

## VMFormer: End-to-End Video Matting with Transformer

Jiachen Li<sup>1,3\*</sup>, Vidit Goel<sup>3</sup>, Marianna Ohanyan<sup>3</sup>, Shant Navasardyan<sup>3</sup>, Yunchao Wei<sup>2</sup>, Humphrey Shi<sup>1,3</sup>

<sup>1</sup>SHI Labs @ Georgia Tech & Oregon & UIUC, <sup>2</sup>BJTU, <sup>3</sup>Picsart AI Research (PAIR)

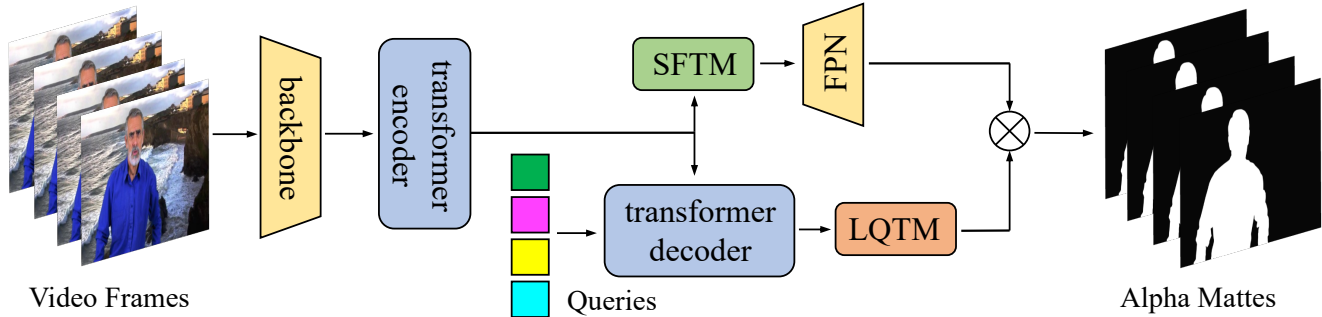


Figure 1. **An overview of the VMFormer.** VMFormer has two branches for feature modeling and query modeling. The feature modeling branch takes video clips as input and extracts feature sequences through the backbone and the transformer encoder. The query modeling branch learns the global representation of feature sequences through the transformer decoder. Temporal modeling is applied to each branch individually to improve predictions with temporal context. Finally, VMFormer makes predictions of alpha mattes with queries and feature maps. The detailed architecture of each branch is illustrated in Figure 2.

### Abstract

Video matting aims to predict the alpha mattes for each frame from a given input video sequence. Recent solutions to video matting have been dominated by deep convolutional neural networks (CNN) for the past few years, which have become the de-facto standard for academia and industry. However, they have the inbuilt inductive bias of locality and do not capture the global characteristics of an image due to the CNN-based architectures. They also need long-range temporal modeling considering computational costs when dealing with feature maps of multiple frames. In this paper, we propose **VMFormer**: a transformer-based end-to-end method for video matting. It makes predictions on alpha mattes of each frame from learnable queries given a video input sequence. Specifically, it leverages self-attention layers to build global integration of feature sequences with short-range temporal modeling on successive frames. We further apply queries to learn global representations through cross-attention in the transformer decoder with long-range temporal modeling upon all queries. In the prediction stage, both queries and corresponding feature maps are used to make the fi-

nal prediction of alpha mattes. Experiments show that VMFormer outperforms previous CNN-based video matting methods on synthetic benchmarks with different input resolutions, as an end-to-end video matting solution built upon a full vision transformer with predictions on the learnable queries. The project is open-sourced at <https://chrisjuniorli.github.io/project/VMFormer>.

### 1. Introduction

Video matting aims to predict the alpha mattes of each frame from a given input video sequence. It has received considerable attention from industry and academia in recent years since it can be widely used in applications like video conferencing and video editing. Given a video sequence  $\mathbf{I} = \{I_1, I_2, \dots, I_T\}$ , where each frame  $I_i$  is assumed as a composition of a foreground image  $F_i$  and a background image  $B_i$  [41] with a coefficient alpha matte map  $\alpha_i \in [0, 1]$ :

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad (1)$$

where  $\alpha_i \in [0, 1]$  is the alpha matte. To predict precisely  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_T\}$ , which is an under-constrained problem since for each pixel, there are seven unknown values

\*Work is partially done during an internship at Picsart AI Research

(three from  $F_i$ , three from  $B_i$ , and one from  $\alpha_i$ ) while only three known equations from  $I_i$  on  $R, G, B$  channels separately. To estimate  $\alpha$ , some solutions expect an auxiliary input called *trimap*, which is a segmentation ground truth of foreground, background, and unknown regions of frames. Such *trimap-based* methods [23, 38, 42, 45, 47, 50, 51] take inputs from both RGB frames and user-provided trimaps to predict alpha mattes based on deep convolutional neural networks (CNN). Since trimap annotations are expensive for videos, some *trimap-free* solutions [19, 24, 25, 36] are proposed for predicting alpha mattes using only RGB frames as inputs.

However, most state-of-the-art CNN-based solutions suffer from the intrinsic inductive bias of CNN, especially on the locality of the receptive field and translation equivariance, which makes it hard to capture the global characteristics of the image while performing the task. Some methods [31, 38] further adopt attention mechanisms to alleviate this issue while the decoder part remains CNN-based. Moreover, in these models, temporal modeling is usually performed on feature maps of successive frames [25] or used as a post-processing approach to improving global temporal consistency [19]. These temporal modeling techniques lack long-range modeling and would be computationally expensive if implemented upon the feature maps of the whole video sequence. Recently, some vision transformer solutions [11, 29] adopt self-attention model interaction between non-local pixels and get a global receptive field. Regarding downstream tasks like detection and segmentation, some methods [9, 40] leverage multiple queries and cross-attention to recognize all instances in images, which does not apply to video understanding since predictions are independent for each frame.

Motivated by these observations, we propose **VMFormer**: an end-to-end video matting solution based on a transformer as shown in Figure 1. It contains two branches: feature modeling and query modeling. The feature modeling branch has a CNN-based backbone and a transformer encoder for feature extraction of input video sequences. The self-attention layers in each transformer encoder block involve global modeling across the whole picture of each frame. The query modeling branch leverage the transformer decoder with cross-attention and queries for predictions of alpha mattes. Each query corresponds to an input frame. The cross-attention layer in each transformer decoder block integrates the learnable queries with corresponding feature sequences globally. In terms of temporal modeling, We add a short-range feature-based temporal modeling (SFTM) module based on recurrent aggregations of successive feature maps. To add long-range temporal modeling in VMFormer while avoiding the heavy computations of applying it to feature sequences, we adopt a long-range query-based temporal model (LQTM) that learns the weights for

each query and updates all queries with the learned weights. In the end, the queries directly make matrix multiplications with the feature maps of the corresponding frame to predict alpha mattes accordingly.

We train and evaluate VMFormer with synthetic training and testing data from VideoMatte240K [24], BG20K [22], and DVM [38]. We also evaluate VMFormer on the testing set used in RVM to compare with other state-of-the-art CNN-based models without re-training the model. Extensive experiments and ablation studies show that VMFormer performs better than previous CNN-based networks while keeping the real-time inference speed.

To summarize, our contributions are as follows:

- We propose VMFormer: an end-to-end trimap-free video matting method based on a vision transformer and make predictions with learnable queries.
- We further build an efficient long-range query-based temporal modeling (LQTM) module and short-range feature-based temporal modeling (SFTM) module to improve the performance of VMFormer on video matting.
- To our best knowledge, VMFormer is the first end-to-end video matting solution built upon a transformer. We hope it motivates the community to explore more solutions on this track.

## 2. Related Works

### 2.1. Image Matting

Image matting is to estimate the accurate foreground in the image, which is an important computer vision task [41]. Mathematically, given an Image  $I$  as a combination of unknown foreground image  $F$  and background image  $B$ , which has

$$I = \alpha F + (1 - \alpha)B \quad (2)$$

Previous image matting solutions mainly focus on low-level features to distinguish the transition areas between foregrounds and backgrounds [1, 2, 6, 10, 12, 14]. In recent years, deep learning based methods use neural networks to estimate alpha mattes from image feature maps with auxiliary trimap supervisions in an end-to-end manner [23, 31, 33, 42, 45, 47, 51]. Other methods also explore trimap-free solutions with only segmentation network [5] or coarse annotations [28, 48].

### 2.2. Video Matting

When extending image matting to video sequences, trimap-based solutions focus on building temporal modeling with attention-based spatial-temporal feature aggregation [38, 50]. For trimap-free solutions, background matting series [24, 36] use an auxiliary input of the back-

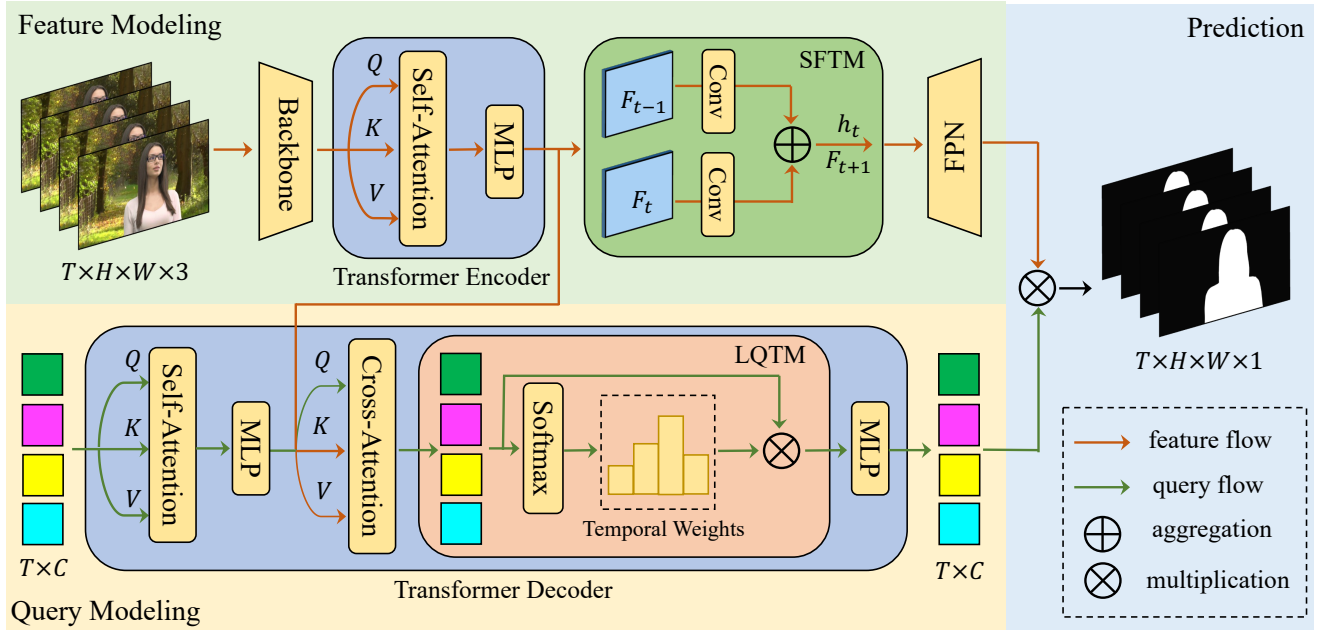


Figure 2. **The Architecture of the VMFormer.** It contains two separate branches for modeling features and queries: a) The feature modeling path contains a CNN-based backbone network to extract feature pyramids and a transformer encoder that integrates feature sequences globally with short-range feature-based temporal modeling (SFTM) on feature maps of successive frames. b) The query modeling path has a transformer decoder for queries to learn global representations of feature sequences. Long-range query-based temporal modeling (LQTM) is built upon all queries with learnable temporal weights. The final alpha matte predictions are based on matrix multiplication between queries and feature maps. LayerNorm, residual connection, and repeated blocks are omitted for simplicity.

ground image for the first frame and estimate alpha mattes of the whole video sequences. MODNet [19] employs a self-supervised method to model temporal consistency. RVM [25] leverages segmentation data to make the matting model robust to real-world scenes and VideoMatt [21] explores different CNN-based temporal modeling methods. Although previous works explore the utilization of attention mechanisms in video matting solutions, the mainstream is still CNN-based models. Meanwhile, to our best knowledge, cross-attention has never been used so the predictions are still based on CNN kernels with limited perception.

### 2.3. Vision Transformer

Recently, vision transformers are popular in the computer vision community. ViT [11] and CCT [16] use a vanilla transformer encoder for image classification. Swin [29] and NAT [15] further employ hierarchical local vision transformers as the backbone. DETR [3] adopts a full transformer for object detection and Deformable DETR [52] proposes multi-scale deformable attention to reducing the training efforts with faster convergence. For pixel-prediction tasks like segmentation, Max-Deeplab [40], MaskFormer [8, 9] and SeMask [18] use learnable queries for masks predictions through cross-attention in image segmentation tasks. The queries are

used for predicting instance-level segmentation results. In video instance segmentation, vision transformer based solutions [7, 43, 44] further leverage queries for predictions of instance masks and build temporal matching across frames. However, there are not many explorations on utilizing a transformer to solve image-level video segmentation tasks like video matting, since predicting whole images across multiple frames brings a huge burden on memory, let alone the computationally expensive operators like self-attention and cross-attention. Furthermore, matching queries with feature sequences remains a challenging task for video understanding.

## 3. VMFormer

In this section, we introduce how we establish the whole VMFormer architecture from feature modeling and query modeling to the prediction stage, as shown in Figure 2.

### 3.1. Feature Modeling

**Backbone** The backbone network takes video sequence  $\mathbf{I} = \{I^1, I^2, \dots, I^T\} \in \mathbb{R}^{T \times H \times W \times 3}$  as input where  $T$  is the number of frames and extracts a pyramid of feature maps  $\mathbf{F} = \{F_1, F_2, F_3, F_4\}$  with dimension  $F_i \in \mathbb{R}^{T \times \frac{H}{2^i} \times \frac{W}{2^i} \times C_i}$ . Then, they are projected to

same dimension  $C$  and flattened into feature sequences  $\mathbf{X} \in \mathbb{R}^{T \times L \times C}$  where  $L = \sum_{i=1}^4 \frac{HW}{4^i}$  as inputs for transformer.

**Transformer Encoder** The transformer encoder is built upon  $N_{enc}$  consecutive blocks. Each encoder block consists of a self-attention layer (SA) and a multi-layer perceptron (MLP). LayerNorm (LN) and residual connection are applied to the output of them. Given input feature sequences  $\mathbf{X}$ , it goes through each encoder block and the overall feature sequences are the concatenation of all frames as  $\mathbf{X}_{enc} = \{X_{enc}^1, X_{enc}^2, \dots, X_{enc}^T\} \in \mathbb{R}^{T \times L \times C}$ .

**SFTM** We reshape the feature sequence  $\mathbf{X}_{enc}$  into feature pyramids  $\mathbf{F}_{enc} = \{F_{enc_1}, F_{enc_2}, F_{enc_3}, F_{enc_4}\}$  for predictions. To involve temporal modeling between feature maps of consecutive frames, we first use a simple feature aggregation that aggregates feature maps of two consecutive frames  $F_{enc_i}^t$  and  $F_{enc_i}^{t-1}$  with a convolution layer, which brings performance improvement as shown in the ablation study 4.3 of short-range feature-based temporal modeling (SFTM). To explicitly make the temporal context transferable across frames, we propose a recurrent aggregation, which updates the feature maps at each frame with the recurrent features  $h_{enc_i}^t$  iteratively on each frame. The  $h_{enc_i}^t$  is initialized with  $F_{enc_i}^0$  and updated the same way as  $F_{enc_i}^{t+1}$ . The experiments show that the model benefits from recurrent aggregation more than direct feature aggregation, and we use recurrent aggregation as the SFTM module in the VMFormer. Then, we use an FPN on top of these updated feature pyramids.

### 3.2. Query Modeling

**Transformer Decoder** For the query modeling part, the main goal is to use the cross-attention module to make learnable queries interact with whole feature sequences so that they can learn global representations and make better predictions for alpha mattes. To make training and inference flexible, we first generate a sequence of queries  $\mathbf{Q} \in \mathbb{R}^{T \times C}$  as inputs to  $N_{dec}$  consecutive decoder blocks, which has the same length  $T$  to the feature sequences so that each query can interact with corresponding feature sequence  $X_{enc}^i$ . In each decoder block, each query  $Q^i$  firstly goes through the self-attention layer (SA) based module. Then, it interacts with the corresponding feature sequence  $X_{enc}^i$  globally in the cross-attention layer (CA) based module. In this module, the key  $K_{enc}^i$  and value  $V_{enc}^i$  are linear transformation from feature sequence  $X_{enc}^i$ . The dot-product between query  $Q^i$  and the key  $K_{enc}^i$  produces the normalized weights for the value  $V_{enc}^i$ , which outputs the updated query  $Q^i$ .

**LQTM** To add temporal modeling based on queries since

they have learned effective global representation from the cross-attention layer and are less computationally expensive than feature maps. We add an attention-based long-range temporal modeling (LQTM), and the attention weights are learned from the queries themselves. We first apply a self-attention-based approach, which uses a multi-head self-attention layer to compute the  $Q^{tem}$  with temporal attention weights of all queries, where the learnable queries show better time consistent with the temporal attention weights, as shown in ablation study 4.3 of long-range query-based temporal modeling (LQTM). Considering the simplicity of queries and self-attention may be overkill for queries, we further use additive attention, which adopts a normalized linear layer to learn the temporal weights. It is also computationally efficient, which only has a cost of  $\mathcal{O}(TC)$  that is much less than  $\mathcal{O}(THWC)$  if it is applied to video feature sequences. As a result, we select the additive attention for the LQTM module. The  $Q^{tem}$  is then added to  $Q^i$  with LayerNorm, residual connection, and MLP applied followingly as  $Q_{dec}^i$ . Both the outputs of the learnable queries and the feature sequences at each frame  $Q_{dec}^i \in \mathbb{R}^C$  and  $X_{enc}^i \in \mathbb{R}^{L \times C}$  (Since  $X_{enc}^i$  stays the same after cross-attention, so the notation remains unchanged) are concatenated as  $\mathbf{Q}_{dec} \in \mathbb{R}^{T \times C}$  and  $\mathbf{X}_{enc} \in \mathbb{R}^{T \times L \times C}$  for final predictions.

### 3.3. Prediction Stage

**Predictor** After getting queries  $\mathbf{Q}_{dec}$  from the transformer decoder, we make predictions based on the largest feature map and the queries  $\mathbf{Q}_{dec}$ , which are projected on the largest feature map with a batch matrix multiplication and the outputs are further upsampled as final predictions of alpha mattes  $\alpha \in \mathbb{R}^{T \times H \times W \times 1}$ .

**Loss Functions** To compute losses during the training stage, we use a combination loss function

$$L = \lambda L_{focal} + L_{dice} + L_{tmp} \quad (3)$$

which includes pixel-level focal loss [26] with balanced coefficient  $\lambda$ ,  $\alpha$  is a balanced term and  $p_{x,y}$  is the prediction on the pixel with coordinate  $(x, y)$ .

$$L_{focal} = -\alpha(1 - p_{x,y})^\gamma \log(p_{x,y}) \quad (4)$$

$L_{dice}$  is dice loss [30] and  $g_{x,y}$  is the ground truth on the pixel with coordinate  $(x, y)$ .  $L_{tmp}$  is a temporal consistency loss [25].

$$L_{dice} = 1 - \frac{2 \times \sum_{x,y} p_{x,y} g_{x,y}}{\sum_{x,y} p_{x,y}^2 + \sum_{x,y} g_{x,y}^2} \quad (5)$$

$$L_{tmp} = \left\| \frac{\partial p_{x,y}}{\partial t} - \frac{\partial \hat{g}_{x,y}}{\partial t} \right\|_2 \quad (6)$$



	MAD↓	MSE↓	Grad↓	Conn↓	FPS
$N_{enc} = 1$	7.02	2.01	1.38	0.56	109.5
$N_{enc} = 2$	<b>6.57</b>	<b>1.57</b>	<b>1.10</b>	<b>0.49</b>	97.5
$N_{enc} = 3$	6.51	1.55	1.12	0.48	84.1
$N_{dec} = 1$	7.02	2.01	1.38	0.56	109.5
$N_{dec} = 2$	6.90	1.93	1.36	0.54	90.2
$N_{dec} = 3$	6.87	1.92	1.35	0.53	78.6

Table 1. Ablation Study on Number of Encoder and Decoder Blocks. We select  $N_{enc} = 1$  and  $N_{dec} = 1$  for VMFormer considering the trade-off between inference speed and accuracy.

	MAD↓	MSE↓	Grad↓	Conn↓
$\lambda = 1$	7.02	2.01	1.38	0.56
$\lambda = 5$	<b>7.02</b>	1.81	1.25	<b>0.56</b>
$\lambda = 10$	7.17	<b>1.77</b>	<b>1.22</b>	0.57

Table 2. Ablation Study on the Coefficient  $\lambda$  of Focal Loss. We select  $\lambda = 5$  for the VMFormer.

	MAD↓	MSE↓	Grad↓	Conn↓
$F_1$	<b>7.02</b>	<b>2.01</b>	1.38	<b>0.56</b>
$F_1 + F_2$	7.03	2.04	1.38	0.56
$F_1 + F_2 + F_3$	7.02	2.01	<b>1.35</b>	0.56

Table 3. Ablation Study on the Auxiliary Supervision. We only select supervision from the final predictions  $F_1$ .

**Inference** During inference process, given a input video clip  $\mathbf{I} \in \mathbb{R}^{t \times H \times W \times 3}$  and  $t$  is the length of the video clip. It is split into consecutive sequences  $\mathbf{I}_i \in \mathbb{R}^{T \times H \times W \times 3}$  and  $T$  could be any length during inference. Then, each  $\mathbf{I}_i$  is sent to VMFormer from the backbone, transformer encoder, and transformer decoder to predictor sequentially. We only use the outputs  $\alpha_i \in \mathbb{R}^{T \times H \times W \times 1}$  from the last transformer decoder block as predictions. Finally, all outputs  $\alpha_i$  are combined to  $\alpha \in \mathbb{R}^{t \times H \times W \times 1}$  as alpha matte predictions of the corresponding input video clip  $\mathbf{I}$ .

## 4. Experiments

### 4.1. Dataset and Evaluation

**Dataset** Following previous works, we also use synthetic training data from the merged image foreground dataset ImageMatte [25], the video foreground dataset VideoMatte240K [24], the image background dataset BG20K [22] and the video background dataset from DVM [38]. Following RVM [25], we split ImageMatte and VideoMatte240K into 420/15/21 and 475/4/5 video clips for training, validating, and testing. Similarly, BG20K is split into 15000/500/4500 image sets, and DVM is split into 3080/37/162 image sets for training, validating, and testing. During training, we randomly select foreground from ImageMatte and VideoMatte240K, background from BG20K, and DVM to composite video clips on the fly. During test-

	MAD↓	MSE↓	Grad↓	Conn↓
12-epoch	7.02	2.01	1.38	0.56
24-epoch	6.40	1.44	0.96	0.47
36-epoch	<b>6.24</b>	<b>1.30</b>	<b>0.91</b>	<b>0.44</b>
48-epoch	6.29	1.32	0.92	0.46

Table 4. Ablation Study on the Training Epochs. We use the result of 36-epoch training as a strong baseline model.

	MAD↓	MSE↓	Grad↓	Conn↓
Baseline	6.24	1.30	0.91	0.44
<i>SFTM</i>				
+ feature aggregate	6.06	1.28	0.82	0.42
+ recurrent aggregate	5.14	0.71	0.53	0.29
<i>LQTM</i>				
+ self attention	5.67	0.79	0.62	0.39
+ additive attention	5.56	0.75	0.61	0.33
<i>Both</i>	<b>4.91</b>	<b>0.55</b>	<b>0.40</b>	<b>0.25</b>

Table 5. Ablation Study on the Short-range Feature-based Temporal Modeling (SFTM) and Long-range Query-based Temporal Modeling (LQTM) Modules. We use recurrent aggregate for SFTM and additive attention for LQTM in the final model of VMFormer.

ing, we composite 50 challenging clips of resolutions at  $512 \times 288$  and  $1920 \times 1080$  for testing the VMFormer and other models.

**Evaluation** We mainly consider video matting accuracy and temporal coherence for evaluation. For video matting accuracy, we use Mean Absolute Difference (MAD), Mean Squared Error (MSE), Gradient (Grad), and Connectivity (Conn) [34] as evaluation metrics. We also scale MAD, MSE, Grad, and Conn by  $10^3$ ,  $10^3$ ,  $10^{-3}$ , and  $10^{-3}$ , respectively, for convenience of reference. For all these metrics, the lower number represents better performance.

### 4.2. Implementation Details

**Training Settings** For training the VMFormer network, we use 8 RTX A6000 GPUs with batch size at 2 video clips per GPU. The optimizer is Adam, with different initial learning rates at different model modules. The backbone is under  $2 \times 10^{-5}$  and other modules are under  $2 \times 10^{-4}$  with weight decay at  $1 \times 10^{-4}$ . All models in ablation studies are trained for 12 epochs and the learning rates are decayed at the 6th and 8th epoch by a factor of 0.1. Given an input video clip, it is further split into sequences with length  $T = 5$  for training and inference due to the GPU memory constraints. Then, for augmentation, each frame of the input video sequences is randomly resized to the longest side in the range of [288, 320, 352, 392, 416, 448, 480, 512] with random horizontal flipping.

**Backbone** The backbone we used for VMFormer is Ima-

Model	MAD↓	MSE↓	Grad↓	Conn↓
<i>LR: 512 × 288</i>				
BGMv2† [24]	6.63	1.79	1.54	0.50
MODNet [37]	10.39	5.65	2.02	1.04
RVM [25]	5.99	1.17	1.10	0.34
VMFormer	<b>4.91</b>	<b>0.55</b>	<b>0.40</b>	<b>0.25</b>
<i>HD: 1920 × 1080</i>				
BGMv2† [24]	27.01	21.31	20.34	49.69
MODNet [37]	11.56	6.47	15.23	17.23
RVM [25]	6.45	1.63	11.59	5.74
VMFormer	<b>4.81</b>	<b>0.78</b>	<b>4.90</b>	<b>3.34</b>

Table 6. Comparisons on the Compositest Test Set. We tested the models with inputs at low-resolution  $512 \times 288$  and high-resolution  $1920 \times 1080$ . † indicates an augmented BGMv2 with background input in each frame.

geNet [20] pretrained Mobilenetv3-Large [17] since it is the most frequently used backbones in prior works. We use the implementation and weights from torchvision [32].

**Transformer Encoder** The transformer encoder and decoder are based on Deformable DETR [52]. We adopt multi-scale deformable self-attention to replace multi-head attention for faster convergence and efficient computing. We set the channel  $C = 256$ , the number of attention head  $M = 8$ , and the number of key sampling points  $K = 4$  for each multi-scale deformable attention module.

**Transformer Decoder** The number of queries is set to be 5 for training at the transformer decoder with channels  $C = 256$  to be compatible with the length  $T = 5$  of training samples. The positional embedding has the same dimensions as the queries. During feed-forward propagation, both the self-attention and cross-attention are implemented based on a multi-scale deformable attention module, in which the number of attention head  $M = 8$  and the number of key sampling points  $K = 4$ .

**Predictor** After getting learnable queries and the recovered feature maps from the feature pyramid network, we simply multiply queries to feature maps with a batch matrix multiplication implemented based on torch.einsum [35]. Then, the predicted alpha mattes are upsampled by bilinear interpolation to the original size of input sequences.

**Loss Functions** For focal loss, we set  $\lambda$  to 5 for final models based on ablation experiments on  $\lambda$ .  $\alpha$  and  $\gamma$  are set to 0.25 and 2, respectively.

**Inference** During inference, each video clip for testing has 100 frames, and we set  $T = 20$  for inference with input at

Model	# Params	MAD↓	MSE↓	Grad↓	Conn↓
<i>LR: 512 × 288</i>					
DeepLabV3 [4]	60.99 M	14.47	9.67	8.55	1.69
FBA [13]	34.69 M	8.36	3.37	2.09	0.75
BGMv2 [24]	4.99 M	25.19	19.63	2.28	3.26
MODNet [37]	6.49 M	9.41	4.30	1.89	0.81
RVM [25]	3.75 M	6.08	1.47	0.88	0.41
VMFormer	7.46 M	<b>6.02</b>	<b>1.00</b>	<b>0.75</b>	<b>0.37</b>
<i>HD: 1920 × 1080</i>					
BGMv2† [37]	4.99 M	20.35	14.26	22.79	-
MODNet [37]	6.49 M	11.13	5.54	15.30	-
RVM [25]	3.75 M	6.57	1.93	10.55	-
VMFormer	7.46 M	<b>6.20</b>	<b>1.53</b>	<b>6.30</b>	<b>5.38</b>

Table 7. Comparisons on the Compositest Test Set from RVM. Conn under  $1920 \times 1080$  were not reported in RVM’s paper due to expensive computation. †BGMv2 was tested by us with an augmented version since it was not reported in RVM. VMFormer shows state-of-the-art performance without re-training the model.

Model	Backbone	# Params	FPS
BGMv2 [24]	Mobilenetv3	4.99 M	195.9
MODNet [24]	Mobilenetv3	6.49 M	124.0
RVM [25]	Mobilenetv3	3.75 M	131.5
VMFormer	Mobilenetv3	7.46 M	109.5
w/o LQTM	Mobilenetv3	7.37 M	110.1
w/o SFTM	Mobilenetv3	6.99 M	114.4

Table 8. Speed Comparisons between different models. All models are tested on a single A6000 GPU without extra optimization on inference.

$512 \times 288$  and  $T = 5$  for inference with input at  $1920 \times 1080$  on a single RTX A6000 GPU with batch size at 1.

### 4.3. Ablation Studies

**Number of Encoder and Decoder Blocks** In Table 1, we change the number of transformer encoder and decoder blocks to evaluate if deeper architecture improves the performance. We set  $N_{enc} = 1, 2$  and  $N_{dec} = 1, 2$  due to the GPU memory constraints. It shows that the model benefits from deeper architecture like other CNN-based models.

**Coefficient of Focal Loss** Since we observe that the loss value of the focal loss is much smaller than dice loss and temporal consistency loss, we add a balanced coefficient  $\lambda$  for focal loss and test the performance under different values of  $\lambda$ . It shows that improving the coefficient of focal loss could improve temporal consistency as shown in Table 2 and we choose  $\lambda = 5$  for the VMFormer and following experiments.

**Training Epochs** In all ablation studies above, we use a 12-epochs training setting for fair comparisons. We extend the training scheduler to 24 and 36 epochs to evaluate the effect of a longer training scheduler, considering that recent vision transformer works [3, 52] require longer training schedulers

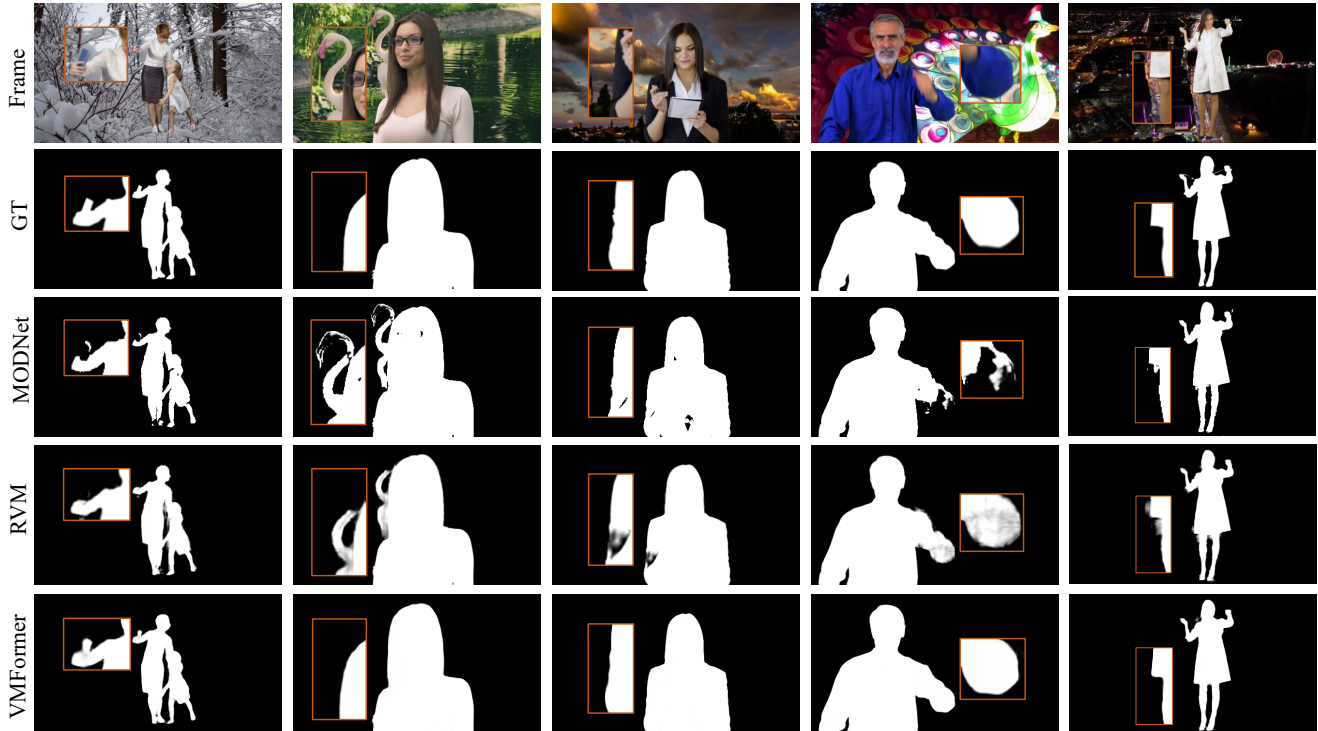


Figure 3. Visualization of alpha matte predictions from MODNet, RVM, and VMFormer under challenging frames from the synthetic test set. VMFormer shows a better ability to distinguish ambiguous foreground from background regions as shown in the magnified image patches. Please zoom in for details.

for better convergence. The learning rates decayed at the 12th/18th and 20th/30th epoch by 0.1, respectively. Experiments show that a longer training scheduler consistently improves the performance of VMFormer as shown in Table 4.

**Auxiliary Supervisions** Considering that queries could make predictions based on each level of the feature pyramids, we further compute losses on predictions of the lower resolution feature maps  $F_2$  and  $F_3$ . Experiment results are shown in Table 3 that extra supervisions from  $F_2$  and  $F_3$  are not improving performance. To summarize, both adding auxiliary supervisions from more transformer decoder blocks and predictions of intermediate levels are not helpful to VMFormer. This conclusion is not aligned with observations from other vision transformers [3] [44], we think it implies that prediction of a single-channel alpha matte is relatively easy for transformer compared with instance-level recognition tasks.

**Temporal Modeling** To evaluate the effectiveness of temporal modeling module SFTM and LQTM, we train VMFormer without any temporal modeling in  $3\times$  scheduler as a strong baseline for comparison. For SFTM, we evaluate both feature aggregation and recurrent aggregation as discussed in Section 3.1, and recurrent aggregation shows better performance on all metrics compared to feature aggregation.

We further evaluate self-attention and additive attention for LQTM as discussed in Section 3.2, and additive attention achieves better performance with lower computations. Ultimately, we adopt recurrent aggregation-based SFTM and additive attention-based LQTM for the VMFormer. Experiments show that the model benefits from both sides and achieves the best performance, as shown in Table 5.

#### 4.4. Comparisons to State-of-the-art Methods

To make fair comparisons to previous methods, we select the most recent state-of-the-art trimap-free video matting solutions, including BGMv2 [24], MODNet [37] and RVM [25] for comparisons, and evaluate all models on the same synthetic test set. Then, we evaluate the pre-trained VMFormer on the test set used in RVM [25].

**Synthetic Test Set** In Table 6, we compare VMFormer to three state-of-the-art CNN-based trimap-free video matting solutions BGMv2 [24], MODNet [37] and RVM [25]. We used the pre-trained MobileNetV3 [17] based checkpoints since they perform better than the reproduced ones that re-trained on our training sets. We evaluate all models on the synthetic test set that contains 50 challenging video clips, and each one has 100 frames for evaluation. They are tested under the resolution at  $512 \times 288$  and  $1920 \times 1080$  to show

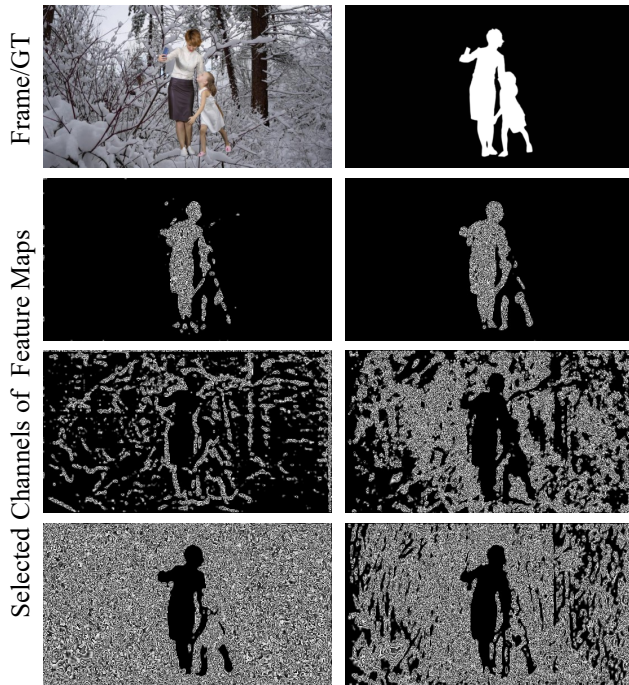


Figure 4. Visualization of the largest feature map ( $\frac{H}{2} \times \frac{W}{2}$ ) from VMFormer under a challenging input frame. We selected feature maps of 6 single channels for presentation. VMFormer can better capture global information and distinguish foreground/background in the whole image. Please zoom in for details.

the robustness of the models to different resolutions. It shows that VMFormer reaches the best performance on all metrics concerning to the accuracy of alpha matte predictions among all these models. Specifically, it reaches 4.91 MAD, 0.55 MSE, 0.40 Grad, and 0.25 Conn under both inputs of  $512 \times 288$ . For input of  $1920 \times 1080$ , it reaches 4.81 MAD, 0.78 MSE, 4.90 Grad, and 3.34 Conn, respectively.

**RVM’s Test Set** In Table 7, we further compare VMFormer to other methods on the synthetic test set used in RVM. All these methods except VMFormer are trained on both matting data and segmentation data [27, 39, 46]. We evaluate the models under input resolution both at  $512 \times 288$  and  $1920 \times 1080$  to show the robustness of the models to different resolutions. VMFormer outperforms other models without further joint training on the segmentation data under both inputs of low resolution and high resolution.

**Speed Comparison** Since video matting is close to industrial applications, thus real-time inference is always preferable for video matting algorithms. Previous CNN-based methods MODNet [37], BGMv2 [24] and RVM [25] have shown real-time inference speed in Table 8. All these methods are tested on a single A6000 GPU under the input of  $512 \times 288$  without extra optimization during inference. It

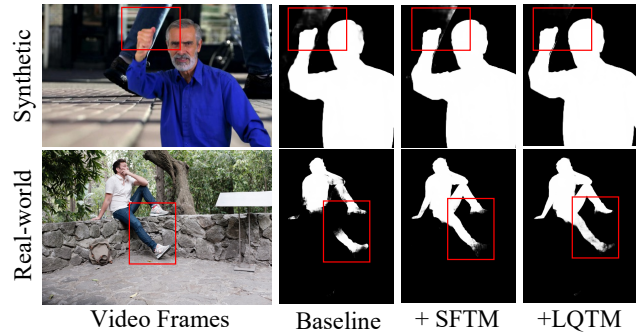


Figure 5. Visualization of both synthetic and real-world data under VMFormer with temporal modeling modules added.

shows that VMFormer could reach real-time inference and comparable FPS to CNN-based models.

#### 4.5. Visualization

**Challenging Video Frames** We visualize some challenging synthetic video frames from the test set and the corresponding alpha matte predictions as shown in Figure 3. It shows that VMFormer can better distinguish ambiguous foregrounds from backgrounds compared with other CNN-based models like MODNet and RVM, especially on color obfuscation.

**Feature Map Channels** We further visualize some channels of feature maps in Figure 4 and we observe that VMFormer has many redundant empty predictions, which is aligned with observations from other vision transformer works [49]. Moreover, VMFormer generates high-quality feature maps that are useful for accurate alpha matte predictions.

**Temporal Modeling** We also explore how visualizations of alpha matte change as temporal modeling modules are added to the VMFormer model in Figure 5. It shows that adding SFTM and LQTM gradually improves the quality of alpha matte under both synthetic and real-world data as inputs.

### 5. Conclusion

In this paper, we propose VMFormer: an end-to-end transformer-based method for video matting, which uses self-attention in the transformer encoder to model the global integration of feature sequences. It leverages cross-attention in the transformer decoder to model interactions between whole feature sequences and queries as query modeling. We further build a long-range query-based temporal modeling (LQTM) module and a short-range feature-based temporal modeling (SFTM) module to encode temporal information for better predictions of alpha mattes. Extensive experiments show that VMFormer outperforms previous CNN-based methods and has robust performance on other benchmarks with comparable real-time inference speeds.



## References

- [1] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 29–37, 2017. 2
- [2] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3, 6, 7
- [4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6
- [5] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 618–626, 2018. 2
- [6] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 2
- [7] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 3
- [8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021. 3
- [9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3
- [10] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001. 2
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3
- [12] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. A cluster sampling method for image matting via sparse coding. In *European Conference on Computer Vision*, pages 204–219. Springer, 2016. 2
- [13] Marco Forte and François Pitié.  $f$ ,  $b$ , alpha matting. *arXiv preprint arXiv:2003.07711*, 2020. 6
- [14] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005. 2
- [15] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022. 3
- [16] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. 3
- [17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 6, 7
- [18] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021. 3
- [19] Zhanghan Ke, Kaican Li, Yurou Zhou, Qihua Wu, Xiangyu Mao, Qiong Yan, and Rynson WH Lau. Is a green screen really necessary for real-time portrait matting? *arXiv preprint arXiv:2011.11961*, 2020. 2, 3
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 6
- [21] Jiachen Li, Marianna Ohanian, Vidit Goel, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. VideoMatt: A simple baseline for accessible real-time video matting. In *CVPR Workshops*, 2023. 3
- [22] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*, 2021. 2, 5
- [23] 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. 2
- [24] 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. 2, 5, 6, 7, 8
- [25] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. *arXiv preprint arXiv:2108.11515*, 2021. 2, 3, 4, 5, 6, 7, 8
- [26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4
- [27] 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. 8

- [28] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian-sheng Hua. Boosting semantic human matting with coarse annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8572, 2020. [2](#)
- [29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. [2](#), [3](#)
- [30] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016. [4](#)
- [31] GyuTae Park, SungJoon Son, JaeYoung Yoo, SeHo Kim, and Nojun Kwak. Matteformer: Transformer-based image matting via prior-tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11696–11706, June 2022. [2](#)
- [32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. [6](#)
- [33] 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. [2](#)
- [34] 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](#)
- [35] Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*, 2021. [6](#)
- [36] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2291–2300, 2020. [2](#)
- [37] 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. [6](#), [7](#), [8](#)
- [38] 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](#), [5](#)
- [39] supervise.ly. Supervisely person dataset. *supervise.ly*, 2018. [8](#)
- [40] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. [2](#), [3](#)
- [41] Jue Wang and Michael F Cohen. Image and video matting: a survey. 2008. [1](#), [2](#)
- [42] Yu Wang, Yi Niu, Peiyong Duan, Jianwei Lin, and Yuanjie Zheng. Deep propagation based image matting. In *IJCAI*, volume 3, pages 999–1006, 2018. [2](#)
- [43] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021. [3](#)
- [44] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021. [3](#), [7](#)
- [45] 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. [2](#)
- [46] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. [8](#)
- [47] Haichao Yu, Ning Xu, Zilong Huang, Yuqian Zhou, and Humphrey Shi. High-resolution deep image matting. *arXiv preprint arXiv:2009.06613*, 2020. [2](#)
- [48] 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. [2](#)
- [49] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. [8](#)
- [50] Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-sheng Hua, Hujun Bao, Qixing Huang, and Weiwei Xu. Attention-guided temporal coherent video object matting. *arXiv preprint arXiv:2105.11427*, 2021. [2](#)
- [51] Bingke Zhu, Yingying Chen, Jinqiao Wang, Si Liu, Bo Zhang, and Ming Tang. Fast deep matting for portrait animation on mobile phone. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 297–305, 2017. [2](#)
- [52] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. [3](#), [6](#)