Streamable Neural Fields - Supplementary

A Training Algorithms

Alg. 1 and Alg. 2 shows the pseudo-code of our progressive and slimmable training described in section 3.1 in our main paper.

Algorithm 1 Progressive training

```
Require: Inputs $x$, targets $y$
$\theta = \{\}$
while not done do
    $\theta_{\text{new}} \leftarrow$ GrowNetwork()
    Initialize $\theta_{\text{new}}$
    $\theta \leftarrow \theta \cup \{\theta_{\text{new}}\}$
    for epoch = 0 to $n_{steps}$ do
        Predict signal values $\hat{y} = f_{\theta}(x)$
        Compute loss $\mathcal{L}(y, \hat{y})$
        Compute gradients $\nabla_{\theta_{\text{new}}}\mathcal{L}$
        Update $\theta_{\text{new}}$
    end for
end while
```

Algorithm 2 Slimmable training

```
Require: Inputs $x$, targets $y$
Require: Parameters $\{\theta_{w_1}, \dots, \theta_{w_K}\}$
for epoch = 0 to $n_{steps}$ do
    $\theta = \{\}$
    for $i = 1$ to $K$ do
        $\theta \leftarrow \theta \cup \{\theta_{w_i}\}$
        Predict signal values $\hat{y} = f_{\theta}(x)$
        Compute loss $\mathcal{L}(y, \hat{y})$
        Compute gradients $\nabla_{\theta}\mathcal{L}$
        Accumulate gradients
    end for
    Update $\theta_{w_i}, \forall i \in \{1, ..., K\}$
end for
```

B Ablation Studies

Weight initialization In this section, we demonstrate the effectiveness of our proposed initialization method. While leaving the lateral connection as SIREN initialization [4], our method assigns zero to other newly added weights. The newly added zero weights encourage the larger sub-network to start training when the network output is close to the output of the smaller pre-trained sub-network. We trained two same *streamable (progressive)* models on every image in the Kodak dataset but initialized the newly added weights differently. Fig. 1 shows that our initialization method supports faster convergence. Also, we empirically found that our method converges at a lower loss compared to the SIREN initialization. Furthermore, careful weight initialization is crucial when using sine activation to maintain the activation and gradient distribution at each layer [4]. Results shown in Fig. 3 imply that adopting our initialization method does not harm the SIREN activation and gradient statistics.

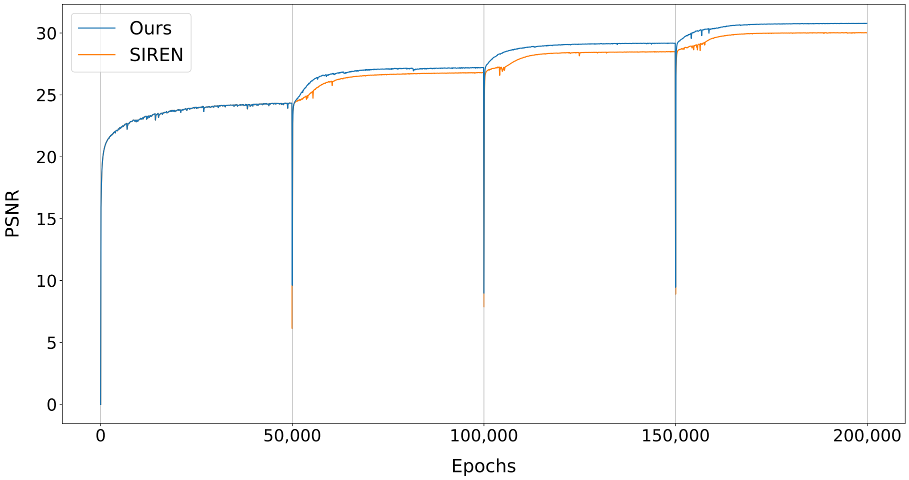

Fig. 1. Averaged training loss and PSNR curve of two distinct weight initialization methods. We made the networks grow at every 50,000 epochs. Our proposed initialization makes the model converge at higher PSNR.

Training stability We visualized the optimization trajectories of a *streamable (progressive)* and an *individual* model on 3D loss surfaces to better understand how our model keeps the training process stable. Following Li et al. [2], we applied PCA to matrix $M = [\theta_0 - \theta_n; \ldots; \theta_{n-1} - \theta_n]$ and used the two leading principal components as direction vectors: δ and η. Note that θ_i denotes network parameters at epoch i [2]. One can plot a loss surface by moving the final parameters along the two directions and computing loss values. More formally, we plot a function $f(\alpha, \beta) = L(\theta_n + \alpha\delta + \beta\eta)$, where α and β are step sizes. To plot the trajectories, we sampled the parameters every 5 epochs, computed the loss values, and projected them onto the loss surfaces.

As shown in the first column of Fig. 2, the loss surface of each model's smallest network has no extreme non-convexities. However, the trajectories of the *individual* model's larger networks traverse through non-convex regions (the second row), while the larger networks of the *streamable (progressive)* model search for minima predominantly in low-loss regions (the first row). This suggests that the knowledge (i.e., parameters) learned by the smaller sub-network gives a good initial point. Since the larger network starts from a point where the loss value is already low, it converges fast and maintains the training process stable.

C Experimental Details and Additional Results

We provide training details and additional results that are not specified in the main paper. We show the metric values of each plot in the main paper along

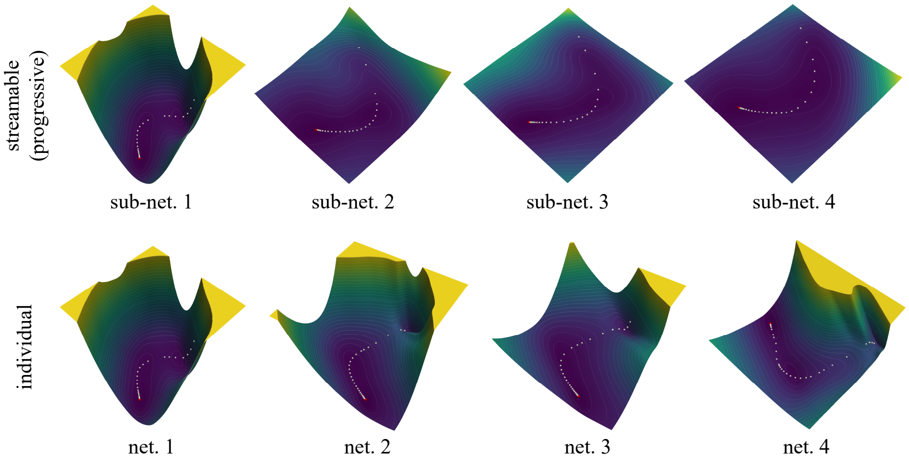

Fig. 2. Visualization of loss surface and training trajectory. The *individual* models' surface is more complex than that of *streamable (progressive)* models. Each model is trained on Kodak image 19.

with qualitative results. For every table, bold faces represent the best metric of each sub-network.

Spectral growing For 1D sinusoidal function reconstruction, we manufactured a target function $f(x) = \sum_{i=1}^{10} \sin(2\pi k_i x + \phi_i)$. We set $k_i \in \{5, 10, ..., 50\}$, $\phi_i \sim U(0, 2\pi)$ and sampled x uniformly over $[0, 1]$. We used an MLP of three hidden layers with sine activations [4]. We trained using Adam optimizer with a learning rate of 10^{-4}. Starting with a width of 10 (the channel size of hidden layers), we gradually increased the size up to 40. Each sub-network was trained for 150 epochs. For image experiment, each model has five layers with sine activations and no activation on the final layer. We used Adam [1] with a learning rate of 2×10^{-4} and trained for 50,000 epochs. For SDF experiment, We used learning rate of 10^{-4} and trained for 15,000 epochs. The same loss function described in [4] is used. Other configurations are same as image experiment.

Table. 1 and Table. 2 shows the quantitative results of spectral growing of images and 3D shapes respectively. Qualitative results are depicted in Fig. 5 and Fig. 6. To show the growing frequency more directly, we fit an 800×800 sunflower image illustrated in Fig. 4. The second row shows the frequency domain of images reconstructed by *streamable (progressive)* neural fields. A larger network reconstructs higher frequency components that are not captured by smaller sub-networks. We used the same network architecture and training settings used in Kodak experiments.

Spatial/temporal growing For image spatial growing experiment, we used same configurations used in image spectral growing experiment. For video temporal growing experiment, we used six-layer MLP and learning rate of 10^{-4}. Other configurations are same as image experiment.

Quantitative results of baseline comparison experiments of spatial/temporal growing are shown in Table. 3 and Table. 4 respectively. Qualitative results are depicted in Fig. 7 and Fig. 8. To obtain a longer temporal growing video, we fit the "bikes sequence" video, which consists of 272 × 640 pixels and 240 frames using the same network architecture and training settings used in the UVG dataset [3] experiments. We provided the reconstructed video in *.mp4* format to show the result.

sub-network	metrics	individual	streamable (slimmable [5])	streamable (progressive)
1	params.	3,903	3,903	3,903
	PSNR↑	24.39	23.72	24.39
	SSIM↑	0.597	0.578	0.597
	LPIPS↓	0.559	0.582	0.559
2	params.	15,003	15,003	15,057
	PSNR↑	27.22	25.64	**27.30**
	SSIM↑	0.701	0.643	**0.704**
	LPIPS↓	0.380	0.462	**0.298**
3	params.	33,303	33,303	33,507
	PSNR↑	**29.78**	27.29	29.29
	SSIM↑	**0.795**	0.704	0.775
	LPIPS↓	0.246	0.372	**0.190**
4	params.	58,803	58,803	58,681
	PSNR↑	**31.19**	29.19	30.91
	SSIM↑	**0.836**	0.770	0.825
	LPIPS↓	0.149	0.271	**0.129**

Table 1. Quantitative results of spectral growing on 24 Kodak images.

sub-network	shapes	individual	streamable (progressive)
1	params.	66,897	66,897
	armadillo	0.321	0.321
	dragon	0.104	0.104
	happy buddha	0.109	0.109
2	params.	133,873	133,981
	armadillo	0.137	**0.032**
	dragon	0.128	**0.030**
	happy buddha	0.054	**0.019**
3	params.	198,388	198,657
	armadillo	0.059	**0.023**
	dragon	0.056	**0.025**
	happy buddha	0.042	**0.018**

Table 2. Chamfer distance of spectral growing 3D shapes.

sub-network	metrics	individual	streamable (slimmable [5])	streamable (progressive)
1	params.	14,517	14,517	14,517
	PSNR↑	30.04	24.49	30.04
	SSIM↑	0.820	0.606	0.820
	LPIPS↓	0.190	0.508	0.190
2	params.	29,034	29,067	29,095
	PSNR↑	28.82	24.99	**29.62**
	SSIM↑	0.801	0.640	**0.821**
	LPIPS↓	0.202	0.442	**0.148**
3	params.	43,551	44,307	44,515
	PSNR↑	29.14	26.07	**29.55**
	SSIM↑	0.807	0.686	**0.816**
	LPIPS↓	0.190	0.388	**0.155**
4	params.	58,068	58,803	58,453
	PSNR↑	29.65	27.58	**29.91**
	SSIM↑	**0.819**	0.734	0.817
	LPIPS↓	0.177	0.307	**0.149**

Table 3. Quantitative results of spatial growing on 8 Kodak images.

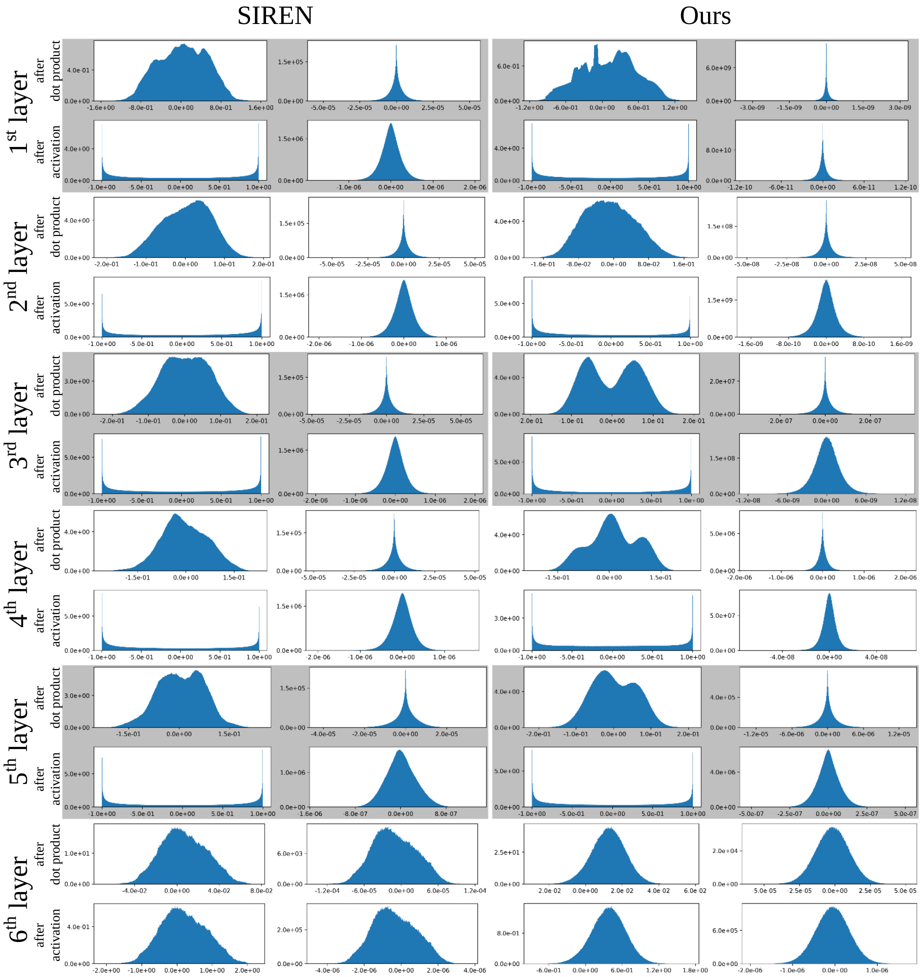

Fig. 3. Activation and gradient statistics of SIREN and our initialization after 5 training epochs. Along with SIREN, our initialization method also preserves the distribution. Note that our model has no activation on the output layer.

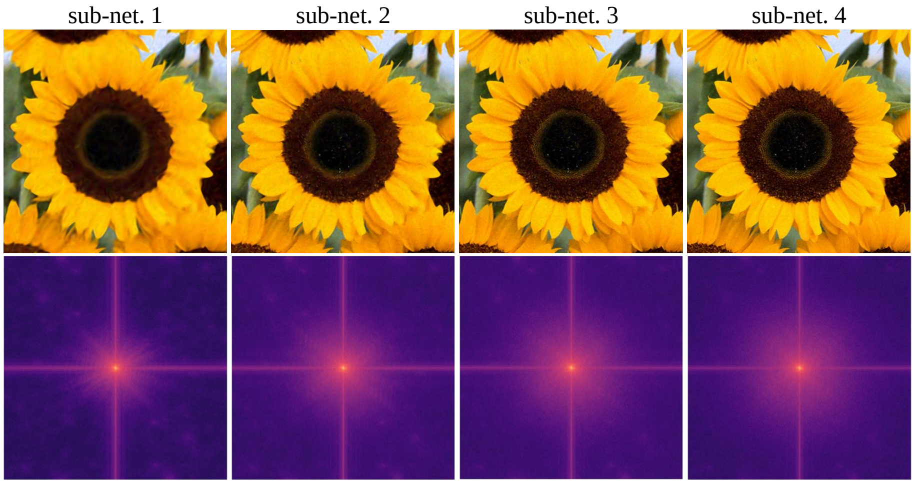

Fig. 4. Frequency spectrum of spectral growing images represented by *streamable (progressive)* neural fields. As the network grows, high-frequency components are gradually reconstructed (darker is smaller).

sub-network	metrics	individual	streamable (slimmable [5])	streamable (progressive)
1	params.	441,635	441,635	441,635
	PSNR↑	37.20	32.97	37.27
	SSIM↑	0.957	0.890	0.957
	LPIPS↓	0.023	0.105	0.023
2	params.	883,270	882,836	882,116
	PSNR↑	37.49	34.17	**37.99**
	SSIM↑	**0.959**	0.912	0.958
	LPIPS↓	0.022	0.058	**0.020**
3	params.	1,324,905	1,316,867	1,317,096
	PSNR↑	37.60	34.54	**38.22**
	SSIM↑	**0.960**	0.914	0.958
	LPIPS↓	0.022	0.047	**0.017**

Table 4. Quantitative results of temporal growing on 7 UVG videos.

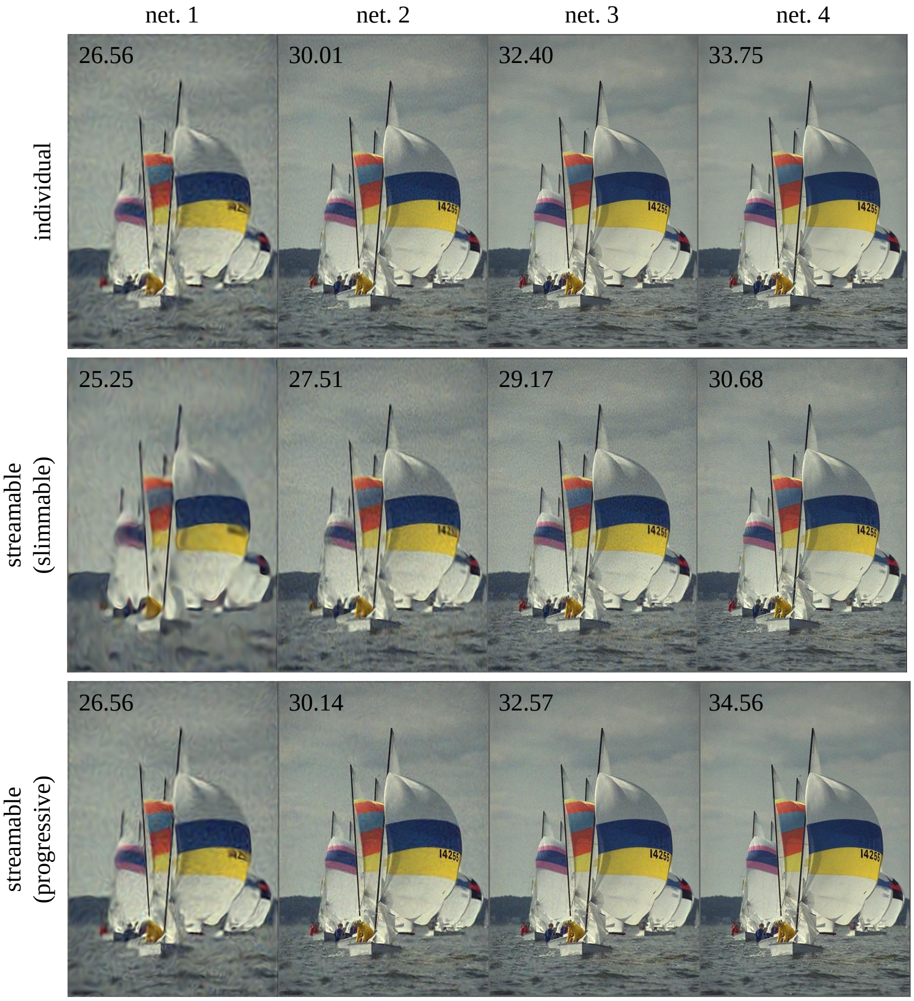

Fig. 5. Qualitative results and PSNRs of spectral growing on Kodak image 9. *Streamable (progressive)* neural field has the same or higher representation power compared to individually trained models.

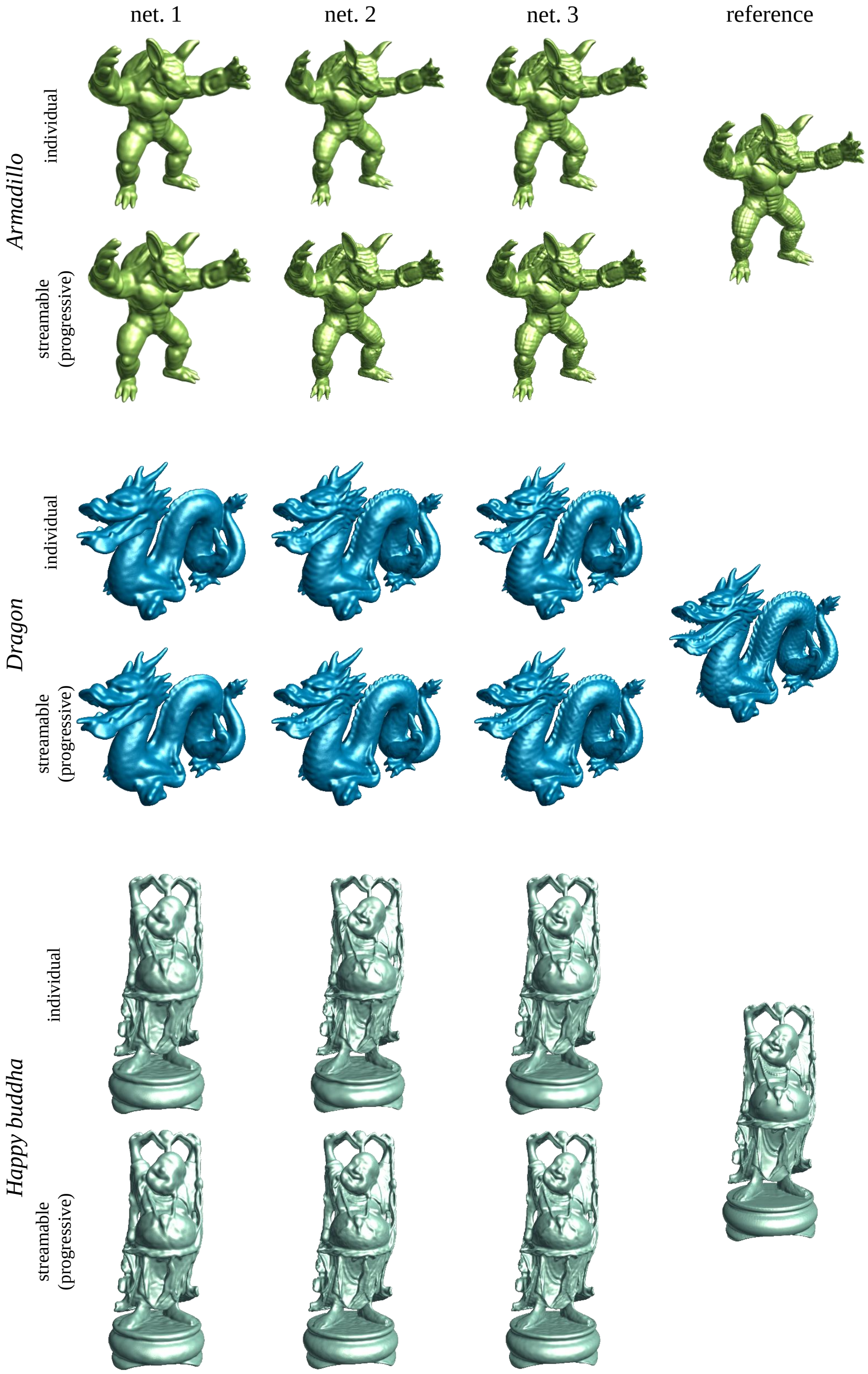

Fig. 6. Qualitative results of spectral growing on 3D shapes.

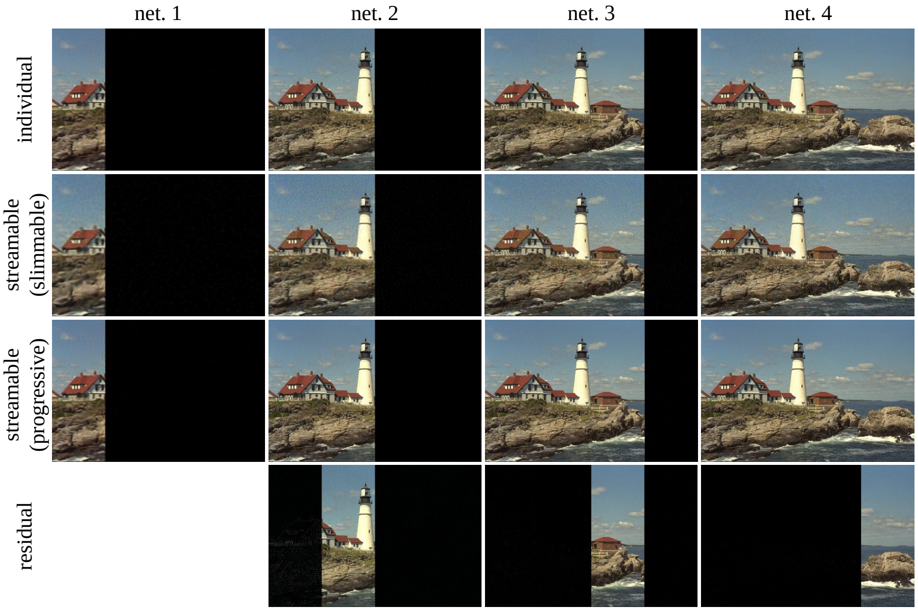

Fig. 7. Qualitative result of spatial growing on Kodak image 21.

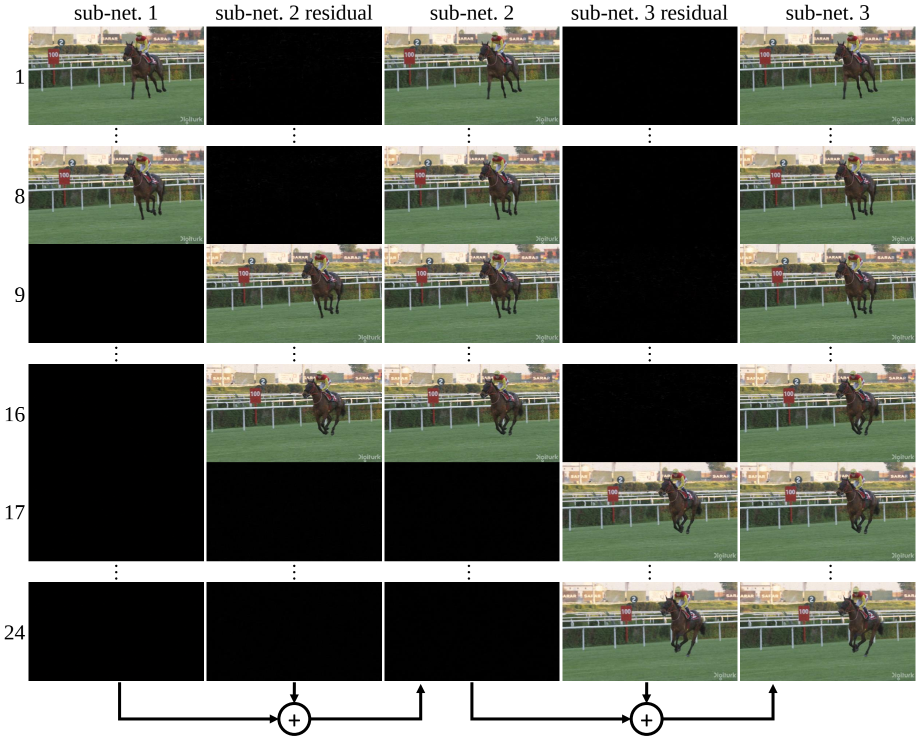

Fig. 8. Qualitative result of temporal growing on video (*Jockey*). A larger sub-network reconstructs the exact residual frames that are not represented by the smaller ones.

References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
2. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. In: Advances in Neural Information Processing Systems. vol. 31, pp. 6391–6401. Curran Associates, Inc. (2018)
3. Mercat, A., Viitanen, M., Vanne, J.: UVG dataset: 50/120fps 4k sequences for video codec analysis and development. In: Proceedings of the 11th ACM Multimedia Systems Conference, MMSys. pp. 297–302. ACM (June 2020)
4. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Advances in Neural Information Processing Systems. vol. 33, pp. 7462–7473. Curran Associates, Inc. (2020)
5. Yu, J., Yang, L., Xu, N., Yang, J., Huang, T.: Slimmable neural networks. In: International Conference on Learning Representations (2019)