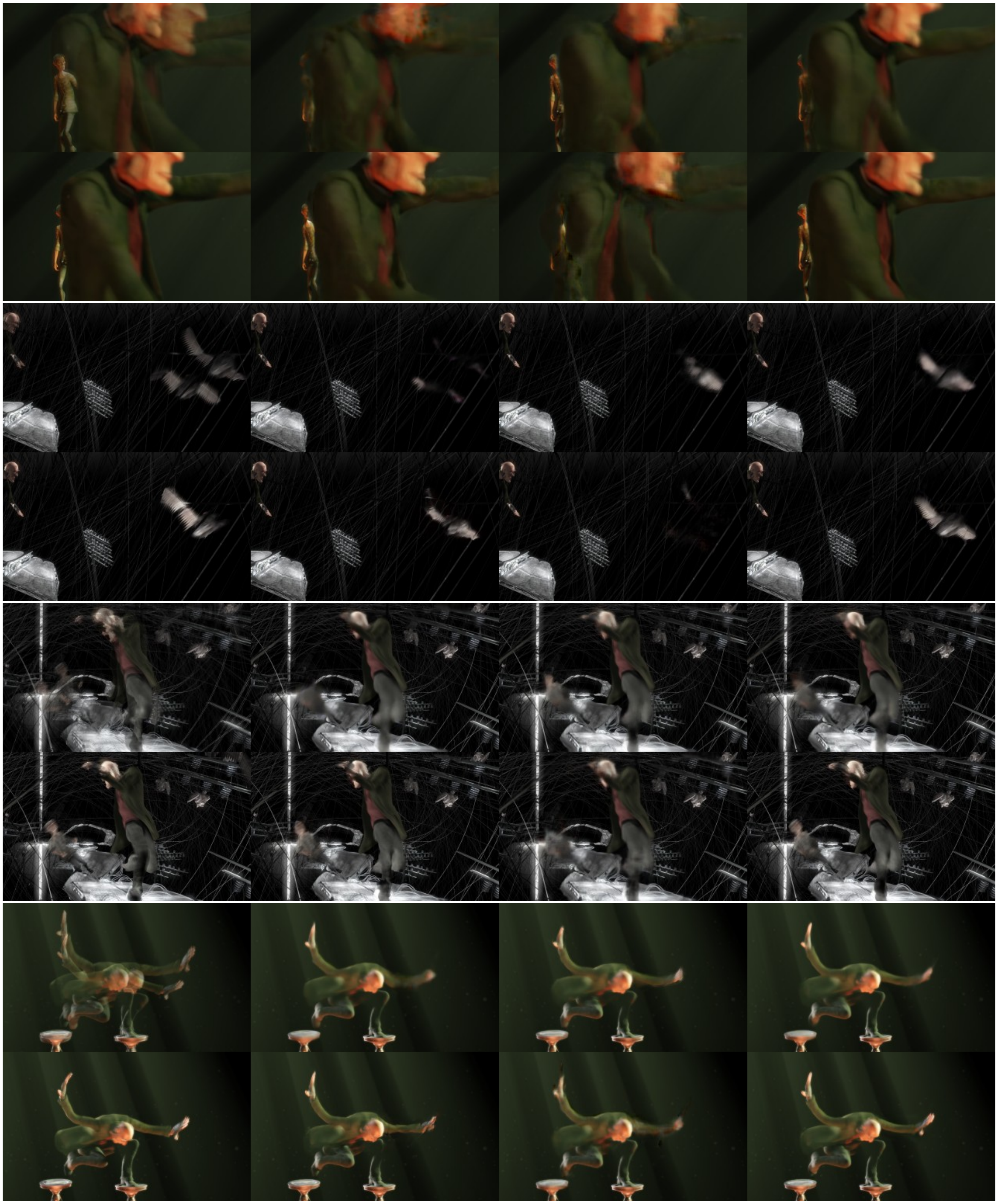
IFRNet
RIFE

VFIformer
Ours L_S

Figure 5. Samples from the Big Buck Bunny [4] user study.

Inputs GTruth | ABME FILM L_S | IFRNet RIFE | VFIformer Ours L_S

Figure 6. Samples from the Cosmos Laundromat [2] user study.

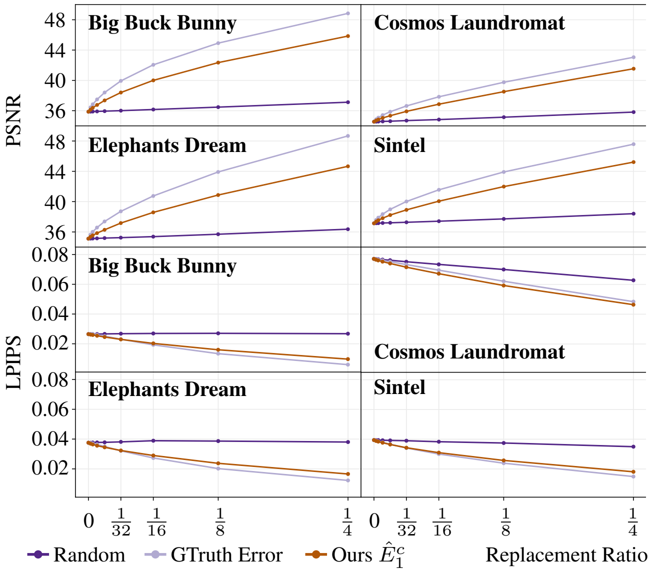


Inputs	ABME	IFRNet	VFIformer
GTruth	FILM L_S	RIFE	Ours L_S

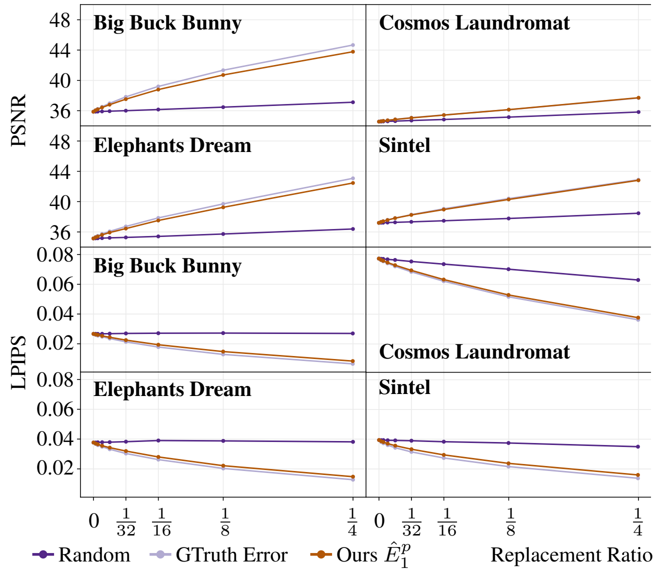
Figure 7. Samples from the Elephants Dream [1] user study.

Inputs	ABME	IFRNet	VFIformer
GTruth	FILM L_S	RIFE	Ours L_S

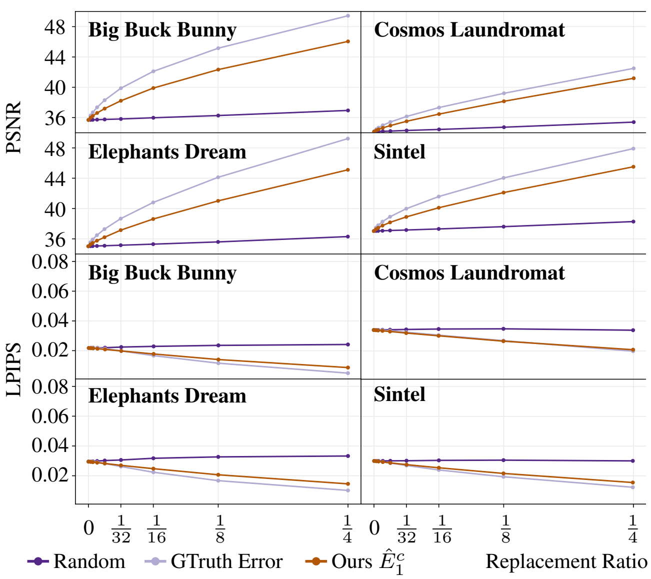
Figure 8. Samples from the Sintel [5] user study.



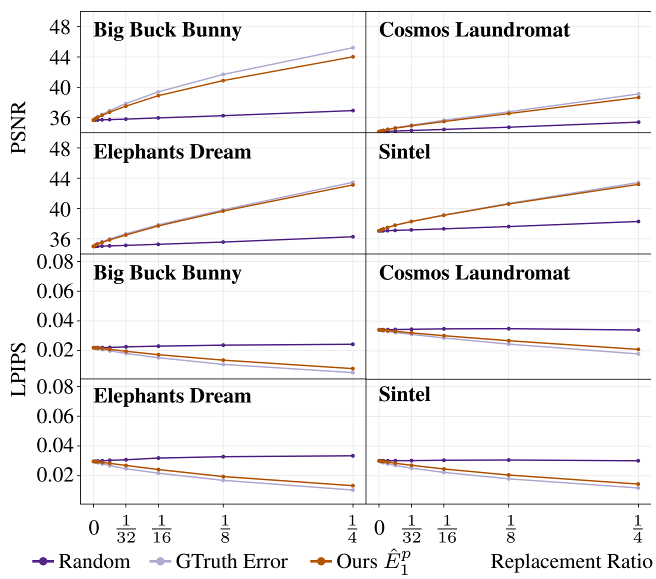
(a) Replacement of patches using the color error prediction of our L_1 variant compared to ground truth $L2$ error.



(b) Replacement of patches using the perceptual error prediction of our L_1 variant compared to ground truth LPIPS error.



(c) Replacement of patches using the color error prediction of our L_S variant compared to ground truth $L2$ error.



(d) Replacement of patches using the perceptual error prediction of our L_S variant compared to ground truth LPIPS error.

Figure 9. Evaluation of our error prediction.

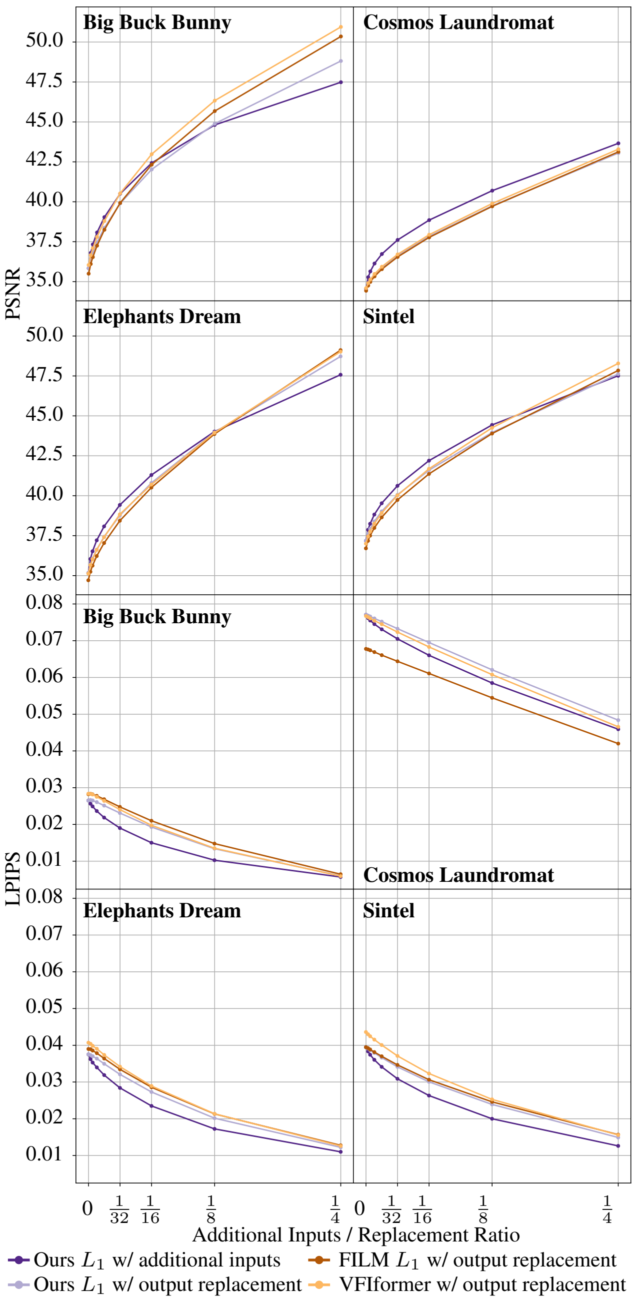


Figure 10. We evaluate how our L_1 variant handles additional inputs compared to a replacement of the output. We use $L2$ error to select patches and show the replacement of outputs from FILM L_1 and VFIformer for comparison.

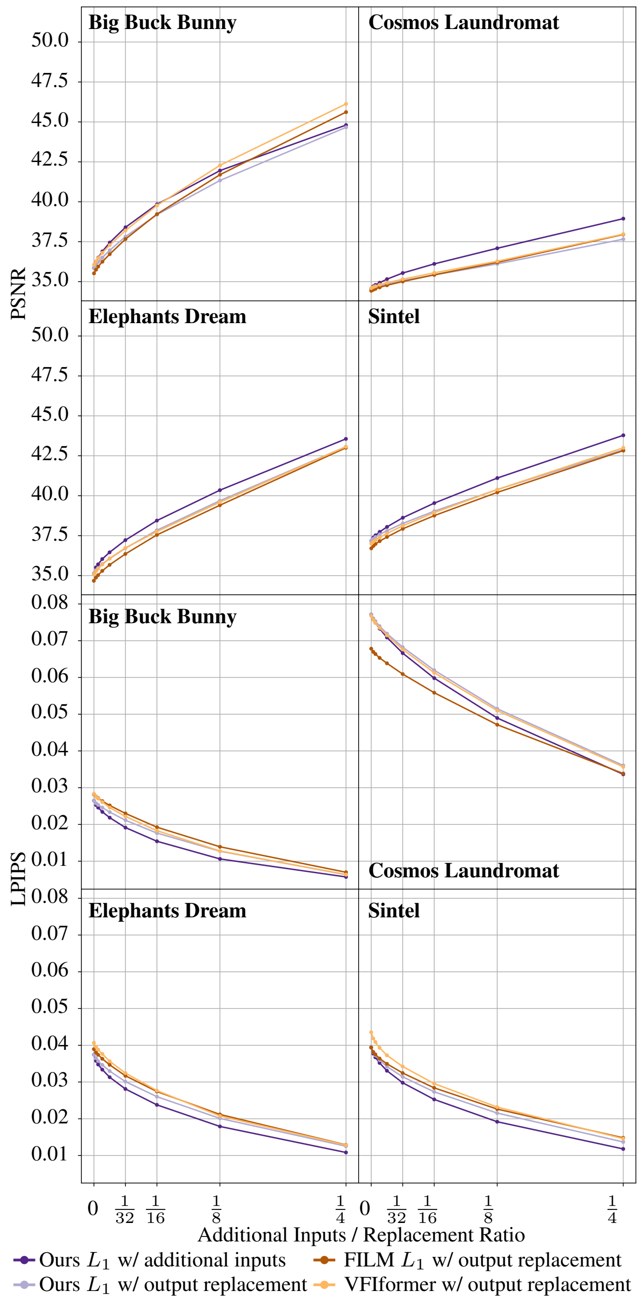


Figure 11. We evaluate how our L_1 variant handles additional inputs compared to a replacement of the output. We use LPIPS error to select patches and show the replacement of outputs from FILM L_1 and VFIformer for comparison.

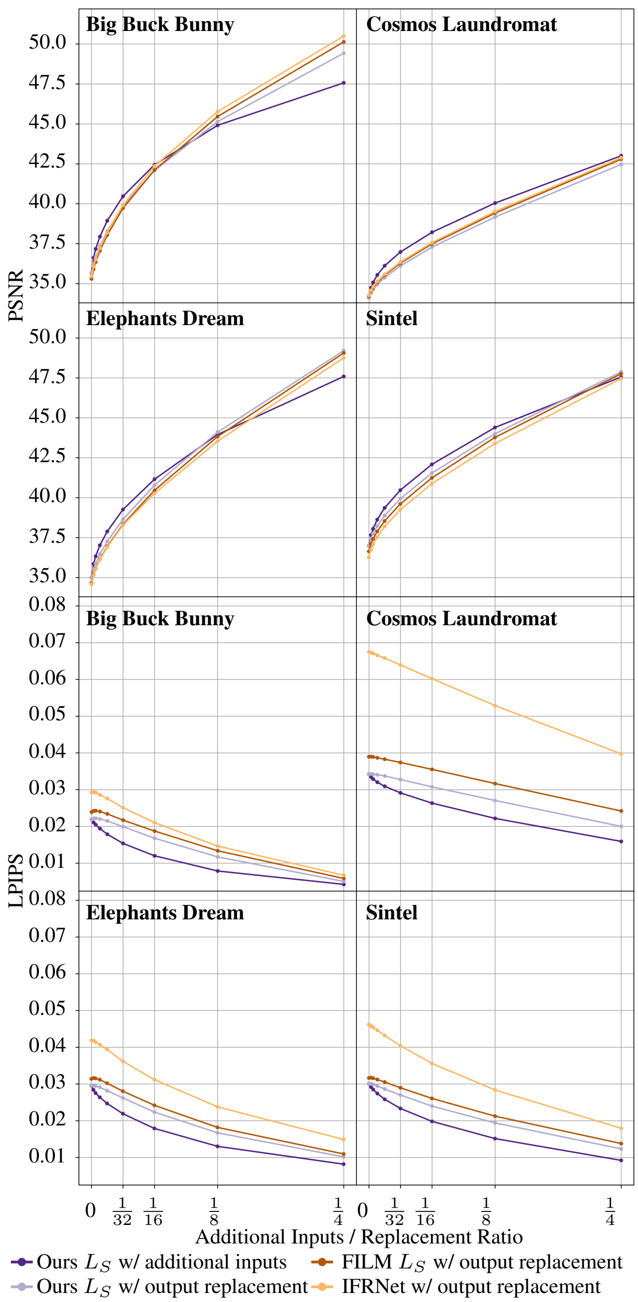


Figure 12. We evaluate how our L_S variant handles additional inputs compared to a replacement of the output. We use $L2$ error to select patches and show the replacement of outputs from FILM L_S and IFRNet for comparison.

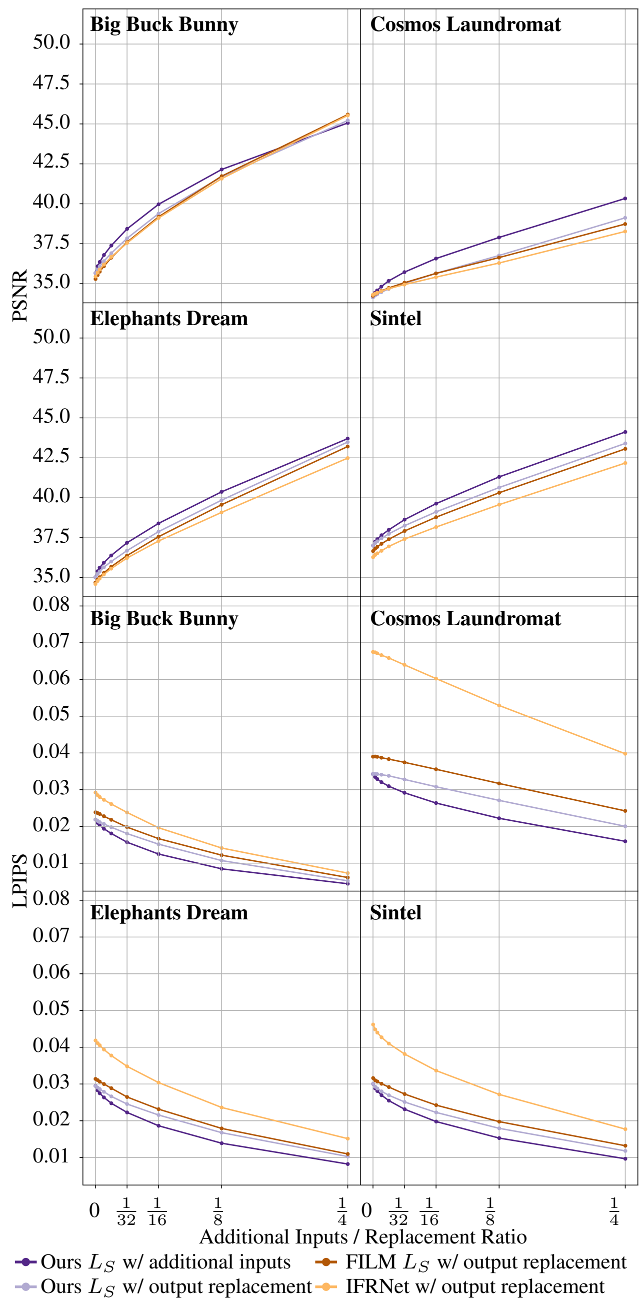


Figure 13. We evaluate how our L_S variant handles additional inputs compared to a replacement of the output. We use LPIPS error to select patches and show the replacement of outputs from FILM L_S and IFRNet for comparison.

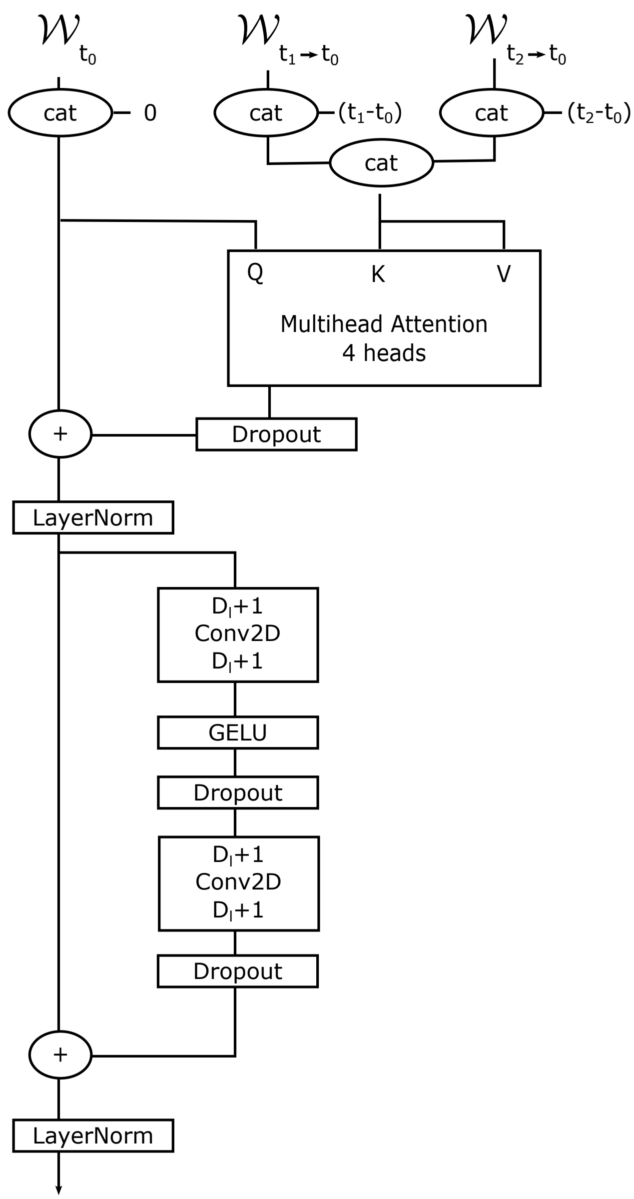


Figure 14. Architecture of the MACE block. Note that the image tensors of shapes (B, C, H, W) and $(B, C, 2, H, W)$ need to be reshaped into $(1, BHW, C)$ and $(2, BHW, C)$ for the multihead attention module.

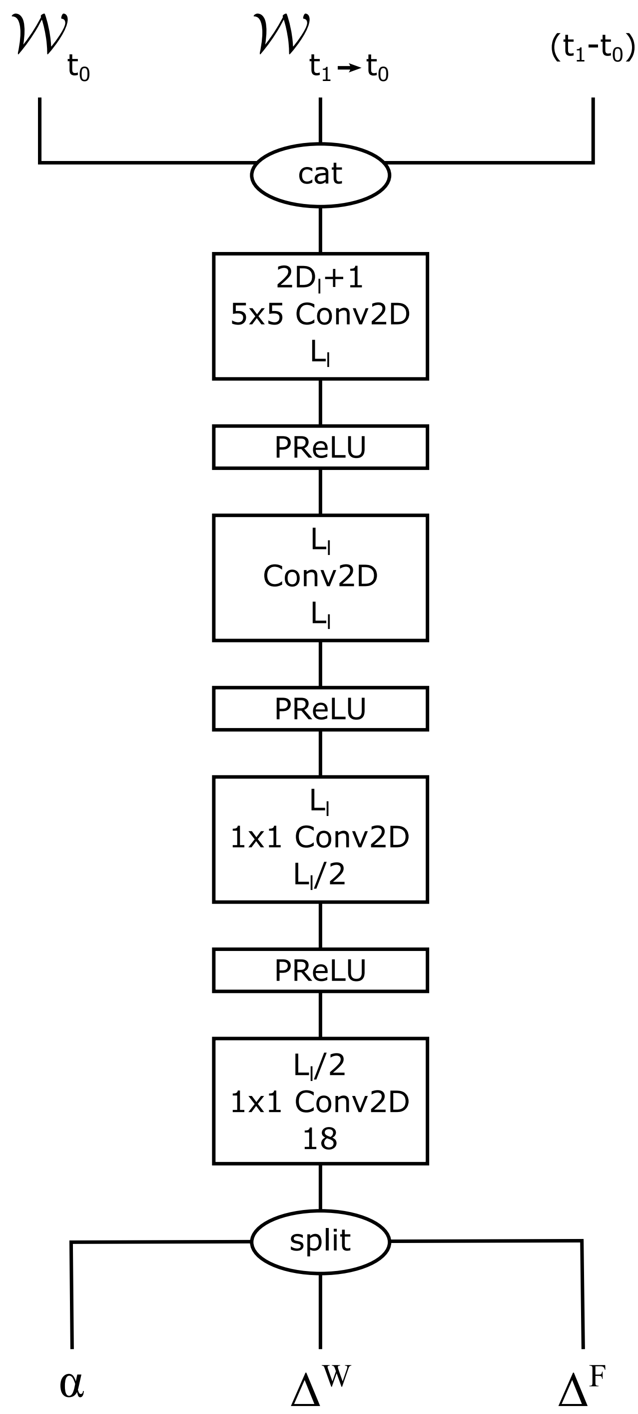


Figure 15. Architecture of the flow and context residual module. $L_1 := 128$ and $L_{i\in\{2..6\}} := 256$

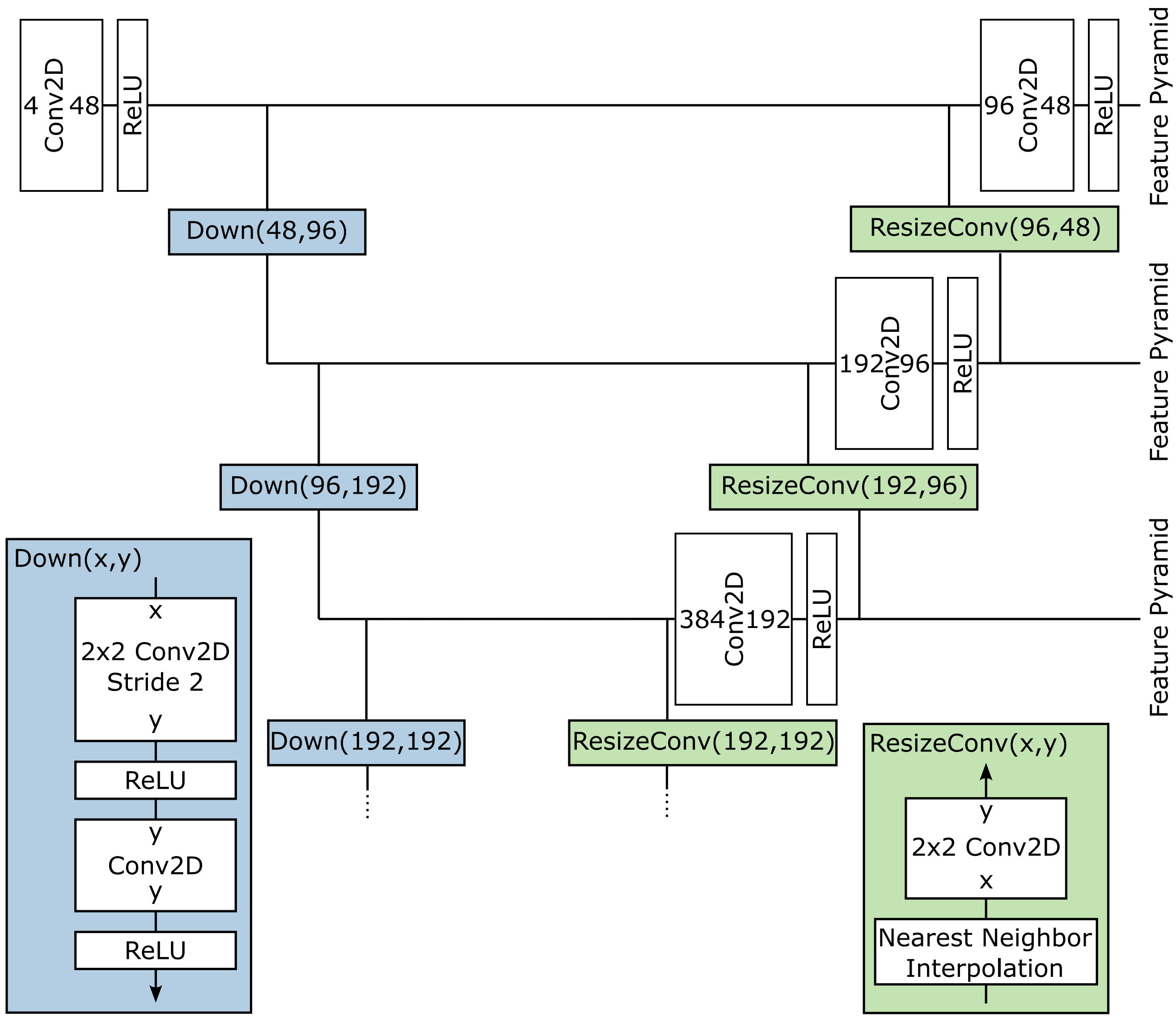


Figure 16. Architecture of the deep feature extraction. We repeat the last layer 3 more times as indicated by the dotted lines.