Adaptive Human Matting for Dynamic Videos

Chung-Ching Lin, Jiang Wang, Kun Luo, Kevin Lin, Linjie Li, Lijuan Wang, Zicheng Liu
Microsoft
{chungching.lin,jiangwang,kun.luo,keli,linjli,lijuanw,zliu}@microsoft.com

Figure 1. Our method predicts reliable alpha mattes on dynamic videos without requiring trimaps or pre-captured backgrounds.

Abstract

*The most recent efforts in video matting have focused on eliminating trimap dependency since trimap annotations are expensive and trimap-based methods are less adaptable for real-time applications. Despite the latest tripmap-free methods showing promising results, their performance often degrades when dealing with highly diverse and unstructured videos. We address this limitation by introducing **Ada**ptive **M**atting for Dynamic Videos, termed **AdaM**, which is a framework designed for simultaneously differentiating foregrounds from backgrounds and capturing alpha matte details of human subjects in the foreground. Two interconnected network designs are employed to achieve this goal: (1) an encoder-decoder network that produces alpha mattes and intermediate masks which are used to guide the transformer in adaptively decoding foregrounds and backgrounds, and (2) a transformer network in which long- and short-term attention combine to retain spatial and temporal contexts, facilitating the decoding of foreground details. We benchmark and study our methods on recently introduced datasets, showing that our model notably improves matting realism and temporal coherence in complex real-world videos and achieves new best-in-class generalizability. Further details and examples are available at https://github.com/microsoft/AdaM.*

1. Introduction

Video human matting aims to estimate a precise alpha matte to extract the human foreground from each frame of an input video. In comparison with image matting [6, 10, 14, 21, 30, 39, 44, 47], video matting [2, 5, 11, 18, 19, 33, 36, 38] presents additional challenges, such as preserving spatial and temporal coherence.

Many different solutions have been put forward for the video matting problem. A straightforward approach is to build on top of image matting models [44], which is to implement an image matting approach frame by frame. It may, however, result in inconsistencies in alpha matte predictions across frames, which will inevitably lead to flickering artifacts [42]. On the other hand, top performers leverage dense trimaps to predict alpha mattes, which is expensive and difficult to generalize across large video datasets. To alleviate the substantial trimap limitation, OTVM [35] proposed a one-trimap solution recently. BGM [22, 33] proposes a trimap-free solution, which needs to take an additional background picture without the subject at the time of capture. While the setup is less time-consuming than creating trimaps, it may not work well if used in a dynamic background environment. The manual prior required by these methods limits their use in some real-time applications, such as video conferencing. Lately, more general solutions, e.g., MODNet [16] and RVM [23], have been proposed which involve manual-free matting without auxiliary inputs. However, in challenging real-world videos, backgrounds are inherently non-differentiable at some points, causing these solutions to produce blurry alpha mattes.

It is quite challenging to bring together the benefits of both worlds, i.e., a manual-free model that produces accurate alpha mattes in realistic videos. In our observation, the

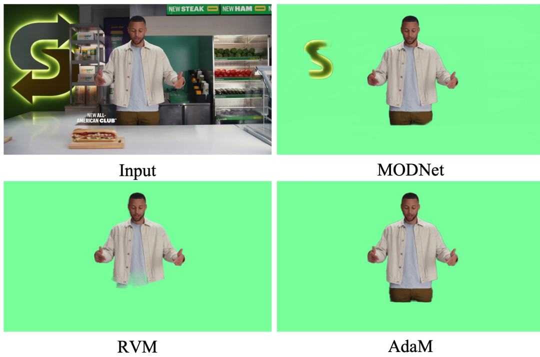


Figure 2. Qualitative sample results of MODNet [16], RVM [23] and the proposed AdaM on real video scenes.

significant challenges can mostly be explained by the inherent unconstrained and diverse nature of real-world videos. As a camera moves in unstructured and complex scenes, foreground colors can resemble passing objects in the background, which makes it hard to separate foreground subjects from cluttered backgrounds. This might result in blurred foreground boundaries or revealing backgrounds behind the subjects, as shown in Fig. 2. MODNet [16] and RVM [23] are both nicely designed models with auxiliary-free architectures to implicitly model backgrounds. In complex scenes, however, models without guidance may experience foreground-background confusion, thereby degrading the alpha matte accuracy.

In this paper, we aim to expand the applicability of the matting architecture such that it can serve as a reliable framework for human matting in real-world videos. Our method does not require manual efforts (e.g., manually annotated trimaps or pre-captured backgrounds). The main idea is straightforward: understanding and structuring background appearance can make the underlying matting network easier to render high-fidelity foreground alpha mattes in dynamic video scenes. Toward this goal, an interconnected two-network framework is employed: (1) an encoder-decoder network with skip connections produces alpha mattes and intermediate masks that guide the transformer network in adaptively enhancing foregrounds and backgrounds, and (2) a transformer network with long- and short-term attention that retain both spatial and temporal contexts, enabling foreground details to be decoded. Based on a minimal-auxiliary strategy, the transformer network obtains an initial mask from an off-the-shelf segmenter for coarse foreground/background (Fg/Bg) guidance, but the decoder network predicts subsequent masks automatically in a data-driven manner. The proposed method seeks to produce accurate alpha mattes in challenging real-world environments while eliminating the sensitivities associated with handling an ill-initialized mask. Compared to the recently published successes in video matting study, our main contributions are as follows:

- We propose a framework for human matting with unified handling of complex unconstrained videos without requiring manual efforts. The proposed method provides a data-driven estimation of the foreground masks to guide the network to distinguish foregrounds and backgrounds adaptively.
- Our network architecture and training scheme have been carefully designed to take advantage of both long- and short-range spatial and motion cues. It reaches top-tier performance on the VM [23] and CRGNN [42] benchmarks.

2. Related Work

The recent video matting studies can be summarized into two main categories: 1) trimap-based matting [35, 40, 42]; 2) trimap-free matting [16, 20, 22, 23, 33].

Trimap-based matting. Previous solutions mainly focus on estimating alpha mattes from image feature maps with auxiliary trimap supervisions, and extending the propagation temporally. The trimap serves to reduce the difficulty of the matting problem. Several methods [40, 42], have recently been proposed, which require multiple human-annotated trimaps for network guidance. Yet, the manual labor involved in creating well-defined trimap annotations is very costly. Recently, Seong et al. [35] propose one-trimap video matting network. The method requires only a single trimap input and performs trimap propagation and alpha prediction as a joint task. The model, however, is still limited by its requirement for a trimap in real-time applications. In addition, with an ill-initialized trimap, errors can be propagated over time.

Trimap-free matting. In order to relax the strong requirement of trimaps, studies from recent years have explored the idea of using background data [22, 33], annotated or predicted segmentation masks [11, 18, 41] as the guidance for matting networks, or designing fully automated auxiliary-free [23, 38, 38] methods.

Background-based matting. As an alternative to trimap input, background matting methods [22, 33] require users to take an additional picture of the background without the subject. This step is much less time-consuming than creating a trimap. However, the methods are constrained to videos with relatively static backgrounds. It is also less feasible to use the methods in real-time because background inputs must be captured beforehand.

Mask-guided matting. Some works in image matting use predicted or annotated segmentation masks to provide a rough guide to the foreground region [47], thereby relaxing the model dependence on the trimap. However, the direct application of these image matting methods to individual video frames neglects the temporal information in videos. The frame-by-frame mask generation could also largely restrict its use in real-time settings.

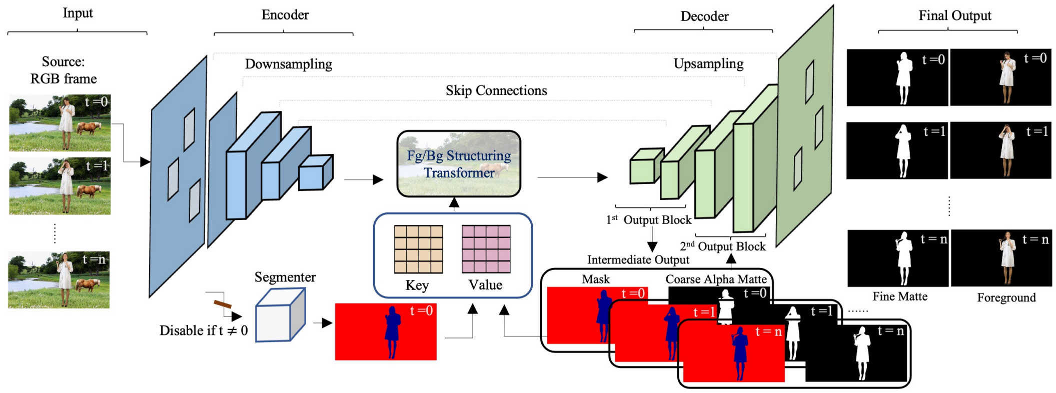


Figure 3. Overview of AdaM. Our framework includes an off-the-shelf segmenter [12], an encoder-decoder network, and a Fg/Bg structuring transformer network. Given an RGB frame, the encoder (Sec. 3.1) extracts low- and high-resolution features. At frame $t = 0$ only, the segmenter obtains an initial coarse mask for embedding into the initial Key and Value feature map. The transformer network (Sec. 3.2) employs long- and short-term attention to explicitly model Fg/Bg appearance. The decoder (Sec. 3.3) concatenates the Fg/Bg features encoded in the transformer and the high-resolution fine features from the encoder to produce an intermediate Fg/Bg mask and a final foreground alpha matte. The newly predicted Fg/Bg mask functions as a new reference mask used to update Value adaptively.

Auxiliary-free matting. Recently, some methods [16, 20, 23] explore auxiliary-free frameworks for video matting. For example, MODNet [16] considers frames as independent images and proposes a technique for suppressing flicker by making comparisons between adjacent frames. Concurrently, RVM [23] proposes a recurrent architecture to exploit temporal information in videos and achieves significant improvements in temporal coherence and matting quality. However, due to the lack of guidance, these methods can be largely compromised in challenging scenes where the cluttered background is present or the color distribution of the background is similar to that of the foreground.

Unlike previous auxiliary-free methods, our model employs a minimal-auxiliary approach. Except for the initial mask generated by an off-the-shelf segmenter, our network predicts the subsequent masks on its own and uses them to guide the network to render reliable alpha mattes in real-world videos.

3. Method

An overview of AdaM architecture is illustrated in Figure 3. Our framework includes an off-the-shelf segmenter [12], a fully-convolutional encoder-decoder network, and a Fg/Bg structuring transformer network. The pipeline begins with an RGB video frame being input into the encoder and the segmenter. The encoder (Section 3.1) takes the RGB input and extracts low-resolution coarse features as well as high-resolution fine features. The low-resolution coarse features are fed into the transformer network for efficient Fg/Bg modeling, while the high-resolution fine features are supplied to the decoder module for fine-grained alpha matte decoding. Given the first video frame, the segmenter estimates an initial coarse mask. It is noted that the segmenter only serves as an adjunct for obtaining the introductory mask and it could be replaced with a more sophisticated segmentation network [3, 4, 15]. The initial mask and the low-resolution features are embedded into the initial Key and Value and stored in the transformer. Each of the subsequent input frames acts as a query. The transformer network (Section 3.2) employs long- and short-term attention to explicitly model Fg/Bg appearance. Specifically, a per-pixel comparison is made between the coarse low-resolution features and the Key/Value to estimate the features of the foreground or background. Short-term attention is applied to capture local-to-local spatial interdependency and motion continuity, while long-term attention helps to model local-to-global context reasoning. With the Fg/Bg structuring network, more meaningful semantic information is encoded and a cleaner foreground region is supplied to the decoder, making it easier for the decoder to recover finer details in the foreground. Following that, the Fg/Bg features encoded in the transformer and the high-resolution fine features from the encoder are concatenated together in the decoder (Section 3.3), producing an intermediate Fg/Bg mask and a final foreground alpha matte. The newly predicted Fg/Bg mask functions as a new reference mask used to update Value adaptively.

3.1. Encoder

We adopt MobileNetV2 [32] as our backbone. In a given video, frames are sequentially processed. For each video frame, the encoder network extracts features at 1/2, 1/4, 1/8, and 1/16 scales with different levels of abstraction ($F^{1/2}$, $F^{1/4}$, $F^{1/8}$, $F^{1/16}$). $F^{1/16}$ is fed into the transformer to model the foreground and background more efficiently. The rest of the features are fed into the decoder through skip connections to provide more fine-grained details for alpha matte predictions.

3.2. Fg/Bg Structuring Network

Our method is designed to explicitly model Fg/Bg appearance. The core design is inspired by recent successes of memory mechanisms applied to natural language processing [17, 27, 37] and vision models [26, 29, 34, 46]. As the transformer architecture consists of Query, Key and Value in the multi-head attention, it is suitable to dynamically determine which input regions we want to attend more than others. We thus adopt the transformer architecture and extend the concept of memory networks in [29, 46] to make it appropriate to model and propagate spatio-temporal Fg/Bg information for the video matting task.

In our pipeline, the Fg/Bg information of previous frames is stored in Key and Value, and a current frame to be processed acts as Query. At each frame, the transformer network takes the low-resolution feature map from the encoder and the previously estimated mask from the decoder as inputs. Through visual semantics and the mask information, the network learns to predict whether a given feature belongs in the foreground or the background. Key provides invariant visual semantic information and serves as an addressing function. The feature map of the current frame is used as a Query. Comparing the similarity of Query and Key, the network retrieves relevant Fg/Bg features in Value, and propagates them over space and time through long- and short-term attention. The Fg/Bg features of the current frame are then fed to the decoder for fine-grained alpha estimation.

We first review and define the essential elements of the transformer. An attention operation is the core component of a transformer layer, which linearly projects input features Z into queries Q, keys K, and values V:

$$Q = ZW_Q, \quad K = ZW_K, \quad V = ZW_V, \tag{1}$$

where W_Q, W_K, W_V are learnable linear projection layers. To obtain an output feature Z', a self-attention operation can be expressed as:

$$Z' = \mathrm{Attn}(Q, K, V) = \mathrm{Softmax}(QK^\top/\sqrt{d})V. \tag{2}$$

We extend the basic attention operation to include the previously computed features stored in Key and Value. This is to encourage the network to learn spatial interdependence and motion continuity embedded in videos. In specific, after processing a video frame, the computed features are retained for subsequent iterations. When processing the frame I^t at time t ($t > 0$), our model refers to Keys and Values computed from earlier iterations to learn long-term and short-term contextual dependencies. The attention operation of our Fg/Bg structuring transformer Ξ can be represented as:

$$Z'^t = \Xi(Q^t, \hat{\boldsymbol{K}}^t, \hat{\boldsymbol{V}}^t), \tag{3}$$

$$\hat{\boldsymbol{K}}^t = \mathit{concat}(K^{t-n}, \ldots, K^{t-1}), \tag{4}$$

$$\hat{\boldsymbol{V}}^t = \mathit{concat}(V^{t-n}, \ldots, V^{t-1}), \tag{5}$$

where $concat()$ denotes concatenation along the feature dimension, K^{t-1} and V^{t-1} are Key and Value at time $t-1$, respectively. In this formulation, the query Q^t attends to $\hat{\boldsymbol{K}}^t$ and $\hat{\boldsymbol{V}}^t$ that contain previously computed features up to n steps in the past.

Fg/Bg update scheme. As outlined in the method overview, given the previous frame I^{t-1}, our decoder outputs an intermediate mask, m^{t-1}, which is used to embed estimated Fg/Bg features of I^{t-1} into Value. The Fg/Bg embedded feature map, $V^{t-1}_{f/b}$, can be expressed as:

$$V^{t-1}_{f/b} = V^{t-1} + (m_d^{t-1}E_f + (1 - m_d^{t-1})E_b), \tag{6}$$

where m_d^{t-1} is the downsized Fg/Bg mask of m^{t-1}, and E_f and E_b are learnable embeddings for foreground f and background b, respectively. To process frame I^t, we concatenate Value feature map $\hat{\boldsymbol{V}}^t_{f/b}$ to include $V^{t-1}_{f/b}$:

$$\hat{\boldsymbol{V}}^t_{f/b} = \mathit{concat}(V^{t-n}_{f/b}, \ldots, V^{t-1}_{f/b}), \tag{7}$$

As both $\hat{\boldsymbol{K}}^t$ and $\hat{\boldsymbol{V}}^t_{f/b}$ contain Fg/Bg information accumulated from prior iterations, the network performs attention operations along the temporal domain for each input query.

Long-term local-to-global attention. In highly dynamic scenes, the foreground and background variations over a sequence of frames could be non-local. In light of this, we employ long-term local-to-global attention to capture non-local variations. A conventional transformer operation is used to perform non-local attention on $\hat{\boldsymbol{K}}^t$ and $\hat{\boldsymbol{V}}^t_{f/b}$:

$$Z'^t_L = \mathrm{Attn}(Q^t, \hat{\boldsymbol{K}}^t, \hat{\boldsymbol{V}}^t_{f/b}), \tag{8}$$

In this formulation, every pixel in the query feature map can attend to all the stored Key and Value features. The network can thus capture long-term temporal variations. As non-local propagation involves high computational complexity and many features are redundant, this operation is only performed every l frames.

Short-term local-to-local attention. Generally, scene changes are smooth and continuous across frames. In the context of video matting, capturing fine-grained local temporal variations would be beneficial for ensuring temporal consistency. Thus, in addition to long-term local-to-global attention, our network also performs short-term local-to-local attention:

$$Z'^t_S = \text{S-Attn}_{\omega,s}(Q^t, \hat{\boldsymbol{K}}^t, \hat{\boldsymbol{V}}^t_{f/b}). \tag{9}$$

In this short-term attention module, $\text{S-Attn}_{\omega,s}$ restricts the attention to a spatial (ω) and temporal (s) tube around the pixel position with a $\omega \times \omega \times s$ tube size.

Finally, the outputs of long-term global attention and short-term local attention module are combined to obtain the final output of Ξ: $Z'^t = Z'^t_L + Z'^t_S$.

3.3. Decoder

We use FPN [24] as our decoder, which contains four upscaling blocks and two output blocks. The decoder takes the output of the transformer and multi-scale features at different stages to estimate the alpha mattes. At each upscaling block, the module concatenates the upsampled output from the previous block and a feature map from the encoder at the corresponding scale through skip-connections. The upscaling process repeats at 1/16, 1/8, 1/4 and 1/2 scale.

We employ a two-stage refinement to progressively refine alpha mattes. The model first produces a Fg/Bg mask and a coarse alpha matte at 1/4 scale of the original resolution, and then predicts a finer alpha matte at full resolution. The Fg/Bg mask prediction is used to update Value. The coarse alpha matte prediction and the feature map at 1/4 scale of the original resolution are concatenated and fed to the subsequent upscaling block for further alpha matte refinement. The second output block produces the final alpha matte at the original resolution.

3.4. Losses

Our model predicts a Fg/Bg mask m_p, a coarse alpha matte α_p^c, and a fine-grained alpha matte α_p^f. We represent the ground truth alpha matte α_{GT}. To learn Fg/Bg masks, we use the ground truth alpha matte α_{GT} to generate pseudo labels m_{GT}^* and compute binary cross entropy loss as:

$$\mathcal{L}^m(m_p) = m_{GT}^*(-log(m_p)) + (1 - m_{GT}^*)(-log(1 - m_p)), \tag{10}$$

where $m_{GT}^* = \alpha_{GT} < \tau$ and τ is a threshold.

To learn alpha matte predictions, we compute the L1 loss $\mathcal{L}_{l1}^{\alpha}$ and Laplacian pyramid loss ($\mathcal{L}_{lap}^{\alpha}$) [10, 14] as:

$$\mathcal{L}_{l1}^{\alpha}(\alpha_p) = ||\alpha_p - \alpha_{GT}||_1 \tag{11}$$

$$\mathcal{L}_{lap}^{\alpha}(\alpha_p) = \sum_{s=1}^{5} \frac{2^{s-1}}{5} ||L_{pyr}^s(\alpha_p) - L_{pyr}^s(\alpha_{GT})||_1 \tag{12}$$

The overall loss function $\mathcal{L}$ is:

$$\begin{aligned} \mathcal{L} = \omega_m \mathcal{L}^m(m_p) &+ \omega_\alpha^c(\mathcal{L}_{l1}^{\alpha}(\alpha_p^c) + \mathcal{L}_{lap}^{\alpha}(\alpha_p^c)) \\ &+ \omega_\alpha^f(\mathcal{L}_{l1}^{\alpha}(\alpha_p^f) + \mathcal{L}_{lap}^{\alpha}(\alpha_p^f)), \end{aligned} \tag{13}$$

where ω_m, ω_α^c, and ω_α^f are loss weights. For our experiments, we empirically determined these weights to be $\omega_m = 0.5$, $\omega_\alpha^c = 0.5$, and $\omega_\alpha^f = 1$.

4. Experiments

We evaluate our framework on VideoMatte240K [22] and CRGNN [42] benchmark datasets, and compared it to leading trimiap-free video matting baselines, including FBA [10], BGMv2 [22] MODNet [16] and RVM [23]. We then study in greater depth the properties of AdaM by an ablation study. All design decisions and hyperparameter tuning were performed on the validation set.

4.1. Datasets and Metrics

The VideoMatte240K (VM) dataset [22] consists of 484 videos. RVM [23] split VM dataset into 475/4/5 video clips for training, validating and testing. The training set is converted into SD and HD sets for different training stages. In evaluation, the testing set is converted and split into VM 512×288 and VM 1920×1080 sets. To ensure a fair comparison, we use the same training, validation, and test sets created by RVM. [23]. VM is the largest public video matting dataset containing continuous frame sequences for evaluating motion coherence. However, due to the high cost of collecting video matting annotations, most existing datasets, including VM, only provide ground-truth alpha and foreground, which must be composited with background images. As far as we are aware, there are only limited annotated real video datasets that can be used to evaluate video matting. To study the generalization capability of the proposed method, we evaluate our method on the real video dataset provided by CRGNN. [42]. The CRGNN real dataset (CRGNN-R) consists of 19 real-world videos. Annotations are made every 10 frames at a 30 fps frame rate, which adds up to 711 labeled frames.

In accordance with previous methods, the video matting accuracy is evaluated by the Mean of Absolute Difference (MAD), Mean Squared Error (MSE), spatial Gradient (Grad) [31] and Connectivity (Conn) [31] errors. Conn metric is not analyzed at high resolution because it is too expansive to compute. We also evaluate the temporal consistency of alpha matte predictions using dtSSD [8].

4.2. Implementation Details

The video matting dataset we use to train our network is VideoMatte240K [22] (VM). Since this is a composited dataset, synthesized videos are usually not realistic and lack the context of real-world footage. Similar to the reasons explained in RVM [23], we train our network on both matting and semantic segmentation datasets (PASCAL [9], COCO [25], YouTubeVOS [45]) to reduce the gap between synthetic and real domains. Since the high-definition image matting dataset (Adobe Image Matting [44], AIM) is not available for licensing reasons, we do not train our model using AIM and Distinctions-646 [30]. It is expected that our model could be further enhanced when including HD image matting data in the training pipeline.

We divide the training process into three stages. First, we initialize the MobileNetV2 [32] encoder with the weights pre-trained on ImageNet [7] and train the whole model on image segmentation datasets (PASCAL [9], COCO [25]). We apply multiple random motion augmentations [23] on static image segmentation datasets to synthesize video sequences. The model is trained for 100K iterations using AdamW with a learning rate of 1e-4 and weight decay of 0.03. Second, we train our model alternatively on low-

Method	Features		VM 512×288					VM 1920×1080			
	Backbone	HD Training	MAD ↓	MSE ↓	Grad ↓	Conn ↓	dtSSD ↓	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
FBA [10]	ResNet50	✓	8.36	3.37	2.09	0.75	2.09	-	-	-	-
BGMv2 [22]	MobileNetV2	✓	25.19	19.63	2.28	3.26	2.74	-	-	-	-
MODNet [16]	MobileNetV2	✓	9.41	4.30	1.89	0.81	2.23	11.13	5.54	15.3	3.08
RVM [23]	MobileNetV3	✓	6.08	1.47	0.88	0.41	1.36	6.57	1.93	10.55	1.90
RVM-Large [23]	ResNet50	✓	-	-	-	-	-	5.81	0.97	9.65	1.78
AdaM	MobileNetV2	-	**5.55**	**0.84**	**0.80**	**0.33**	1.38	**4.61**	**0.46**	**6.06**	**1.47**
AdaM	MobileNetV2	✓	**5.30**	**0.78**	**0.72**	**0.30**	**1.33**	**4.42**	**0.39**	**5.12**	**1.39**

Table 1. Comparison with state-of-the-art methods on the test split of VM [22] 512×288 and VM 1920×1080 datasets.

Method	HD Traning	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
MODNet [16]	✓	9.50	4.33	16.94	6.22
RVM [23]	✓	15.42	9.22	18.34	6.95
AdaM	-	**7.13**	**3.05**	17.96	**5.76**
AdaM	✓	**5.94**	**2.79**	**16.71**	**5.45**

Table 2. Evaluation of the transfer capability of MODNet, RVM and AdaM on the whole CRGNN Real dataset.

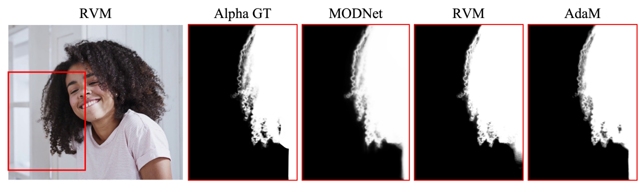


Figure 4. Comparison of alpha matte details.

resolution VM data (odd stages) and video segmentation data (even stages). The model is trained for another 100K iterations using AdamW with a learning rate of 2e-4 and weight decay of 0.07. Third, we train our model only on VM HD data to learn fine-grained alpha matte predictions. The model is trained for 20K iterations using AdamW with a learning rate of 1e-5 and weight decay of 0.07. The linear warmup and cosine learning rate scheduler is used at all three training stages.

The initial segmentation mask ($t = 0$) is estimated by an off-the-shelf Mask R-CNN detector [12, 43] with ResNet50 [13] backbone. The transformer in AdaM consists of three layers with a hidden size of 256D. The step l in long-term attention is 10. To reduce computational complexity, the network stores up to 10 sets of Key and Value features for long-term attention. The window size, ω and s, for the short-term attention are 7 and 1, respectively. All experiments are performed on the Nvidia Tesla V100 GPUs. The mini-batch size is 2 clips per GPU.

4.3. Results on the VM Benchmark

The experimental results on VM dataset are summarized in Table 1. Without training on high-definition data, AdaM outperforms previous state-of-the-art methods in most metrics by a considerable margin in both VM sets. In addition, AdaM with MobileNetV2 backbone outperforms RVM-Large with ResNet-50 in VM 1920×1080 set. We attribute the notable improvement in performance to our integrated framework, which is designed to simultaneously differentiate foreground from background and capture alpha matte details in dynamic environments. The last set of results demonstrates that performance could be further enhanced with training on high-definition video data.

4.4. Results on the CRGNN Benchmark

To evaluate the robustness and generalization of our method trained on the composited dataset to a real-video dataset, we compare the matting quality by transferring MODNet [16], RVM [23], and AdaM to the real dataset CRGNN-R. We do not fine-tune the pre-trained models on CRGNN dataset. Table 2 shows AdaM is competitive among these leading methods, suggesting the proposed method can overcome the domain shift problem and generalize to real-world data more effectively. We believe the proposed framework is a core factor for its general success.

4.5. Qualitative Analysis

As MODNet, RVM and the proposed method do not require manual inputs (e.g., trimaps or pre-captured backgrounds), we are able to conduct further tests on real-world videos. In Figure 4, we compare the alpha matte predictions. Our method is able to predict fine-grained details more accurately. Figure 7 presents qualitative comparisons on real-world videos from YouTube.[1] Each sub-figure illustrates a challenging case. Despite being top-performing methods, MODNet and RVM produce blurry foreground boundaries in videos with dynamic backgrounds or reveal some backgrounds behind the subjects, indicating that real-world video matting presents a fundamental challenge for existing techniques. Our visualization depicts comparably reliable results in highly challenging scenes (e.g., fast camera movement in Figure 7 (a), fast human motion in in Figure 7 (b), and cluttered background in Figure 7 (e)).

4.6. Ablation Study

For a controlled evaluation, the ablations are performed using the same training recipe on the VM 1920×1080 set.

[1] The examples are drawn from test experiments on videos retrieved from YouTube for research purposes to evaluate AdaM's robustness in complex scenarios.

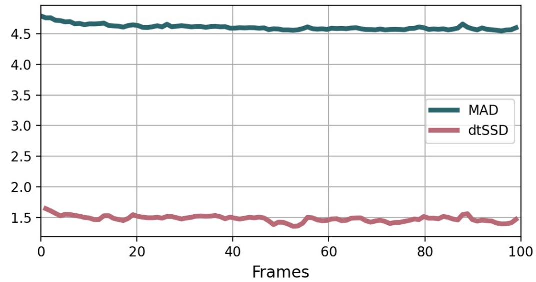


Figure 5. MAD and dtSSD performance of AdaM across frames.

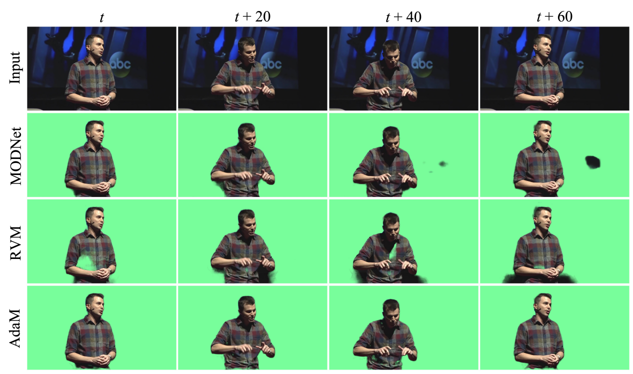


Figure 6. Evaluation of temporal coherence. Video scenes: A Ted talk speaker talks with very fast pose gesture movements.

Performance over temporal dimension. Figure 5 shows the average alpha MAD and dtSSD of AdaM along time steps in VM test set. Overall, we observe the error (MAD) and temporal consistency (dtSSD) measures improve in the first 10 frames and then stay stable over time. Further, we examine the temporal coherence of AdaM. In Figure 6, a Ted talk speaker gives a speech with rapid gesture movements. In comparison with MODNet [16] and RVM [23], our model produces less undesirable artifacts and more consistent alpha matte predictions.

Performance in static and dynamic background. Table 3 compares our model performance on static and dynamic backgrounds in VM. In dynamic videos, both foreground subjects and background move, while in static videos, only foreground subjects move. In static videos, BGMv2 produces the best results as it uses ground-truth background as an additional input. However, its performance drops significantly in dynamic videos. Overall, our model performs robustly on both static and dynamic backgrounds and presents little performance difference in two sets.

BG	Method	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
Static	BGMv2 [22]	4.33	0.32	4.19	1.33
	MODNet [16]	11.04	5.42	15.80	3.10
	RVM [23]	5.64	1.07	9.80	1.84
	AdaM	4.56	0.43	5.91	1.41
Dynamic	BGMv2 [22]	42.45	37.05	17.30	4.61
	MODNet [16]	11.23	5.65	14.79	3.06
	RVM [23]	7.50	2.80	11.30	1.96
	AdaM	4.65	0.48	6.20	1.51

Table 3. Performance in static and dynamic backgrounds.

Shrot- and long-term attention. To evaluate the effect of short- and long-term attention in transformer, we train several models, turning them on and off. Table 4 shows that the models with single attention produce inferior results. Comparing these two models, the model with short-term attention performs better in dtSSD. As short-term local attention propagates contextual information among neighboring pixels in nearby frames, it helps enhance temporal consistency in alpha matte predictions. In contrast, long-term global attention addresses recurrence and dependencies of Fg/Bg, contributing to MAD and MSE. With both short- and long-range attention, spatial interdependency and motion consistency can be captured.

Short-	Long-	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
✓		5.32	0.61	6.76	1.55
	✓	4.97	0.53	6.70	1.63
✓	✓	4.61	0.46	6.06	1.47

Table 4. Short-term and long-term attention.

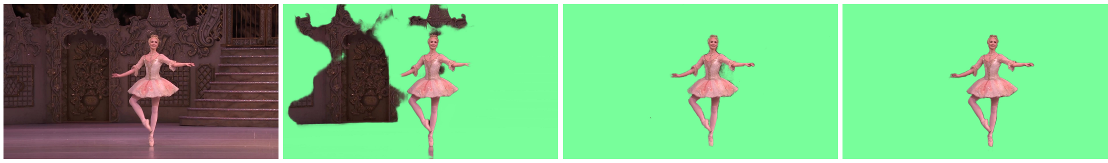
Fg/Bg update mechanism. We study the effect of the Fg/Bg update scheme described in Sec. 3.2. In this experiment, we test two variants. First, we disable the update mechanism. This is equivalent to removing Equation 6. Second, we directly convert the final alpha matte prediction into a Fg/Bg mask and use it to update Value in transformer. The network without an update does not directly absorb Fg/Bg information. Although the previous Key and Value still hold information stored from prior time-steps, the network processes Fg/Bg information in an implicit way. As Table 5 shows, the performance decreases without the update mechanism. When the network is updated using alpha matte prediction, the performance also declines. As an alpha matte is represented by a float value in the range [0,1], in a direct conversion, an incorrect alpha matte prediction will be more likely to result in Fg/Bg mask errors, which will be propagated across frames. The mask update scheme, on the other hand, is intended to provide effective control over the integrated network for utilizing meaningful semantics to a greater extent, thus yielding better performance.

If update	Update by	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
-	-	6.15	1.53	10.55	2.13
✓	Alpha	5.65	0.89	8.27	1.68
✓	Mask	4.61	0.46	6.06	1.47

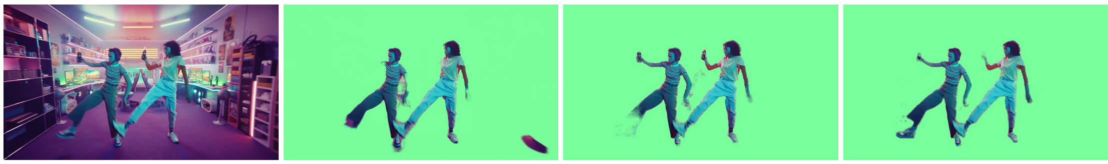
Table 5. Effect of Fg/Bg update mechanism.

Performance of different segmenters and backbones. We use Mark R-CNN [12] to obtain an introductory mask for initiating Value in transformer. We replace Mask R-CNN with a recent method, Transfiner [15]. Table 6 shows the results are similar, suggesting that the segmenter does not have a direct influence. Our method generates subsequent masks by itself and performs self-updates to mitigate the sensitivities associated with handling an ill-initialized

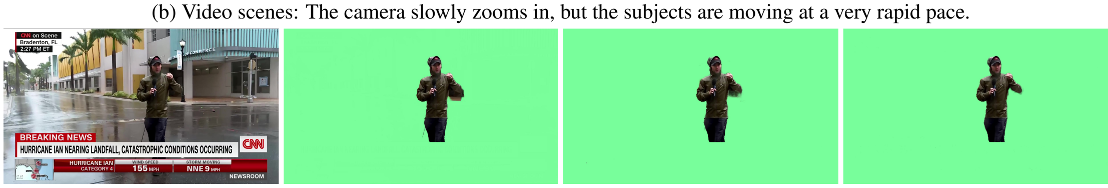
Input MODNet RVM AdaM



(a) Video scenes: The ballet dancer moves at a moderate tempo, but the camera moves quickly.

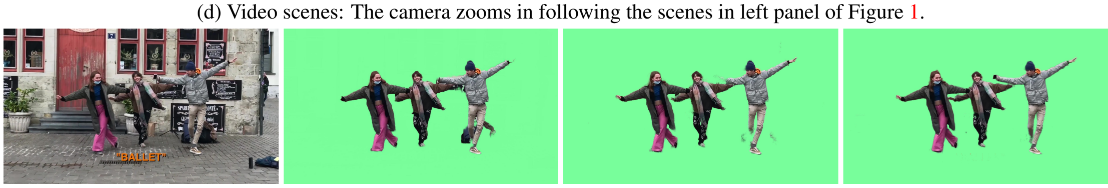


(b) Video scenes: The camera slowly zooms in, but the subjects are moving at a very rapid pace.



(c) Video scenes: A reporter is standing in a hurricane area. A strong wind blows his clothes.

(d) Video scenes: The camera zooms in following the scenes in left panel of Figure 1.



(e) Video scenes: Street scenes are captured by a camera following the subjects' movements.

Figure 7. Comparisons of our model to existing methods on complex real-world videos.

mask. The results support our hypothesis that the impact of the segmenter is modest. Further, we replace MobileNet [32] backbone with ResNet50 [13]. As expected, the model with ResNet50 is able to learn rich feature representations and further enhances performance. However, it comes at a cost of higher computational complexity.

	Model	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
Segmenter	Mask R-CNN	4.61	0.46	6.06	1.47
	Transfiner	4.58	0.45	6.01	1.47
Backbone	MobileNetV2	4.61	0.46	6.06	1.47
	ResNet50	4.53	0.42	4.99	1.34

Table 6. Performance of variants with different segmenters and backbones.

4.7. Speed

The purpose of our method is to provide an alternative for reliable video matting in complex real-world scenarios. We manage to enhance the accuracy over previous attempts and still maintain real-time capability. For example, the inference time of our model for a HD video on an Nvidia GTX 2080Ti GPU is 42 fps. Comparatively, our method does not achieve the same speed as MODNet [16] and RVM [23], but it is capable of attaining high accuracy in challenging real-world videos. A few simple ways of enhancing model throughput are to perform mixed precision inferences or to integrate TensorRT [28] with an ONNX [1] backend. While beyond the scope of this study, we expect many of such speed improvement techniques to be applicable to our work.

5. Conclusion

We have introduced AdaM for reliable human matting in diverse and unstructured videos, where extensive foreground and background movements appear and affect the way by which many heretofore existing methods perform. The competitiveness of our method on the composited and real datasets suggests that explicit modeling of Fg/Bg appearance to help guide the matting task can be a vital contributing factor to video matting success. Comprehensive ablation studies indicate several key factors and advantages of the proposed model, including its design, temporal coherence, and generalizability to a wide range of real videos.

References

[1] Junjie Bai, Fang Lu, Ke Zhang, et al. Onnx: Open neural network exchange. https://github.com/onnx/onnx, 2019. 8

[2] Xue Bai, Jue Wang, and David Simons. Towards temporally-coherent video matting. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pages 63–74. Springer, 2011. 1

[3] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 3

[4] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 3

[5] Inchang Choi, Minhaeng Lee, and Yu-Wing Tai. Video matting using multi-frame nonlocal matting laplacian. In *European Conference on Computer Vision*, pages 540–553. Springer, 2012. 1

[6] Yutong Dai, Brian Price, He Zhang, and Chunhua Shen. Boosting robustness of image matting with context assembling and strong data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11707–11716, 2022. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[8] Mikhail Erofeev, Yury Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, pages 99–1, 2015. 5

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 5

[10] Marco Forte and François Pitié. f, b, alpha matting. *arXiv preprint arXiv:2003.07711*, 2020. 1, 5, 6

[11] Minglun Gong, Liang Wang, Ruigang Yang, and Yee-Hong Yang. Real-time video matting using multichannel poisson equations. In *Proceedings of Graphics Interface 2010*, pages 89–96. 2010. 1, 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 6, 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 8

[14] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4130–4139, 2019. 1, 5

[15] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4412–4421, 2022. 3, 7

[16] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1140–1147, 2022. 1, 2, 3, 5, 6, 7, 8

[17] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016. 4

[18] Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. Temporally coherent video matting. *Graphical Models*, 72(3):25–33, 2010. 1, 2

[19] Dingzeyu Li, Qifeng Chen, and Chi-Keung Tang. Motion-aware knn laplacian for video matting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3599–3606, 2013. 1

[20] Jiachen Li, Vidit Goel, Marianna Ohanyan, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. Vmformer: End-to-end video matting with transformer. *arXiv preprint arXiv:2208.12801*, 2022. 2, 3

[21] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11450–11457, 2020. 1

[22] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8762–8771, 2021. 1, 2, 5, 6, 7

[23] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022. 1, 2, 3, 5, 6, 7, 8

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[26] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *European Conference on Computer Vision*, pages 661–679. Springer, 2020. 4

[27] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016. 4

[28] NVIDIA. Tensorrt. https://github.com/NVIDIA/TensorRT, 2019. 8

[29] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 4

[30] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13676–13685, 2020. 1, 5

[31] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1826–1833. IEEE, 2009. 5

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3, 5, 8

[33] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Computer Vision and Pattern Regognition (CVPR)*, 2020. 1, 2

[34] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *European Conference on Computer Vision*, pages 629–645. Springer, 2020. 4

[35] Hongje Seong, Seoung Wug Oh, Brian Price, Euntai Kim, and Joon-Young Lee. One-trimap video matting. *arXiv preprint arXiv:2207.13353*, 2022. 1, 2

[36] Ehsan Shahrian, Brian Price, Scott Cohen, and Deepu Rajan. Temporally coherent and spatially accurate video matting. In *Computer Graphics Forum*, volume 33, pages 381–390. Wiley Online Library, 2014. 1

[37] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015. 4

[38] Jiayu Sun, Zhanghan Ke, Lihe Zhang, Huchuan Lu, and Rynson WH Lau. Modnet-v: Improving portrait video matting via background restoration. *arXiv preprint arXiv:2109.11818*, 2021. 1, 2

[39] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11120–11129, 2021. 1

[40] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. Deep video matting via spatio-temporal alignment and aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6975–6984, 2021. 2

[41] Zhen Tang, Zhenjiang Miao, and Yanli Wan. Temporally consistent video matting based on bilayer segmentation. In *2010 IEEE International Conference on Multimedia and Expo*, pages 370–375. IEEE, 2010. 2

[42] Tiantian Wang, Sifei Liu, Yapeng Tian, Kai Li, and Ming-Hsuan Yang. Video matting via consistency-regularized graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4902–4911, 2021. 1, 2, 5

[43] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[44] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2970–2979, 2017. 1, 5

[45] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 5

[46] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021. 4

[47] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. Mask guided matting via progressive refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1154–1163, 2021. 1, 2