Memory Efficient Matting with Adaptive Token Routing

Yiheng Lin[1, 2], **Yihan Hu**[1, 2, 4], **Chenyi Zhang**[1, 2, 4],
Ting Liu[4], **Xiaochao Qu**[4], **Luoqi Liu**[4], **Yao Zhao**[1, 2, 3*], **Yunchao Wei**[1, 2, 3]
[1]Institute of Information Science, Beijing Jiaotong University
[2]Visual Intelligence + X International Joint Laboratory of the Ministry of Education
[3]Pengcheng Laboratory, Shenzhen, China
[4]MT Lab, Meitu Inc
{yiheng.lin, Yihan.hu, chenyi22, yzhao, yunchao.wei}@bjtu.edu.cn, {lt, qxc, llq5}@meitu.com

Abstract

Transformer-based models have recently achieved outstanding performance in image matting. However, their application to high-resolution images remains challenging due to the quadratic complexity of global self-attention. To address this issue, we propose MEMatte, a **m**emory-**e**fficient **m**atting framework for processing high-resolution images. MEMatte incorporates a router before each global attention block, directing informative tokens to the global attention while routing other tokens to a Lightweight Token Refinement Module (LTRM). Specifically, the router employs a local-global strategy to predict the routing probability of each token, and the LTRM utilizes efficient modules to simulate global attention. Additionally, we introduce a Batch-constrained Adaptive Token Routing (BATR) mechanism, which allows each router to dynamically route tokens based on image content and the stages of attention block in the network. Furthermore, we construct an ultra high-resolution image matting dataset, UHR-395, comprising 35,500 training images and 1,000 test images, with an average resolution of 4872×6017. This dataset is created by compositing 395 different alpha mattes across 11 categories onto various backgrounds, all with high-quality manual annotation. Extensive experiments demonstrate that MEMatte outperforms existing methods on both high-resolution and real-world datasets, significantly reducing memory usage by approximately 88% and latency by 50% on the Composition-1K benchmark.

Code — https://github.com/linyiheng123/MEMatte

Introduction

Natural image matting is a crucial task in computer vision that aims to accurately separate the foreground object by predicting the alpha matte. Given a foreground $F \in \mathbb{R}^{H\times W\times C}$, a background $B \in \mathbb{R}^{H\times W\times C}$, and an alpha matte $\alpha \in \mathbb{R}^{H\times W}$, a natural image $I \in \mathbb{R}^{H\times W\times C}$ can be represented as:

$$I_i = \alpha_i F_i + (1-\alpha_i)B_i, \quad \alpha \in [0,1], \tag{1}$$

where H, W, and C denote the height, width, and channel of the image, respectively, and i denotes the pixel index.

*Corresponding author.

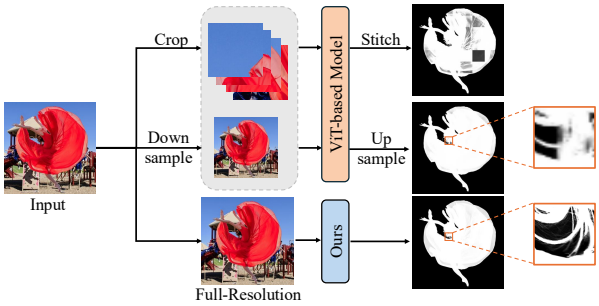

Figure 1: Illustration of the importance of full-resolution input in image matting. The crop-and-stitch manner introduces artifacts, while downsampling manner causes distortion.

In recent years, vision transformers (ViT) have achieved impressive results in image matting (Hu et al. 2025; Yao et al. 2024) due to their long-range modeling capability. Nevertheless, applying ViT-based methods on memory-limited GPUs is challenging as global attention has quadratic complexity and full-resolution input for matting is crucial. As shown in Figure 1, image matting demands high precision for each pixel, and simply downsampling the image leads to distortion. Moreover, a patch-based crop-and-stitch manner is also inappropriate as it destroys global context and introduces artifacts. These limitations raise the question: how can we reduce the memory cost of global attention while maintaining full-resolution input?

A common approach to solve this problem involves token pruning (Rao et al. 2021; Liang et al. 2022) and token merging (Bolya et al. 2023), where a fixed ratio of redundant tokens is progressively discarded at specific network stages. However, this paradigm presents two problems: **P1)** Applying a fixed ratio results in poor performance and inefficiency. For instance, a fixed ratio causes unnecessary computational overhead for simple images and performance degradation for complex images. Additionally, using a fixed ratio across different network stages is inefficient, since global attention behaves like convolution in the early stages (Park and Kim 2022; Ghiasi et al. 2022). **P2)** Progressive token discarding degrades performance, as discarded tokens are excluded from subsequent calculations. This approach risks losing im-

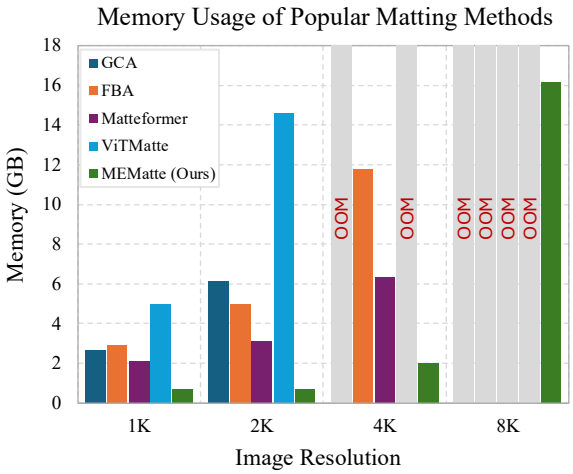

Figure 2: Memory Usage / Image Resolution. OOM denotes an out-of-memory error encountered on an RTX 3090 GPU. The huge increase in memory usage from 4K to 8K of MEMatte is due to the 4x tokens and the quadratic complexity of attention mechanisms. Despite this, MEMatte is capable of processing 8K images on the RTX 3090.

portant tokens in the middle layers, therefore making it unsuitable for precise predictions.

To address these issues, we propose MEMatte. The core idea of MEMatte is to **route informative tokens to the global attention branch while directing other tokens to an efficient branch**. Specifically, a router utilizing a local-global strategy is inserted before each global attention to predict the routing probability for each token. Then a Batch-constrained Adaptive Token Routing mechanism (BATR) is introduced to make the routing decision based on the predicted probability. BATR adaptively adjusts the number of tokens routed to the two branches according to image complexity and the network stage, thus addressing the problem **P1**. Notably, BATR reselects the branch for all tokens at each block, which solves the problem **P2**. We also design a lightweight token refinement module (LTRM) as the efficient branch, which employs depth-wise convolution and efficient channel attention to update the uninformative tokens.

By allocating different computational resources to different tokens, MEMatte significantly decreases memory usage during inference. As shown in Figure 2, MEMatte can process over 4K resolution images on commonly used consumer-level GPUs, such as the RTX 1060, and 8K resolution images on the RTX 3090 GPU. The memory consumption of MEMatte is significantly lower than that of the ViT-based method ViTMatte (Yao et al. 2024), and even more efficient than the Swin-based method MatteFormer (Park et al. 2022) and the CNN-based methods GCA (Li and Lu 2020) and FBA (Forte and Pitié 2020).

We also evaluate MEMatte on Composition-1K dataset, where it reduces memory usage by approximately 88% and inference latency by 50% compared to ViTMatte.

Our contributions are summarized as follows:

- We propose MEMatte, which integrates a router, a Batch-constrained Adaptive Token Routing mechanism, and a lightweight token refinement module, effectively reducing computational overhead during inference.
- We conduct extensive experiments on widely used benchmarks. MEMatte achieves state-of-the-art performance on both high-resolution and real-world datasets.
- We contribute UHR-395, an ultra high-resolution image matting dataset with manual annotations. UHR-395 includes 395 different foreground objects across 11 categories, with an average resolution of 4872×6017.

Related Works

Natural Image Matting

Traditional image matting methods focused on sampling-based (Gastal and Oliveira 2010; He et al. 2011) and propagation-based approaches (Chen, Li, and Tang 2013; He, Sun, and Tang 2010; Levin, Lischinski, and Weiss 2007) to estimate alpha values using known foreground and background colors. With the rise of deep learning, CNN-based methods (Xu et al. 2017; Lu et al. 2019; Li and Lu 2020; Forte and Pitié 2020; Yu et al. 2021b; Sun, Tang, and Tai 2021; Li et al. 2024b) have significantly advanced matting performance. Recently, inspired by the success of transformers (Vaswani et al. 2017; Dosovitskiy et al. 2020; Liu et al. 2021; Li et al. 2022; Liu et al. 2024) in NLP and vision tasks, image matting methods (Park et al. 2022; Cai et al. 2022; Dai et al. 2022; Yao et al. 2024; Li et al. 2024a) have incorporated transformers, yielding impressive performance.

High Resolution Image Matting

Handling high-resolution images in matting is challenging. HDMatte (Yu et al. 2021a) introduced a Cross-Patch Contextual module to model contextual dependencies. BGMv2 (Lin et al. 2021) used a small patch replacement strategy to process high-resolution images. Both of these patch-based methods rely solely on convolutional neural networks, which are ineffective within transformer models because they disrupt the global receptive field of global attention. SparseMat (Sun, Tang, and Tai 2023) utilized dilation and erosion operations to identify active pixels and refined them with sparse convolution (Engelcke et al. 2017). However, this approach often fails to accurately identify pixels requiring correction, particularly with transparent objects.

Token Compression

Common token compression methods typically reduce the number of tokens by token pruning or merging. DynamicViT (Rao et al. 2021) maintained a score vector to prune redundant tokens hierarchically. EViT (Liang et al. 2022) proposed a token reorganization method to identify and fuse image tokens. ToMe (Bolya et al. 2023) introduced a matching algorithm for merging similar tokens. However, these methods encounter out-of-memory errors in high-resolution matting as they retain all tokens during the early stages of the network.

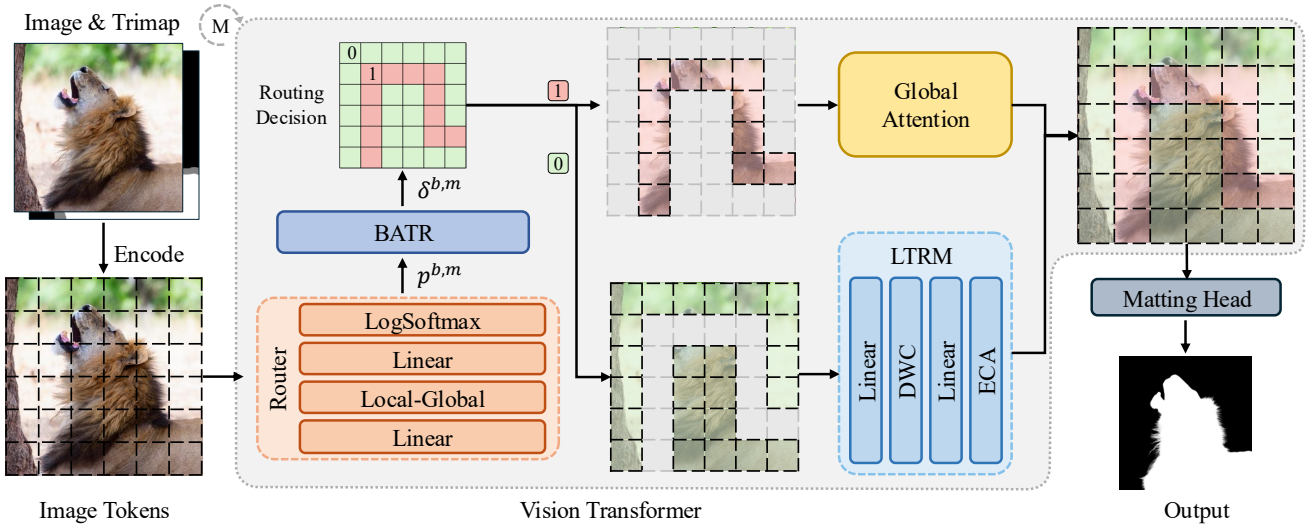

Figure 3: Overall framework of the proposed MEMatte. The router module is inserted before global attention to predict the routing probability $p^{b,m}$ for each token. The BATR then makes the routing decision $\delta^{b,m}$ based on $p^{b,m}$.

Methodology

Preliminaries

In this section, we review the mechanism of global attention. Let the input image be $I \in \mathbb{R}^{H\times W\times C}$. The image is first encoded into tokens $X \in \mathbb{R}^{N\times D}$, where $N = HW/p^2$ represents the number of tokens and p represents the token size. In global attention, X is projected into query $Q \in \mathbb{R}^{N\times D_q}$, key $K \in \mathbb{R}^{N\times D_k}$, and value $V \in \mathbb{R}^{N\times D_v}$. The operation of global self-attention is defined as:

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}})V, \quad (2)$$

where the similarity score computed between query-key pairs leads to an $O(N^2)$ complexity. Specifically, the memory required to store the attention map is approximately $4hN^2$ bytes, where h denotes the number of heads.

These computational and memory challenges underscore the necessity of a more efficient mechanism, prompting us to design MEMatte, which addresses this challenge through adaptive token routing.

Adaptive Token Routing

The overall framework of MEMatte is illustrated in Figure 3. In MEMatte, each transformer block consists of four main components: the **Router** module, the Batch-constrained Adaptive Token Routing (**BATR**) mechanism, the global attention branch, and the Lightweight Token Refinement Module (**LTRM**) branch. After the image tokens enter the transformer block, the router module predicts the routing probability for each token. The BATR then makes the final routing decision, directing tokens to one of the two branches. Finally, the tokens from both branches are merged together.

Router. The router module utilizes a local-global strategy to calculate the routing probability for each token. Given image tokens $X \in \mathbb{R}^{N\times D}$, let $x_i \in \mathbb{R}^D$ represent the i-th token. First, x_i is projected through a layer normalization LN and a linear layer f_θ:

$$z_i = f_\theta(LN(x_i)) \in \mathbb{R}^D. \quad (3)$$

Next, z_i is split along the channel dimension into a local feature z_i^l and a global feature z_i^g. The local feature z_i^l retains the first half of the channels:

$$z_i^l = z_i[:D/2] \in \mathbb{R}^{D/2}, \quad (4)$$

which encodes the information of a single token. The global feature z_i^g is obtained by averaging the second half of the channels over all tokens:

$$z_i^g = \frac{1}{N}\sum_{i=1}^{N} z_i[D/2:] \in \mathbb{R}^{D/2}, \quad (5)$$

which encodes the global contextual information. Then z_i^l and z_i^g are concatenated along the channel dimension:

$$z_i' = Cat(z_i^l, z_i^g) \in \mathbb{R}^D. \quad (6)$$

This local-global strategy ensures that z_i' contains both local and global contextual information. Finally, z_i' is passed to a linear layer f_θ' and a LogSoftmax layer LS to predict the routing probability p_i for the i-th token:

$$log\ p_i = LS(f_\theta'(z_i')) \in \mathbb{R}^2. \quad (7)$$

where $p_{i,0}$ and $p_{i,1}$ represent the probabilities of routing the i-th token to the LTRM branch and the global attention branch, respectively.

Batch-constrained Adaptive Token Routing. BATR possesses two key features that address the issues outlined in the introduction. First, BATR reselects the branch for all tokens in each block, preventing the loss of informative tokens. Second, instead of setting a fixed ratio, BATR adaptively adjusts the number of tokens routed to the two branches based on

image complexity and the network stage. Below, we provide a detailed explanation of how BATR works.

Consider a mini-batch of B samples and M stages of the model, $p_i^{b,m} \in \mathbb{R}^2$ represents the routing probability of the i-th token, where b denotes the sample index and m denotes the stage index. The routing decision $\delta_i^{b,m} \in \{0, 1\}$ is determined as follows:

$$\delta_i^{b,m} = \begin{cases} G(p_i^{b,m}) & \text{Training}, \\ A(p_i^{b,m}) & \text{Inference}. \end{cases} \tag{8}$$

Here, G denotes the Gumbel-Softmax function, which ensures that the routing process is differentiable. A represents the Argmax function, which avoids setting a fixed routing ratio. A value of $\delta_i^{b,m} = 0$ indicates that the i-th token is routed to the LTRM, while $\delta_i^{b,m} = 1$ indicates that the token is routed to global attention.

To ensure that the token routing process is content-aware and stage-aware, we first calculate the average ratio γ of tokens routed to attention across all routers at the batch level:

$$\gamma = \frac{1}{BMN} \sum_{b=1}^{B} \sum_{m=1}^{M} \sum_{i=1}^{N} \delta_i^{b,m} \in [0, 1]. \tag{9}$$

During training, by constraining the value of γ and incorporating matting loss, the model learns to route fewer tokens to the attention branch at the appropriate samples and stages.

Lightweight Token Refinement Module. Due to the high accuracy demands of image matting, simply skipping the global attention for uninformative tokens can result in a performance decline. To address this issue, we employ a Lightweight Token Refinement Module (LTRM), inspired by the Skip-At method (Venkataramanan et al. 2023), to effectively process these tokens.

Given the representation of tokens $X \in \mathbb{R}^{N \times D}$, we first project it using a linear layer f_{θ_1}. Next, we apply a depth-wise convolution DWC to enhance the extraction of texture information in a lightweight manner. Finally, the output is further refined by a linear layer f_{θ_2} and an efficient channel attention module ECA (Wang et al. 2020). The updated tokens Z can be formalized as:

$$Z = ECA(f_{\theta_2}(DWC(f_{\theta_1}(X)))) \in \mathbb{R}^{N \times D}. \tag{10}$$

Adaptation to Ultra High-Resolution Images. For ultra high-resolution images, the routers may direct an excessive number of tokens to the attention branch as they are content-aware and stage-aware. To mitigate this, we set a maximum number of tokens k during inference. If the number of tokens routed to attention exceeds k, only the top k most important tokens are retained based on the scores $p_{i,1}$ provided by the router. Our ablation study shows that this approach has little impact on performance, as the remaining tokens are still updated by the LTRM.

Training Objectives

Distillation loss. To enable the LTRM to emulate the global attention mechanism, we utilize the ViT from ViTMatte as the teacher model. The teacher model guides the output of the MEMatte's ViT to closely approximate that of the original ViT. Given the output features $F_t \in \mathbb{R}^{N \times D}$ from teacher ViT and $F_s \in \mathbb{R}^{N \times D}$ from MEMatte ViT, the distillation loss is defined as:

$$\mathcal{L}_{distill} = \frac{1}{N} \sum_{i=1}^{N} (F_{t,i} - F_{s,i})^2, \tag{11}$$

where i denotes the token index.

Compression loss. To reduce the number of tokens routed to the global attention branch, we introduce a target compression degree $\rho \in [0, 1]$ to constrain the value of γ. Here, ρ is a predefined hyperparameter. The loss item can be formulated as:

$$\mathcal{L}_{compress} = (\rho - \gamma)^2. \tag{12}$$

A smaller value of ρ results in fewer tokens being routed to the global attention branch.

Total loss. We combine the aforementioned losses with the matting loss from ViTMatte to formulate the total loss:

$$\mathcal{L}_{total} = \mathcal{L}_{matting} + \mathcal{L}_{distill} + \mathcal{L}_{compress}. \tag{13}$$

Ultra High-Resolution Dataset

As illustrated in Table 1, we select several widely used image matting datasets for comparison, including DIM (Xu et al. 2017), Distinctions-646 (Qiao et al. 2020), AIM-500 (Li, Zhang, and Tao 2021), PPM-100 (Ke et al. 2022), and Transparent-460 (Cai et al. 2022). These datasets are not suitable benchmarks for high-resolution matting as they fail to satisfy the requirements for both category diversity and high resolution. While Transparent460 and PPM100 offer higher resolutions, the former is primarily composed of transparent objects, and the latter focuses on portraits. Conversely, DIM, Distinctions-646, and AIM-500 suffer from insufficient resolution.

To address these limitations, we propose UHR-395, an ultra high-resolution dataset. UHR-395 comprises 395 foreground objects across 11 categories, such as animals with fur, fire, humans, plants, glass, water, etc. Each object is meticulously annotated by a professional team, with the annotations undergoing multiple validation rounds to ensure high quality. We divide these objects into 355 for training and 40 for testing, producing 35,500 training images and 1,000 test images following the rules in DIM. More details are available in the supplementary materials.

Datasets	Avg Resolution	FN	MC
DIM	1082×1297	481	✓
Distinctions-646	1727×1565	646	✓
AIM-500	1260×1397	500	✓
PPM-100	2875×2997	100	✗
Transparent-460	3772×3804	460	✗
UHR-395 (Ours)	**4872×6017**	395	✓

Table 1: Comparison between different public matting datasets. FN denotes the number of foreground and MC indicates multiple categories. Our dataset is marked in gray.

Datasets / Methods	Composition-1K SAD ↓	MSE ↓	Grad ↓	Conn ↓	Distinctions-646 SAD ↓	MSE ↓	Grad ↓	Conn ↓	$\text{Mem}_{/\text{GB}}$	$\text{Latency}_{/\text{ms}}$	GFLOPs
DIM	50.4	14.0	31.0	50.8	58.73	14.77	77.32	59.89	6.05	142.00	1591.52
IndexNet	45.8	13.0	25.9	43.7	46.73	9.63	43.34	46.86	2.71	94.14	255.29
GCA	35.28	9.00	16.90	32.50	35.33	18.4	28.78	34.29	3.01	179.251	939.45
HDMatt	33.5	7.3	14.5	29.9	-	-	-	-	-	-	-
MGMatting	31.5	6.8	13.5	27.3	33.24	4.5	20.31	25.49	1.96	112.01	434.92
SIM	28.0	5.8	10.8	24.8	23.60	3.93	20.05	22.20	3.49	550.14	2200.43
FBA	25.8	5.2	10.6	20.8	32.37	5.47	24.15	31.70	3.08	538.03	1497.57
TransMatting	24.96	4.58	9.72	20.16	-	-	-	-	-	-	-
Matteformer	23.80	4.03	8.68	18.90	23.90	8.16	12.65	18.90	2.20	220.83	546.61
RMat	22.87	3.9	7.74	17.84	-	-	-	-	-	-	-
ViTMatte-S	21.46	3.3	7.24	16.21	23.43	2.41	11.49	19.02	6.20	186.00	743.03
MEMatte-S	21.90	3.37	7.43	16.77	21.54	2.46	11.06	18.27	$\mathbf{0.71}_{\mathbf{88.5\%}\downarrow}$	$\mathbf{84.99}_{\mathbf{54.3\%}\downarrow}$	661.14
ViTMatte-B	20.33	3.0	6.74	14.78	22.09	1.93	9.59	17.26	12.53	340.15	1710.14
MEMatte-B	21.06	3.11	6.70	15.71	20.02	1.92	9.39	16.97	$\mathbf{1.49}_{\mathbf{88.1\%}\downarrow}$	$\mathbf{178.90}_{\mathbf{47.4\%}\downarrow}$	1432.92

Table 2: Quantitative results on Synthetic Dataset. The memory usage, latency, and GFLOPs are evaluated using the Composition-1K dataset, averaged across images. A dash (-) indicates no official code release. Our method is marked in gray.

Experiments

Datasets and Evaluation

Synthetic Dataset. Due to the high annotation costs, available datasets for image matting are limited. DIM and Distinction-646 composite foreground images with background images from the COCO (Lin et al. 2014) and VOC2012 (Everingham et al. 2010) datasets using Eq. 1 for both training and evaluation. Following other matting methods, we use Composition-1K to represent the DIM test set.

High-Resolution Dataset. Due to the large resolution of our UHR-395 test set (average 5318×7051), most popular image matting methods encounter out-of-memory errors. For a more detailed comparison, we select 344 images larger than 4K resolution from the PPM-100 and Transparent-460 datasets, naming this subset PPT-344, with an average resolution of 3962×4170.

Real-world Dataset. AIM500 is a real-world test dataset that encompasses various objects, including portraits, animals, transparent objects, and fruits. We choose this dataset to evaluate the ability of existing image matting methods to generalize to real-world scenarios.

Following existing methods, we train our model on the DIM training dataset and evaluate it on synthetic, high-resolution, and real-world datasets to demonstrate the superiority of our method in terms of low memory cost and strong generalization ability. We compare our method with popular matting methods introduced in related works, using four commonly used metrics: Sum of Absolute Differences (SAD), Mean Square Error (MSE), Gradient loss (Grad), and Connectivity loss (Conn). Note that lower values of these metrics indicate better alpha matte quality. All experiments are performed on the RTX 3090. More implementation details are in the supplementary material.

Methods	SAD ↓	MSE ↓	Grad ↓	Conn ↓
FBA_D	1062.09	32.49	1196.34	1072.20
FBA_P	1273.63	30.91	456.86	1296.05
Matteformer_D	860.71	23.27	1158.63	823.68
Matteformer_P	3386.12	22.83	2566.05	3478.81
SparseMat	2779.50	20.92	2079.01	2324.84
ViTMatte-S_D	720.21	19.99	1026.79	668.02
ViTMatte-S_P	843.87	17.06	247.51	820.58
ViTMatte-B_D	669.44	17.72	983.17	622.50
ViTMatte-B_P	727.36	12.79	224.95	693.02
MEMatte-S	623.93	11.13	220.09	572.94
MEMatte-S*	**505.88**	**8.50**	**172.99**	**423.76**

Table 3: Quantitative results on our proposed UHR-395. D denotes downsampling the input and P represents dividing the input into patches. $*$ indicates that the model is fine-tuned on our UHR-395 training set.

Main Results

Results on Synthetic Dataset. The quantitative results on the synthetic dataset are shown in Table 2. Our method significantly reduces the computational overhead of the ViTMatte baseline, the average memory usage decreased by 5.49 GB (**-88.5%**) when using ViT-S and by 11.04 GB (**-88.1%**) when using ViT-B. Despite this significant memory cost reduction, the performance degradation on the Composition-1K test set is only around 5%. Moreover, the performance exhibits an improvement when generalized to the Distinction-646 test set.

Results on High-Resolution Dataset. Due to the high resolution of our proposed UHR-395 test set, we evaluate several popular matting methods under two different settings to avoid out-of-memory errors. In the first setting, inference is

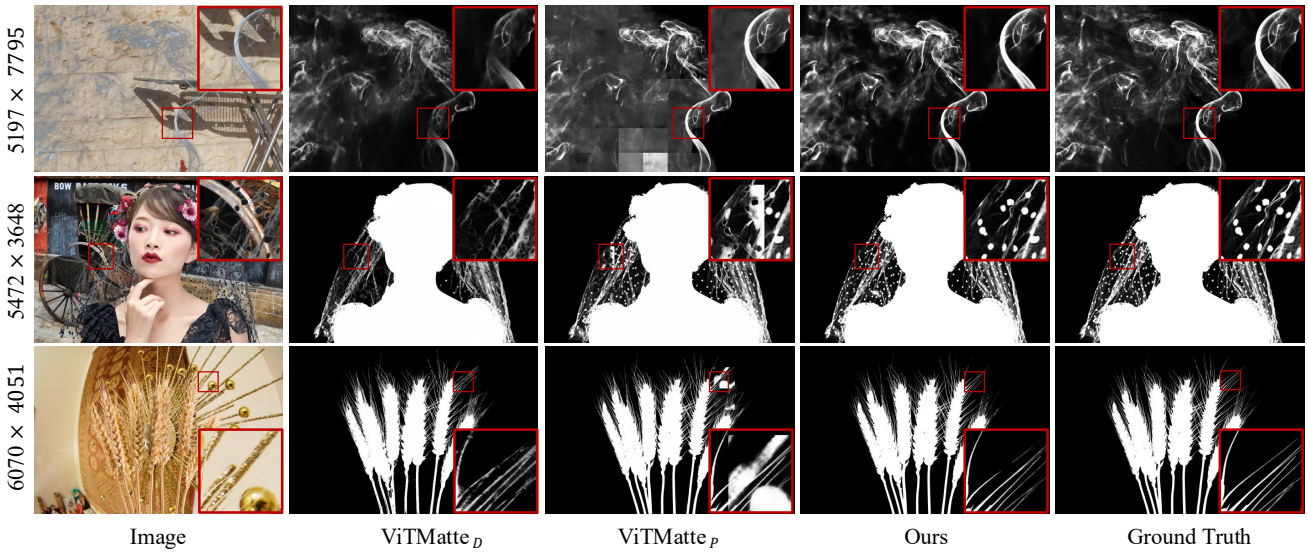

Figure 4: Qualitative comparison of the results on the UHR-395 test set. D denotes downsampling the input and P indicates dividing the input into patches. The resolution of each image is on the left.

Methods	SAD ↓	MSE ↓	Grad ↓	Conn ↓
SparseMat	799.12	20.87	430.96	794.15
FBA	202.94	20.74	80.14	213.16
Matteformer	145.65	14.72	38.759	145.86
ViTMatte-S$_D$	166.33	13.46	127.15	159.40
ViTMatte-S$_P$	141.98	12.80	42.03	142.31
MEMatte-S	**117.05**	**7.91**	**33.38**	**113.03**
ViTMatte-B$_D$	155.14	12.51	118.77	147.82
ViTMatte-B$_P$	151.86	15.02	37.28	152.47
MEMatte-B	**114.95**	**7.92**	**29.85**	**109.54**

Table 4: Quantitative results on PPT-344. D denotes downsampling the input and P indicates dividing the input into patches. Our method is marked in gray.

Methods	SAD ↓	MSE ↓	Grad ↓	Conn ↓
GCA	34.94	38.84	25.76	35.31
MGMatting	55.14	134.35	40.55	53.67
FBA	19.10	16.24	11.46	18.36
Matteformer	26.90	29.01	23.00	26.63
ViTMatte-S	19.57	15.8	12.78	18.85
MEMatte-S	**18.88**	**15.2**	**12.42**	**18.08**
ViTMatte-B	17.33	16.48	14.17	16.30
MEMatte-B	**16.99**	**15.4**	**13.93**	**16.04**

Table 5: Quantitative results on Real-world Dataset AIM500.

performed on downsampled images ($i.e.$ 1024×1024) and then upsampled to the original resolution. In the second setting, a crop-and-stitch strategy is used, conducting inference on cropped patches ($i.e.$ 512×512). The quantitative results on our UHR-395 test set, shown in Table 3, clearly demonstrate the superiority of our method. Besides, as visualized in Figure 4, the downsampling strategy leads to distortion, while the crop-and-stitch strategy introduces artifacts. In contrast, our method produces high-quality output.

Additionally, we evaluate these methods on the PPT-344 dataset, which has a lower resolution, enabling a comprehensive comparison with full-resolution input. ViTMatte still employs the two settings above due to out-of-memory error. As presented in Table 4, our method consistently outperforms the others.

Results on Real-world Dataset. Given the lack of real-world images for training, existing methods typically train their models on synthetic images. Consequently, the performance of the model on real-world dataset becomes critically important. As shown in Table 5, our method demonstrates great generalization ability on real-world images and achieves state-of-the-art performance. We also provide visualizations in supplementary materials.

Visualization of Token Routing. To further investigate the routing mechanism of MEMatte, we visualize the tokens routed to global attention by different routers in Figure 5. We observe several key points: (1) Routers preferentially route informative tokens to global attention, such as the edges of target objects and areas with complex textures. (2) Across different scenarios, the distribution of selected tokens remains sparse, yet the results are consistently excellent. (3) The first router directs only a few tokens to global attention. This is reasonable because global attention in the early network stages behaves like convolution (Park and Kim 2022; Ghiasi et al. 2022), preferring to capture general edges and textures, which are effectively handled by the LTRM.

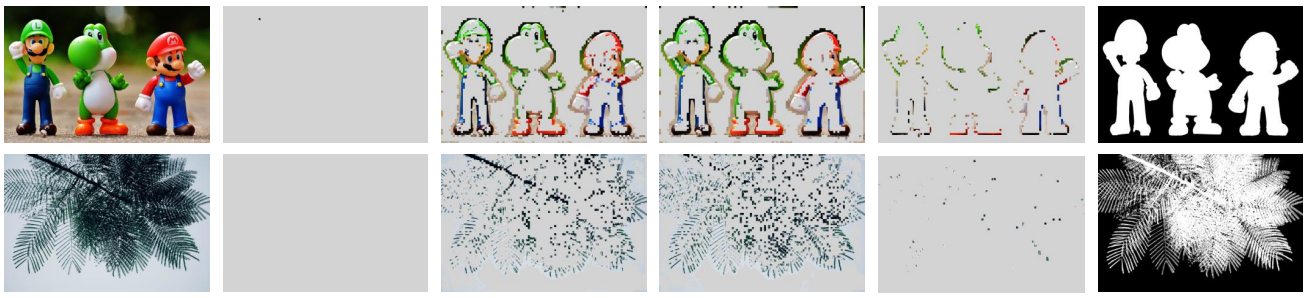

Figure 5: Visualization of the token routing. The retained tokens are routed to global attention branch, while the gray tokens are routed to the LTRM branch. More visualization results are shown in supplementary materials.

Method	SAD ↓	MSE ↓	$\text{Mem}_{/\text{GB}}$	$\text{Latency}_{/\text{ms}}$
ViTMatte	21.46	3.30	6.20	186.00
+ DynamicViT	33.47	9.16	4.20	104.35
+ EViT	42.66	14.25	4.19	67.58
+ ToMe	34.19	7.94	4.21	213.80
+ BATR	**21.90**	**3.37**	**0.71**	84.99

Table 6: Comparisons of different token compression mechanisms. Notably, all methods are based on the ViT-S backbone and are fine-tuned on the DIM training dataset.

Ablation Study

Comparisons of Different Token Compression Methods. To assess the efficacy of our adaptive token routing method in comparison to other token compression methods, we replace the backbone of the ViTMatte with efficient ViT proposed by popular token pruning or merging methods, such as DynamicViT, EViT, and ToMe. As shown in Table 6, these methods exhibit poor performance as they progressively discard a fixed ratio of redundant tokens at specific network stages, leading to the issues mentioned in the introduction. Additionally, these methods also exhibit high memory usage as they retain most tokens during the early stages.

Effect of Reducing Maximum Token Number. Intuitively, reducing maximum token number k might lead to a significant performance drop. However, as shown in Figure 6, as k decreases, the performance metrics remain stable while the memory cost drops rapidly. We attribute this robustness to the effectiveness of the token scoring approach and LTRM.

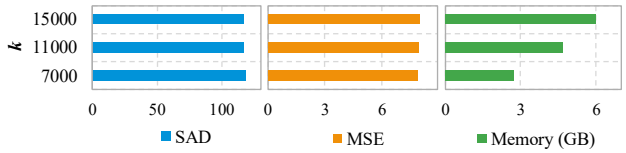

Figure 6: Effect of reducing maximum token number k.

Effect of Target Compression degree. Table 7 shows the effects of target compression degree ρ. As the ρ decreases from 0.75 to 0.1, performance declines until reaching 0.25, beyond which it deteriorates quickly. Consequently, we select 0.25 as the default setting during training for a better trade-off between performance and efficiency.

ρ	SAD ↓	MSE ↓	$\text{Mem}_{/\text{GB}}$	$\text{Latency}_{/\text{ms}}$
0.75	21.78	3.32	4.22	166.09
0.50	21.80	3.34	1.41	99.24
0.25	21.90	3.37	0.71	84.99
0.10	22.68	3.65	0.65	83.43

Table 7: Comparisons of different target compression degree ρ for training. The default setting of our method is marked in gray.

Distill	LTRM	SAD ↓	MSE ↓	$\text{Mem}_{/\text{GB}}$	$\text{Latency}_{/\text{ms}}$
		22.67	3.52	0.74	82.80
✓		22.66	3.60	0.71	79.00
	✓	22.54	3.55	0.75	83.17
✓	✓	**21.90**	**3.37**	**0.71**	83.43

Table 8: Effect of distillation loss and LTRM. Distill denotes distillation loss.

Effect of Distillation and LTRM. Table 8 reports the effectiveness of the distillation loss and LTRM. Using either the distillation loss or LTRM alone yields similar performance to using neither of them. However, combining both of them results in a significant performance improvement. This suggests that the teacher model successfully teaches the LTRM to imitate the mechanism of global attention.

Conclusion

In this paper, we propose MEMatte and an ultra high-resolution dataset UHR-395. MEMatte significantly reduces memory usage and latency during inference compared to existing matting methods, achieving state-of-the-art performance on both high-resolution and real-world datasets. These results highlight the robustness of our token routing mechanism and LTRM. We hope that MEMatte will promote further research on high-resolution matting.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.92470203, No.U23A20314, No.62120106009), Beijing Natural Science Foundation (No. L242022), the Fundamental Research Funds for the Central Universities (2024XKRC082).

References

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2023. Token Merging: Your ViT but Faster. In *International Conference on Learning Representations*.

Cai, H.; Xue, F.; Xu, L.; and Guo, L. 2022. Transmatting: Enhancing transparent objects matting with transformers. In *European conference on computer vision*, 253–269. Springer.

Chen, Q.; Li, D.; and Tang, C.-K. 2013. KNN matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9): 2175–2188.

Dai, Y.; Price, B.; Zhang, H.; and Shen, C. 2022. 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*, 11707–11716.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Engelcke, M.; Rao, D.; Wang, D. Z.; Tong, C. H.; and Posner, I. 2017. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1355–1361. IEEE.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.

Forte, M.; and Pitié, F. 2020. F, B, Alpha Matting. *arXiv preprint arXiv:2003.07711*.

Gastal, E. S.; and Oliveira, M. M. 2010. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, 575–584. Wiley Online Library.

Ghiasi, A.; Kazemi, H.; Borgnia, E.; Reich, S.; Shu, M.; Goldblum, M.; Wilson, A. G.; and Goldstein, T. 2022. What do vision transformers learn? a visual exploration. *arXiv preprint arXiv:2212.06727*.

He, K.; Rhemann, C.; Rother, C.; Tang, X.; and Sun, J. 2011. A global sampling method for alpha matting. In *CVPR 2011*, 2049–2056. Ieee.

He, K.; Sun, J.; and Tang, X. 2010. Fast matting using large kernel matting laplacian matrices. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2165–2172. IEEE.

Hu, Y.; Lin, Y.; Wang, W.; Zhao, Y.; Wei, Y.; and Shi, H. 2025. Diffusion for natural image matting. In *European Conference on Computer Vision*, 181–199. Springer.

Ke, Z.; Sun, J.; Li, K.; Yan, Q.; and Lau, R. W. 2022. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1140–1147.

Levin, A.; Lischinski, D.; and Weiss, Y. 2007. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2): 228–242.

Li, J.; Goel, V.; Ohanyan, M.; Navasardyan, S.; Wei, Y.; and Shi, H. 2024a. Vmformer: End-to-end video matting with transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6678–6687.

Li, J.; Zhang, J.; and Tao, D. 2021. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*.

Li, Y.; Huang, Z.; Yu, G.; Chen, L.; Wei, Y.; and Jiao, J. 2024b. Disentangled Pre-training for Image Matting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 169–178.

Li, Y.; and Lu, H. 2020. Natural image matting via guided contextual attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11450–11457.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, 280–296. Springer.

Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations. In *International Conference on Learning Representations*.

Lin, S.; Ryabtsev, A.; Sengupta, S.; Curless, B. L.; Seitz, S. M.; and Kemelmacher-Shlizerman, I. 2021. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8762–8771.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, H.; Tan, Z.; Tan, C.; Wei, Y.; Wang, J.; and Zhao, Y. 2024. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10770–10780.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Lu, H.; Dai, Y.; Shen, C.; and Xu, S. 2019. Indices Matter: Learning to Index for Deep Image Matting. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*.

Park, G.; Son, S.; Yoo, J.; Kim, S.; and Kwak, N. 2022. Matteformer: Transformer-based image matting via prior-tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11696–11706.

Park, N.; and Kim, S. 2022. HOW DO VISION TRANSFORMERS WORK? In *10th International Conference on Learning Representations, ICLR 2022*.

Qiao, Y.; Liu, Y.; Yang, X.; Zhou, D.; Xu, M.; Zhang, Q.; and Wei, X. 2020. Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13676–13685.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34: 13937–13949.

Sun, Y.; Tang, C.-K.; and Tai, Y.-W. 2021. Semantic image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11120–11129.

Sun, Y.; Tang, C.-K.; and Tai, Y.-W. 2023. Ultrahigh resolution image/video matting with spatio-temporal sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14112–14121.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Venkataramanan, S.; Ghodrati, A.; Asano, Y. M.; Porikli, F.; and Habibian, A. 2023. Skip-attention: Improving vision transformers by paying less attention. *arXiv preprint arXiv:2301.02240*.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11534–11542.

Xu, N.; Price, B.; Cohen, S.; and Huang, T. 2017. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2970–2979.

Yao, J.; Wang, X.; Yang, S.; and Wang, B. 2024. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103: 102091.

Yu, H.; Xu, N.; Huang, Z.; Zhou, Y.; and Shi, H. 2021a. High-resolution deep image matting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3217–3224.

Yu, Q.; Zhang, J.; Zhang, H.; Wang, Y.; Lin, Z.; Xu, N.; Bai, Y.; and Yuille, A. 2021b. Mask guided matting via progressive refinement network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1154–1163.