Supplementary Materials to :
BVINet: Unlocking Blind Video Inpainting with Zero Annotations

1. Network Structure

In the section, we provide the architectures of the encoder and the decoder for mask prediction network (MPNet) and video completion network (VCNet) in Tab.1 and Tab.2, respectively.

Table 1. The architecture of the encoder. We use "same padding" for each convolution layer, and ReLU is added to the end of each convolution layer.

Stage	Output	Input	Architecture
MPNet			
E_{1_1}	$3 \times 216 \times 120$	$3 \times 432 \times 240$	DWT
E_{1_2}	$64 \times 216 \times 120$	$3 \times 216 \times 120$	3×3 $conv$ $ReLU$
E_{1_3}	$64 \times 216 \times 120$	$64 \times 216 \times 120$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 2$
E_{2_1}	$64 \times 108 \times 60$	$64 \times 216 \times 120$	DWT
E_{2_2}	$128 \times 108 \times 60$	$64 \times 108 \times 60$	3×3 $conv$ $ReLU$
E_{2_3}	$128 \times 108 \times 60$	$128 \times 108 \times 60$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 2$
VCNet			
E_{1_1}	$64 \times 432 \times 240$	$3 \times 432 \times 240$	3×3 $conv$ $ReLU$
E_{1_2}	$64 \times 432 \times 240$	$64 \times 432 \times 240$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$
E_{2_1}	$64 \times 216 \times 120$	$64 \times 432 \times 240$	DWT
E_{2_2}	$128 \times 216 \times 120$	$64 \times 216 \times 120$	3×3 $conv$ $ReLU$
E_{2_3}	$128 \times 216 \times 120$	$128 \times 216 \times 120$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$
E_{3_1}	$128 \times 108 \times 60$	$128 \times 216 \times 120$	DWT
E_{3_2}	$256 \times 108 \times 60$	$128 \times 108 \times 60$	3×3 $conv$ $ReLU$
E_{3_3}	$256 \times 108 \times 60$	$256 \times 108 \times 60$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$

2. About Dataset

In Tab. 3, we provide a detailed comparison between the datasets used to train the non-blind baselines [12–16] and those used to train our BVINet. From the table, we can observe that: 1) Existing datasets typically use pixel value 0 to fill the corrupted regions, treating it as the corrupted content. 2)They often employ static, fixed-shape masks or target masks with relatively simple motion to define the corrupted regions. 3) There is a clear boundary between the corrupted and valid regions. In this way, the synthesized corrupted video inherently introduce specific prior knowledge, such as content, border, and shape. These priors make corrupted regions easily locatable from video frame by a deep neural network or even a simple linear classifier, re-

Table 2. The architecture of the decoder. Similar to the encoder, we also use "same padding" for each convolution layer, and ReLU is added to the end of each convolution layer.

Stage	Output	Input	Architecture
MPNet			
D_{1_1}	$64 \times 216 \times 120$	$128 \times 108 \times 60$	$Upsample(2)$ 3×3 $conv$ $ReLU$
D_{1_2}	$64 \times 216 \times 120$	$64 \times 216 \times 120$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 2$
D_{2_1}	$64 \times 432 \times 240$	$64 \times 216 \times 120$	$Upsample(2)$ 3×3 $conv$ $ReLU$
D_{2_2}	$64 \times 432 \times 240$	$64 \times 432 \times 240$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 2$
D_{2_3}	$1 \times 432 \times 240$	$64 \times 432 \times 240$	3×3 $conv$ $Sigmoid$
VCNet			
D_{1_1}	$128 \times 108 \times 60$	$256 \times 108 \times 60$	3×3 $conv$ $ReLU$
D_{1_2}	$128 \times 108 \times 60$	$128 \times 108 \times 60$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$
D_{2_1}	$128 \times 216 \times 120$	$128 \times 108 \times 60$	$IDWT$
D_{2_2}	$64 \times 216 \times 120$	$128 \times 216 \times 120$	3×3 $conv$ $ReLU$
D_{2_3}	$64 \times 216 \times 120$	$64 \times 216 \times 120$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$
D_{3_1}	$64 \times 432 \times 240$	$64 \times 216 \times 120$	$IDWT$
D_{2_2}	$64 \times 432 \times 240$	$64 \times 432 \times 240$	$\begin{bmatrix} 3 \times 3 \ conv \\ ReLU \\ 3 \times 3 \ conv \end{bmatrix} \times 5$
D_{2_3}	$3 \times 432 \times 240$	$64 \times 432 \times 240$	3×3 $conv$ $ReLU$

Table 3. Comparison with related datasets in video inpainting. "Syn" and "Real" denote synthesized and real-world.

Methods	Content	Shape	Border	Syn	Real
VINet [1, 2]	0	fixed	clear	✔	✘
FGVC [3]	0	fixed/object	clear	✔	✘
CPVINet [4]	0	fixed	clear	✔	✘
STTN [5]	0	fixed/object	clear	✔	✘
FuseFormer [6]	0	fixed/object	clear	✔	✘
E2FGVI [7]	0	fixed/object	clear	✔	✘
FGT [8]	0	fixed/object	clear	✔	✘
ProPainter [9]	0	fixed/object	clear	✔	✘
DiffuEraser [10]	0	fixed/object	clear	✔	✘
WaveFormer [11]	0	fixed/object	clear	✔	✘
Ours	image	free-form strokes	blur	✔	✔

sulting in the blind video inpainting task being degraded into a non-blind category [17]. In contrast, our customized dataset employs free-form strokes with varied shapes and complex motions as the corrupted regions, fills real image patches as the corrupted content, and utilizes iterative Gaussian smoothing [18] to blur the boundaries. Such strategy effectively simulates the blind video inpainting setting. In addition, we also collect $1,250$ bullet removal video clips in a real-world scenario.

3. More Results on Synthesized Dataset

In addition to Section 4 of the main paper, we provide more results obtained using the proposed methods in Fig. 1. As can be observed, our method can obtain spatial-temporally consistent inpainted results without any mask annotations.

4. More Results on Real Cases

Subtitle removal is one of the important applications of blind video inpainting. To verify the effectiveness of our method, we use state-of-the-art two methods as our baselines to evaluate the blind video inpainting ability of our model, including one non-blind image inpainting method GNet [19], one video restoration method RAVUNet [20]. To ensure the fairness of the experimental results, these baselines are fine-tuned on our subtitle removal dataset using their released models and codes. Fig· 2 compares the results of bullet removal between OGNet [19], RAVUNet [20], and our method. As shown in Fig. 2, our method effectively eliminates bullets in videos without the need for any mask annotations, and generates better details than the baselines.

5. User Study

To provide a comprehensive comparison, we conducted a user study to evaluate the inpainting results. we invite 15 volunteers and present them 5 diverse inpainted video. In each trial, the inpainting results of different models are shown to volunteers, and the volunteers are required to choose the best one. The results are summarized in Fig. 3. It is evident that the volunteers show a clear preference for our inpainted results compared to other competitors.

6. More Results on MPNet

Some examples of corrupted regions predicted by our MPNet are shown in Fig. 4. As revealed in Fig. 4, the corrupted regions predicted by the full MPNet model are closer to ground-truth. This demonstrates that the effectiveness of MPNet.

References

[1] Dahun Kim, Sanghyun Woo, Joon-Young Lee, et al. Deep blind video decaptioning by temporal aggregation and recurrence. In *Proc. CVPR*, pages 4263–4272, 2019. 2

[2] Dahun Kim, Sanghyun Woo, Joon-Young Lee, et al. Recurrent temporal aggregation framework for deep video inpainting. *IEEE TPAMI*, 42(5):1038–1052, 2020. 2

[3] Chen Gao, Ayush Saraf, Jia-Bin Huang, et al. Flow-edge guided video completion. In *Proc. ECCV*, pages 713–729, 2020. 2

[4] Sungho Lee, Seoung Wug Oh, DaeYeun Won, et al. Copy-and-paste networks for deep video inpainting. In *Proc. ICCV*, pages 4413–4421, 2019. 2

[5] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *Proc. ECCV*, pages 3723–3732, 2020. 2

[6] Rui Liu, Hanming Deng, Yangyi Huang, et al. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proc. ICCV*, pages 14040–14049, 2021. 2

[7] Zhen Li, Cheng-Ze Lu, Jianhua Qin, et al. Towards an end-to-end framework for flow-guided video inpainting. In *Proc. CVPR*, pages 17562–17571, 2022. 2

[8] Kaidong Zhang, Jingjing Fu, and Dong Liu. Flow-guided transformer for video inpainting. In *Proc. ECCV*, pages 74–90, 2022. 2

[9] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, et al. Propainter: Improving propagation and transformer for video inpainting. In *Proc. ICCV*, pages 10477–10486, 2023. 2

[10] Xiaowen Li, Haolan Xue, Peiran Ren, et al. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025. 2

[11] Zhiliang Wu, Changchang Sun, Hanyu Xuan, et al. Waveformer: Wavelet transformer for noise-robust video inpainting. In *Proc. AAAI*, pages 6180–6188, 2024. 2

[12] Zhiliang Wu, Kang Zhang, Hanyu Xuan, et al. DAPC-Net: Deformable alignment and pyramid context completion networks for video inpainting. *IEEE SPL*, 28:1145–1149, 2021. 1

[13] Zhiliang Wu, Changchang Sun, Hanyu Xuan, et al. Divide-and-conquer completion network for video inpainting. *IEEE TCSVT*, 33(6):2753–2766, 2023.

[14] Yanni Zhang, Zhiliang Wu, and Yan Yan. PFTA-Net: Progressive feature alignment and temporal attention fusion networks for video inpainting. In *Proc. ICIP*, pages 191–195, 2023.

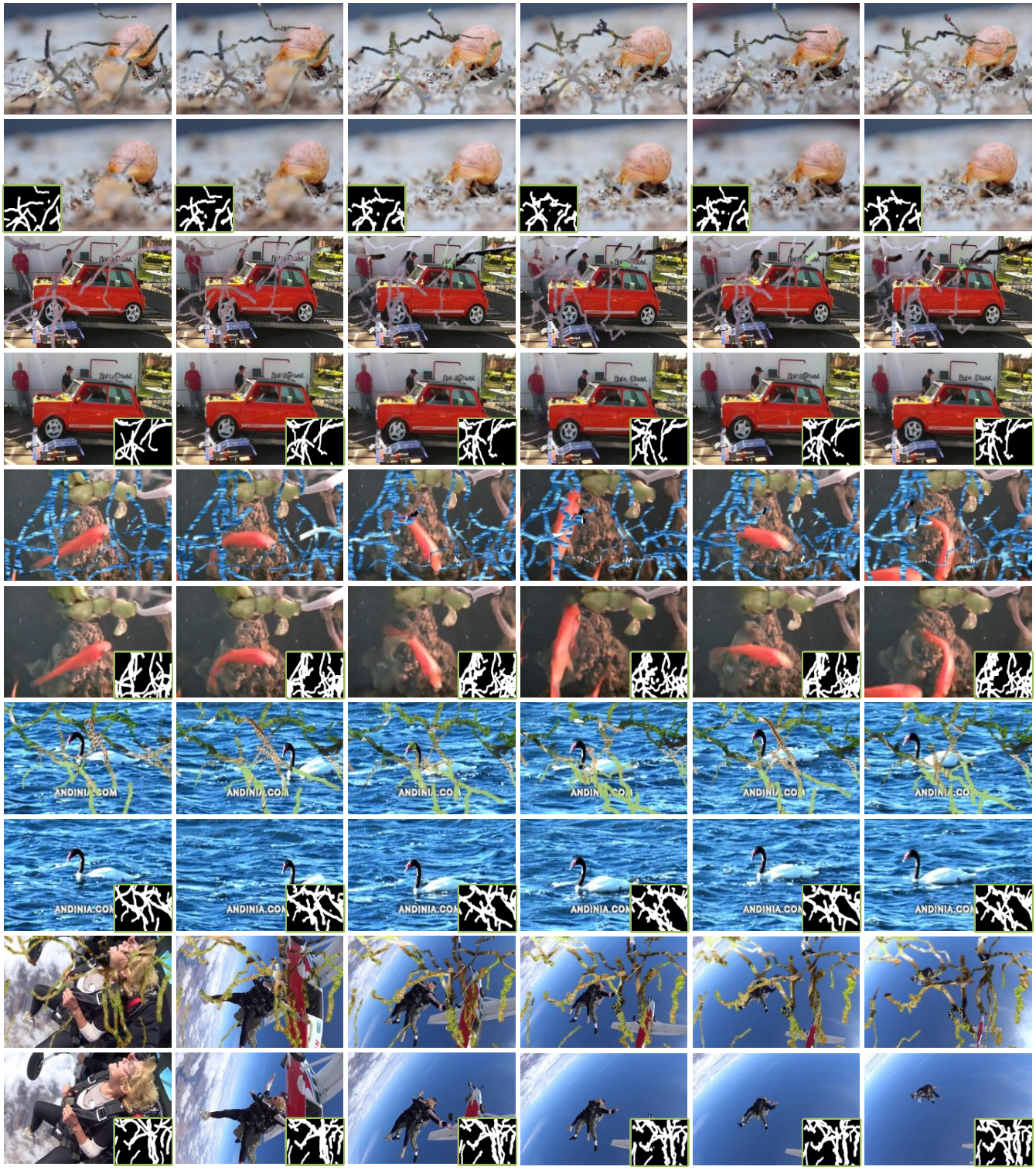

Figure 1. Some example of inpainting results with our method. The top row shows corrupted video frame. The completed results are shown in the bottom row, where green box denotes the mask generated by the model.

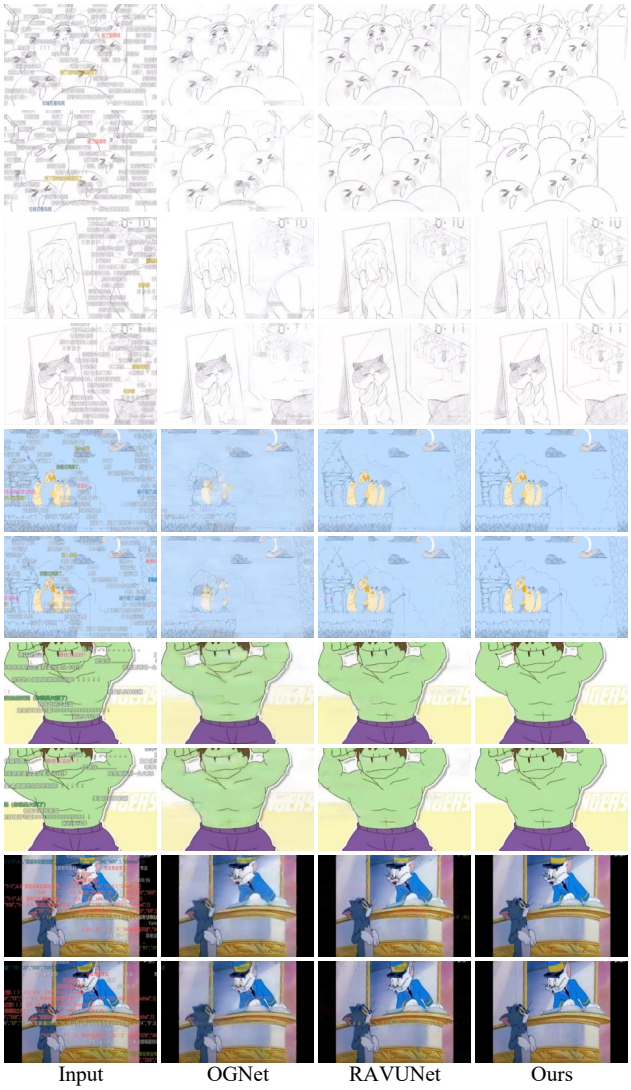

Figure 2. Qualitative results of subtitle removal. Better viewed at zoom level 400%.

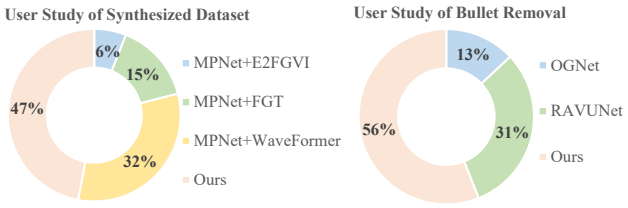

Figure 3. User preference results.

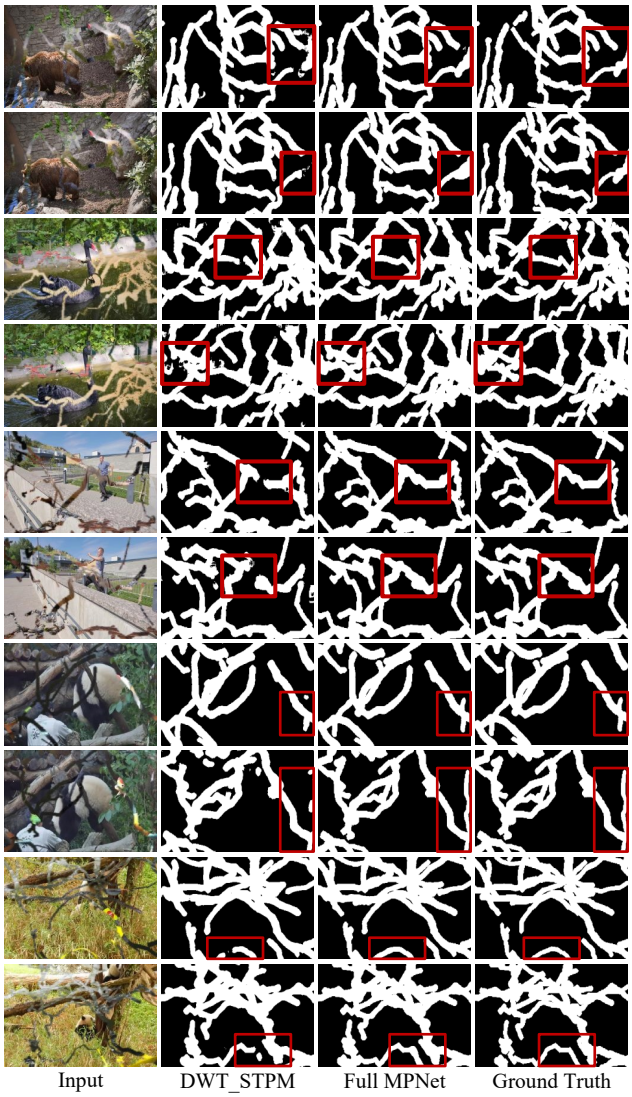

Figure 4. Example of corrupted regions predicted by our MPNet. Better viewed at zoom level 400%.

[15] Zhiliang Wu, Kang Zhang, Changchang Sun, et al. Flow-guided deformable alignment network with self-supervision for video inpainting. In *Proc. ICASSP*, pages 1–5, 2023.

[16] Jianan Wang, Hanyu Xuan, and Zhiliang Wu. Semantic-guided completion network for video inpainting in complex urban scene. In *Proc. PRCV*, page 224–236, 2023. 1

[17] Zhiliang Wu, Hanyu Xuan, Changchang Sun, et al. Semi-supervised video inpainting with cycle consistency constraints. In *Proc. CVPR*, pages 22586–22595, 2023. 2

[18] Yi Wang, Xin Tao, Xiaojuan Qi, et al. Image inpainting via generative multi-column convolutional neural networks. In *Proc. NeurIPS*, 2018. 2

[19] Shruti S Phutke, Ashutosh Kulkarni, Santosh Kumar Vipparthi, et al. Blind image inpainting via omni-dimensional gated attention and wavelet queries. In *Proc. CVPR Workshops*, pages 1251–1260, 2023. 2

[20] Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, et al. Restoration of analog videos using swin-unet. In *Proc. ACMMM*, pages 6985–6987, 2022. 2