Supplementary Materials: Timeline and Boundary Guided Diffusion Network for Video Shadow Detection

Anonymous Authors

This is a text supplementary material for "Timeline and Boundary Guided Diffusion Network for Video Shadow Detection". The outline in this text material is:

- Sec 1 More Details of Our Approach.
- Sec 2 More Implement Details.
- Sec 3 More Experiments.

1 MORE DETAILS OF OUR APPROACH

1.1 Workflow of our pipeline

Here, we give more details about the workflow of our method. To begin with, our TBGDiff follows the previous video shadow detection paradigm [2, 4, 5] in which we detect all the shadow masks by the given frame sequence. That is to say, our workflow is based on clipped video level and each batch will contain a video clip. Empirically, we have 5 frames for each sequence as reported in our manuscript.

We encode all the frames by encoder E such that the DSA can aggregate all the timeline features. Then the aggregated features are used to produce pseudo masks and boundary masks. For a specific frame, it could be individually and parallelly decoded by the Diffusion. For each frame, we use the timeline (past and feature) guidance to guide the Diffusion model.

In terms of the Diffusion model, we detail it with training and inference stage. For training process, we use bit analog [1] to serialize the discrete masks and embed it into the latent features which is controlled by a scale weight. Then we use Guidance Encoder (GE) to yield space-time guidance by inputting the pseudo masks and timeline frames. Last, the decoder predicts the mask of current frame with the guidance. With respect to the sampling stage, the logic is similar to the training stage. To accelerate the inference, we use DDIM [7] strategy to sample. We display the pseudo codes of our TBGDiff including the training (**Algorithm 1**) and inference (**Algorithm 2**) stages and only one frame is shown for simplicity. We can implement the process parallelly for faster speed.

1.2 Details of the DSA

We illustrate the short-term and long-term frames in Fig. 1. We use the interval frames as the long-range frames to conduct residual affinity and use adjacent frames to apply vanilla affinity. Dual scale temporal frames are in consideration. For the first and last one, we copy itself as the adjacent frame. See the computation of the affinity in our manuscript.

1.3 Details of the Guidance Injection

The frames of the guidance from past and future are various. Hence, we concatenate the timeline guidance (g_f and g_p) and current feature, and send them to a residual block to fuse the space-time guidance. Fig. 2 shows the details.

Algorithm 1 : Training

```
def train_one_frame(frames, gts):
    # frames: flattened frames [b*t,3,h,w]
    # gts: flattened labels [b*t,3,h,w]
    fea_all = encoder(frames)
    ## dual scale aggregation, DSA
    fea_all = DSA(fea_all)
    boundary_masks, pseudo_masks = aux_head(fea_all)
    ## shadow boundary-aware attention, SBAA
    fea_all = SBAA(fea_all, boundary_masks)
    # diffusion part
    t, eps = uniform(0, 1), normal(mean=0, std=1)
    ## encode gt via bit analog strategy
    gt_enc = bit_analog(gts[now_idx])
    gt_enc = (sigmoid(gt_enc) * 2 - 1) * weight_scale
    gt_crpt = sqrt(alpha_cumprod(t)) * gt_enc
            +sqrt(1 - alpha_cumprod(t)) * eps
    ## guidance
    pseudo_mask = decoder(fea_all, t)
    past_pairs = concat(frames[past_idx],
                        pseudo_masks[past_idx])
    future_pairs = concat(frames[future_idx],
                          pseudo_mask[future_idx])
    ## produce guidance via Space-Time Embedding
    g_past = GE(past_pairs)
    g_future = GE(future_pairs)
    fea = inject(fea_all[now_idx], g_future, g_past)
    ## fuse corrupted gt and fea, zip the channel
    crpt_fea = _zip(concat(gts_crpt, fea))
    pred = decoder(crpt_fea, t) # [b,1,h,w]
    return loss_func(pred, gts[now_idx])
```

Algorithm 2 : Sampling

```
def sample_one_frame(frames, steps, td=1):
    # frames: flattened frames [b*t,3,h,w]
    # steps: sampling steps
    # td: time difference
    fea_all = encoder(frames)
    ## dual scale aggregation, DSA
    fea_all = DSA(fea_all)
    boundary_masks, pseudo_masks = aux_head(fea_all)
    ## shadow boundary-aware attention, SBAA
    fea_all = SBAA(fea_all, boundary_masks)
    ## pred of t step, initialize with gaussian noise
    pred_t = normal(0, 1) # [b,1,h,w]
    # diffusion part
    for step in range(steps):
        t_now = 1 - step / steps
        t_next = max(1 - (step + 1 + td) / steps, 0)
        ## guidance
        past_pairs = concat(frames[past_idx],
                            pseudo_mask[past_idx])
        future_pairs = concat(frames[future_idx],
                              pseudo_mask[future_idx])
        g_past = GE(past_pairs)
        g_future = GE(future_pairs)
        fea = inject(fea_all[now_idx], g_future, g_past)
        ## fuse corrupted gt and fea, zip the channel
        crpt_fea = _zip(concat(pred_t, fea))
        pred = decoder(crpt_fea, t_now)
        pred_t = ddim(pred_t, pred, t_now, t_next)
    return pred
```

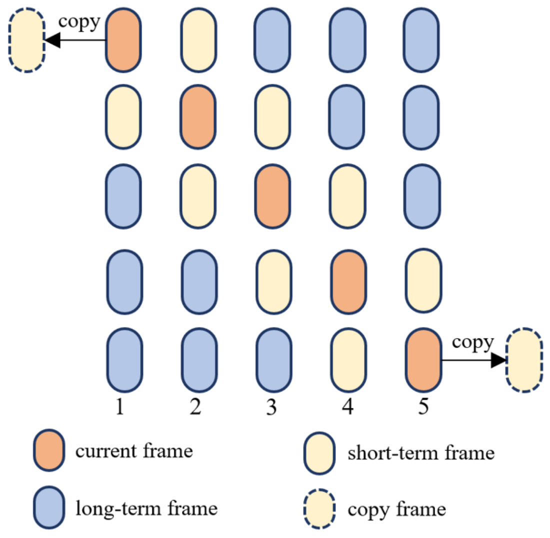

Figure 1: Definition of the long-term and short-term frames.

Table 1: Ablation studies on inputting different frames.

frames	IoU ↑	BER ↓	F_β ↑	MAE ↓
4	0.611	12.33	0.768	0.030
5	**0.667**	**8.58**	**0.797**	**0.023**
6	0.660	9.32	0.772	0.026
7	0.649	9.57	0.784	0.024

2 MORE IMPLEMENT DETAILS

Here, we list more setup details about our experiments. We take use of mixed-precision strategy to accelerate the training and testing, and the total of the training epoch is set as 20. To ensure the reproduction, we utilize a fixed random seed 42 to conduct all the experiments. Following [5], basic data augmentation techniques are adopted, *e.g.*, random flip horizontally and vertically. The code is supported by Pytorch.

As indicated in the manuscript, we adopt hierarchical transformer backbone MiT-B3 [8] to obtain the multi-scale features. For DSA, we only conduct the top-level features (*i.e.*, $\mathbb{R}^{512\times\frac{H}{32}\times\frac{W}{32}}$) for aggregating the temporal semantics, since applying it for all of the levels is redundant and extremely time-consuming due to matrix multiplication [3]. For the Guidance Encoder (GE), we only utilize a light-weight backbone MiT-B1 [8] to encode the timeline frames and pseudo masks jointly.

3 MORE EXPERIMENTS

3.1 Input Different Number of Frames

We add the ablation studies for the number of input frames, where the results are shown in Tab. 1. Apparently, when inputting 5 frames our model can produce the best results.

3.2 More Visual Comparisons

We provide more visual comparisons with *state-of-the-art* methods. Fig. 3 presents the qualitative results on one clipped video. It can be observed that our method can better localize the shadow areas. For

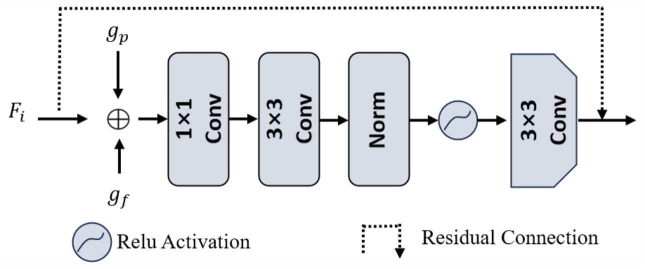

Figure 2: Guidance Injection. A residual block is used to fuse the feature and guidance. In the final 3×3 convolutional layer, we zip the channels.

the video comparison demos, please refer to our ***Supplementary Video*** to check the comparisons. We present 3 complete cases to compare the video shadow detection result, and each video contains 100 frames in 10 FPS speed.

REFERENCES

[1] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. 2023. A generalist framework for panoptic segmentation of images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 909–919.
[2] Zhihao Chen, Liang Wan, Lei Zhu, Jia Shen, Huazhu Fu, Wennan Liu, and Jing Qin. 2021. Triple-cooperative video shadow detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2715–2724.
[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. 2021. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 11781–11794.
[4] Xinpeng Ding, Jingwen Yang, Xiaowei Hu, and Xiaomeng Li. 2022. Learning shadow correspondence for video shadow detection. In *European Conference on Computer Vision*. Springer, 705–722.
[5] Lihao Liu, Jean Prost, Lei Zhu, Nicolas Papadakis, Pietro Liò, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. 2023. SCOTCH and SODA: A Transformer Video Shadow Detection Framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 10449–10458.
[6] Xiao Lu, Yihong Cao, Sheng Liu, Chengjiang Long, Zipei Chen, Xuanyu Zhou, Yimin Yang, and Chunxia Xiao. 2022. Video shadow detection via spatio-temporal interpolation consistency training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3116–3125.
[7] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.
[8] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems* 34 (2021), 12077–12090.

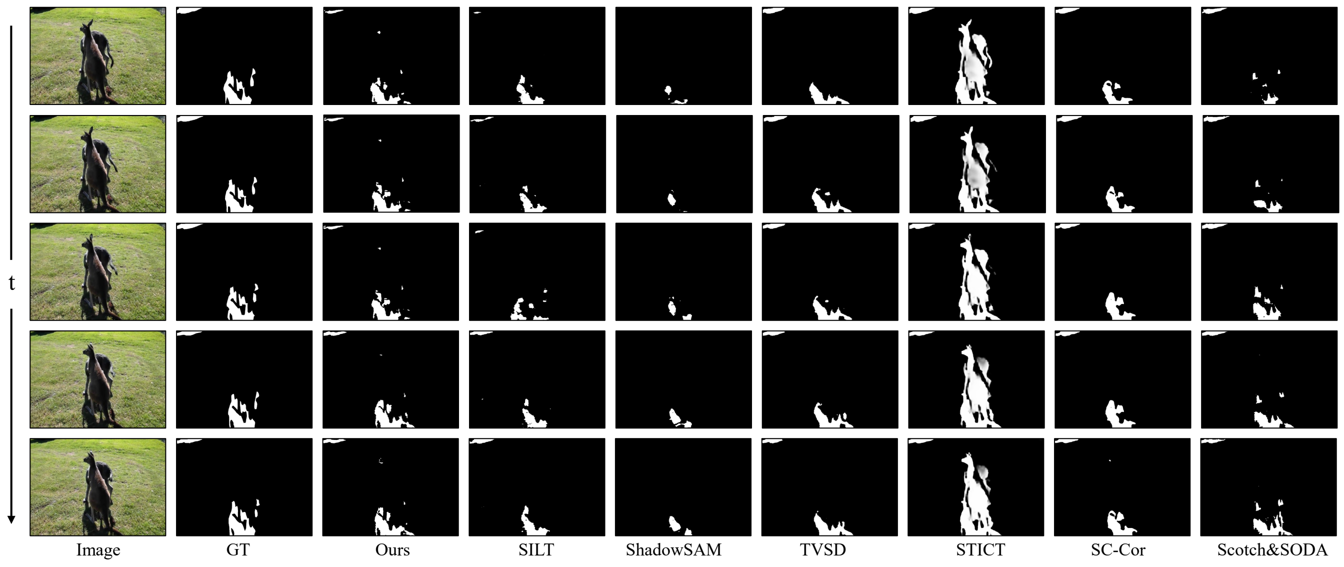

Figure 3: Visual comparisons on one clipped video.