

Supplementary Material

A. Network Architecture

In this work, we remove the subtitles from the video with two steps: caption mask extraction and video decaptioning. The detailed architecture of the caption mask extraction network is reported in Table 1. Input_Feat refers to the source of the feature. K_Size refers to the kernel size of the operators. Stride denotes the stride of the operators, which is omitted if this parameter is inapplicable. Num_Chans denotes the channel number of the output feature map. BN indicates whether a batch normalization layer is used after the operator. Act_Func refers to the non-linear function after the layer.

Module	Input_Feat	K_Size	Stride	Num_Chans	BN	Act_Func
Conv0	Subtitled_frames	3	1	64	T	ReLU
Maxpool0	F_Conv0	2	2	64	F	None
Conv1	F_Maxpool0	3	1	128	T	ReLU
Maxpool1	F_Conv1	2	2	128	F	None
Conv2	F_Maxpool1	3	1	256	T	ReLU
Maxpool2	F_Conv2	2	2	256	F	None
Conv3	F_Maxpool2	3	1	512	T	ReLU
Maxpool3	F_Conv3	2	2	512	F	None
Conv4	F_Maxpool3	3	1	512	T	ReLU
Upsample1	F_Conv4	-	-	512	F	None
DeConv1	Cat[F_Upsample1, F_Conv3]	3	1	256	T	ReLU
Upsample2	F_DeConv1	-	-	256	F	None
DeConv2	Cat[F_Upsample2, F_Conv2]	3	1	128	T	ReLU
Upsample3	F_DeConv2	-	-	128	F	None
DeConv3	Cat[F_Upsample3, F_Conv1]	3	1	64	T	ReLU
Upsample4	F_DeConv3	-	-	64	F	None
DeConv4	Cat[F_Upsample4, F_Conv0]	3	1	64	T	ReLU
OutConv	F_DeConv4	1	1	1	F	Sigmoid

Table 1: The architecture of caption mask extraction network.

Table 2 shows the architecture of the video decaptioning network. We use gated convolution instead of ordinary convolution and adopt Sigmoid as the gated activation function. We do not use batch normalization in this network, and the negative slope of all LeakyReLU layers is set to 0.2. As shown in Fig. 1, we also visualize the learned feature maps of the decaptioning network. The key component of our decaptioning network is the frame attention module that aggregates the valid parts of the input reference feature maps to the target feature map, and the detailed architecture of this layer is shown in Table 3. Unlike the general attention mechanism, we do not use the embedding operation before calculating the attention score, but directly divide the input feature map into patches according to Patch_Size. To deal with subtitles of various sizes, we simultaneously calculate the attention scores in four different Patch_Size values and concatenate them to obtain the final attention result through

Module	Input_Feat	K_Size	Stride	Num_Chan	Act_Func
GatedConv0	Cat[Subtitled_frames, Masks]	3	1	64	Leaky_ReLU
GatedConv1	F_GatedConv0	3	2	128	Leaky_ReLU
GatedConv2	F_GatedConv1	3	2	256	Leaky_ReLU
GatedConv3	F_GatedConv2	3	2	256	Leaky_ReLU
Frame_Atten1	F_GatedConv3	-	-	256	Leaky_ReLU
Frame_Atten2	F_Frame_Atten1	-	-	256	Leaky_ReLU
Frame_Atten3	F_Frame_Atten2	-	-	256	Leaky_ReLU
Frame_Atten4	F_Frame_Atten3	-	-	256	Leaky_ReLU
GatedConv4	Cat[F_Frame_Atten4, F_GatedConv3]	3	1	256	Leaky_ReLU
GatedDeConv1	F_GatedConv4	3	2	256	Leaky_ReLU
GatedConv5	Cat[F_GatedDeConv1, F_GatedConv2]	3	1	256	Leaky_ReLU
GatedDeConv2	F_GatedConv5	3	2	128	Leaky_ReLU
GatedConv6	Cat[F_GatedDeConv2, F_GatedConv1]	3	1	128	Leaky_ReLU
GatedDeConv3	F_GatedConv6	3	2	64	Leaky_ReLU
GatedConv7	Cat[F_GatedDeConv3, F_GatedConv0]	3	1	64	Leaky_ReLU
OutConv	F_GatedConv7	3	1	3	None

Table 2: The architecture of video decaptioning network.

convolution.

Module	Input_Feat	Patch_Size	K_Size	Stride	Act_Func
Attention	F_GatedConv3	(1, 1)	-	-	None
Attention	F_GatedConv3	(2, 2)	-	-	None
Attention	F_GatedConv3	(4, 4)	-	-	None
Attention	F_GatedConv3	(8, 8)	-	-	None
Conv	All_Attention	-	3	1	Leaky_ReLU
GatedConv	Cat[F_Conv, F_GatedConv3]	-	3	1	Leaky_ReLU

Table 3: The architecture of frame attention module.

B. More Ablation Studies

This section presents more ablation studies on our network.

B.1. Effect of Filtering and Thresholding

Unlike the general text detection algorithms, our caption mask extraction network can mark the subtitles and their backgrounds together in the video (as shown in Fig. 2). Some noise points are eliminated by the filtering, and our automatic annotation (the fourth column) is good enough as the supervision information. In addition, our caption mask extraction network can also predict the right location (the last column).

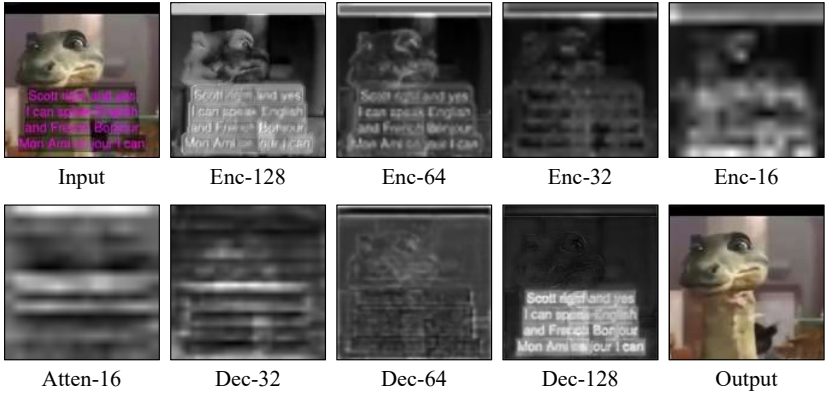


Figure 1: Visualization of learned feature activation. For the visualization, we average each feature maps along with the channel dimension, perform the zero-one normalization, and up-sample to 128×128 . The numbers in the labels denote the spatial resolution of the feature maps. We observe the hierarchical attention operations throughout the network. It can also be seen that the gated convolution can make the network pay more attention to the subtitled area.

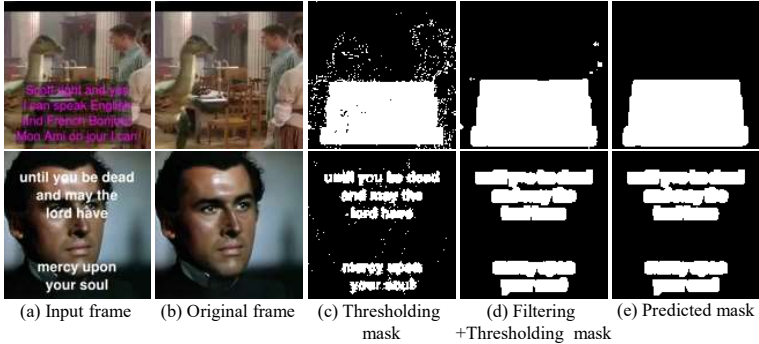


Figure 2: Subtitle mask extraction. From left to right: subtitled frames, original frames, masks after thresholding, masks after filtering and thresholding, and predicted masks by our network.

B.2. Effect of Number of Input Reference Frames

In this paper, we take a “multi-to-one” approach to removing the subtitles in the video. The number of input frames T directly relates to the input batch, enabling the control of the amount of temporal information to be considered at once. As shown in Table 4, we compare the results of four different input frame values. In this experiment, we set the stride of the input frames to 3. The performance tends to be good, with generally large input frames, while the value of 7 provides the best results.

B.3. Effect of Stride of Input Reference Frames

We also compare the results of four different video frame sampling strides. Table 5 shows the ablation studies on stride s . We finally find that a stride of 3 performs the best in our

	Value	MSE↓	PSNR↑	DSSIM↓
Number of frames	3	0.0011	35.0188	0.0318
	5	0.0011	35.0234	0.0317
	7	0.0011	35.0251	0.0317
	9	0.0011	35.0250	0.0317

Table 4: The ablation studies on the number of input frames.

	Value	MSE↓	PSNR↑	DSSIM↓
Stride	1	0.0011	35.0185	0.0318
	2	0.0011	35.0234	0.0317
	3	0.0011	35.0251	0.0317
	4	0.0011	35.0250	0.0317

Table 5: The ablation studies on the stride of input frames.

experiment. Using $T = 7$, our model has an approximately 21 frame-term view range.

B.4. Effect of Loss Function

In the training process, we calculate the L1 loss of the subtitle and non-subtitle areas separately according to the extracted mask, so that the network can focus on the subtitles. To validate the effectiveness of the extracted mask on the network, we do not distinguish whether it is a subtitle area when calculating the loss. Table 6 reports the experimental results, showing that the extracted mask has a noticeable improvement in the final result.

Loss Function	MSE↓	PSNR↑	DSSIM↓
Without Mask	0.0012	34.6722	0.0326
With Mask	0.0011	35.0251	0.0317

Table 6: The ablation studies on the extracted mask.

C. More Results

In this section, more quantitative and qualitative comparisons and results are presented.

To further compare our frame-attention-based decaptioning network with two recent transformer-based video inpainting models, *i.e.*, DSTT [24] and STTN [25], we directly replace the second part of our video decaptioning framework with DSTT and STTN separately. Table 7 and Fig. 3 show the comparisons of the above two methods, BVDNet, and our proposed method. From Table 7, we can find that our method still achieves the best decaptioning performance. Compared with the results of DSTT and STTN in Table 1 in the Manuscript, the performance in Table 7 is better. The reason is that the direct removing subtitle regions lead to information loss, which also validates the superiority of our generic decaptioning framework with the automatic subtitle mask extraction. In Fig. 3, our method generates the superior visual results. For DSTT and STTN, they cannot well remove subtitles with a solid background (see last two rows in Fig. 3). An important reason is that there exists the subtitles in the input image, the direct convolution and attention score calculation will lead to

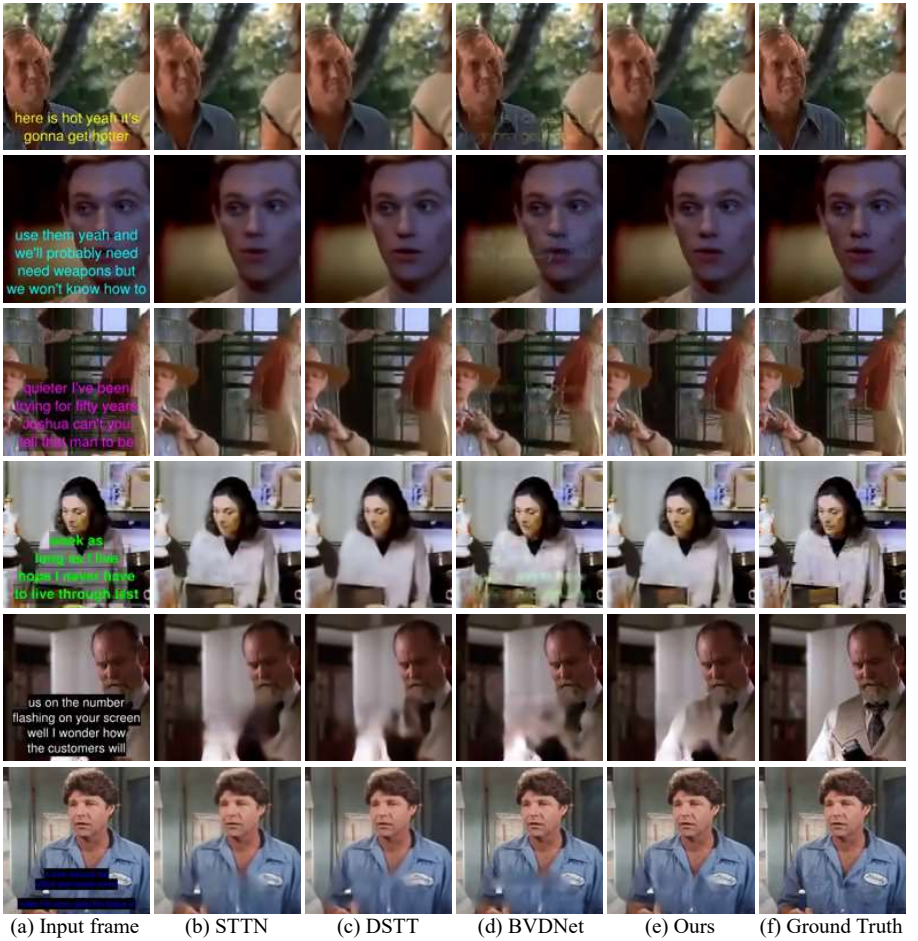


Figure 3: More comparison results on ChaLearn 2018 LAP Inpainting Track2 validation set. From left to right: input subtitled frames, results of STTN [9], DSTT [10], BVDNet [11], our model and ground truth.

the interference of subtitles in the generated results. With the help of the extracted mask and gated convolution, our method can better extract the effective features of the input image and calculate the attention score more reasonably and accurately, so as to better remove the subtitles. In addition, Fig. 4 illustrates more visual results of caption mask extraction and video decaptioning of our method. All these results demonstrate the effectiveness of our proposed method.

As shown in Fig. 5, we also test the decaptioning effect of subtitles with outlines. Since this type of caption does not appear in the dataset, our method fails to tackle it, but it is still able to extract the subtitle mask correctly. In addition, we test the decaptioning effect on scenes with more motion under the subtitles, and the results are reported in Fig. 6. Because we use a sliding window method to process the input data, our model can also restore the area obscured by the subtitles when the scene in the video changes greatly.

Method	MSE↓	PSNR↑	DSSIM↓	LPIPS↓	VFID↓
STTN	0.0012	32.8672	0.0393	0.0777	0.8231
DSTT	0.0011	33.4428	0.0372	0.0751	0.8162
BVDNet	0.0013	34.1275	0.0365	0.0529	0.8001
Ours	0.0011	35.0251	0.0317	0.0497	0.6995

Table 7: Quantitative comparisons of our method with STTN [8], DSTT [2], and BVDNet [10]. ↑: higher is better; ↓: lower is better.

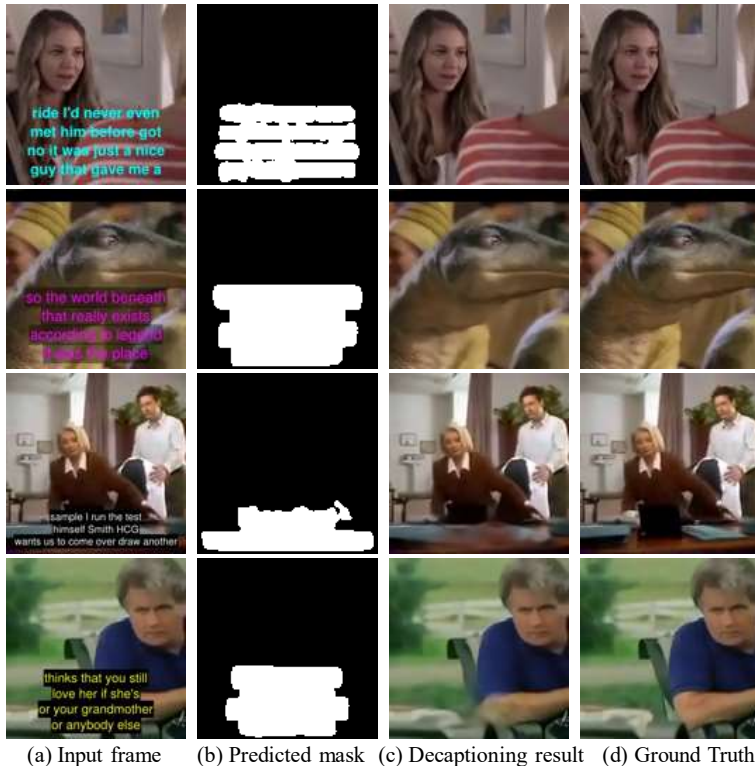


Figure 4: More visual results. From left to right: input subtitled frames, predicted subtitle masks, decaptioning results of our model and ground truth.

Furthermore, we evaluate the performance of our frame-attention-based decaptioning network (the second part of decaptioning framework) on the video inpainting task. We conduct a qualitative and quantitative comparison with two recent transformer-based video inpainting methods, *i.e.*, STTN and DSTT, on the YouTube-VOS dataset. In Table 8, the performance of our method is not best, but competitive. The main reason may be that we choose a "multi-to-one" repair strategy, which is limited by the size of the sliding window and cannot obtain the effective information from the far frame of the video. Fig. 7 shows that our network can also generate reasonable content to fill the missing areas, although the results of our method are a little bit oversmoothing. A possible reason is that we do not conduct the adversarial training like STTN and DSTT.



Figure 5: The decaptioning results of subtitles with outlines. Each group includes subtitled frame, predicted subtitle mask, decaptioning result, and ground truth.

Method	MSE↓	PSNR↑	SSIM↑	VFID↓
STTN	0.00084	35.1334	0.9704	0.0914
DSTT	0.00075	35.5684	0.9715	0.0923
Ours	0.00098	35.0430	0.9682	0.1246

Table 8: Quantitative comparisons of our method with STTN [9] and DSTT [10] on YouTube-VOS dataset. ↑: higher is better; ↓: lower is better.

In Fig. 8, we evaluate the decaptioning performance of Onion-Peel, DSTT, STTN, BVD-Net, and our method on the higher resolution videos with higher quality. The resolution of videos is 432×240 . Due to the lack of high-resolution training dataset, we directly use the trained models on the low-resolution public decaptioning dataset (128×128). From Fig. 8, we can see that our method obtains relatively better visual quality. Since patch size is fixed in the frame attention layer and somehow is suitable for low-resolution video, the test results of our method on the high-resolution video are slight smooth. In the future, we will construct a large-scale high-resolution decaptioning dataset to facilitate the related research, and also adapt our method to solve the high-resolution video decaptioning.



Figure 6: The decapting results of more motion under the subtitles. Each group includes subtitled frame, predicted subtitle mask, decapting result, and ground truth.

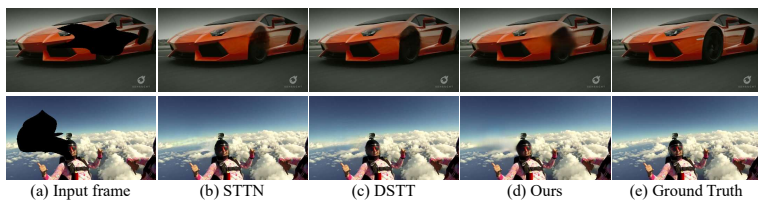


Figure 7: Qualitative comparisons of our method with STTN and DSTT on video inpainting. Black mask indicates the missing regions.

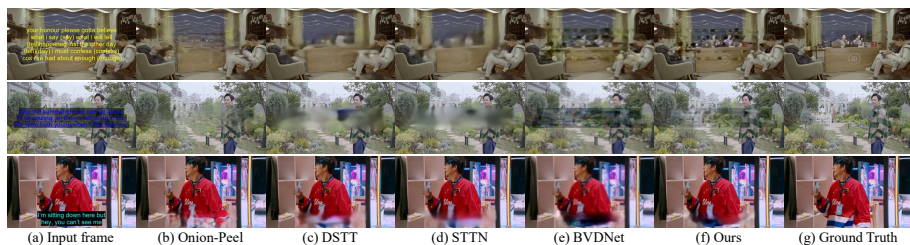


Figure 8: Decapting results on higher resolution video.

References

- [1] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep blind video decaptioning by temporal aggregation and recurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4263–4272, 2019.
- [2] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint arXiv:2104.06637*, 2021.
- [3] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *Proceedings of the European Conference on Computer Vision*, pages 528–543. Springer, 2020.