# DRIP: Unleashing Diffusion Priors for Joint Foreground and Alpha Prediction in Image Matting

**Xiaodi Li**[1,2], **Zongxin Yang**[3], **Ruijie Quan**[1,2], **Yi Yang**[1,2*]
[1] The State Key Lab of Brain-Machine Intelligence, Zhejiang University, Hangzhou, China
[2] CCAI, College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3] DBMI, HMS, Harvard University, Boston, USA
https://github.com/LiNO3Dy/Drip

## Abstract

Recovering the foreground color and opacity/alpha matte from a single image (*i.e.*, image matting) is a challenging and ill-posed problem where data priors play a critical role in achieving precise results. Due to the limited matting datasets, traditional methods usually struggle to produce high-quality estimation.To address this, we explore the potential of leveraging vision priors embedded in pre-trained latent diffusion models (LDM) for estimating foreground RGBA values in challenging scenarios and rare objects. We introduce `Drip`, a novel approach for image matting that harnesses the rich prior knowledge of LDM models. Our method incorporates a switcher and a cross-domain attention mechanism to extend the original LDM for joint prediction of the foreground color and opacity. This setup facilitates mutual information exchange and ensures high consistency across both modalities. To mitigate the inherent reconstruction errors of the LDM's VAE decoder, we propose a latent transparency decoder to align the RGBA prediction with the input image, thereby reducing discrepancies. Comprehensive experimental results demonstrate that our approach achieves state-of-the-art performance in foreground and alpha predictions and shows remarkable generalizability across various benchmarks.

## 1 Introduction

Image matting aims to isolate the foreground object from composited images, a long-standing and fundamental task in vision intelligence [1]. It is indispensable for various downstream applications, such as media production, virtual reality, and image/video editing [2, 3]. Mathematically, image matting begins with solving the inverse problem of the composition equation:

$$Image_i = \alpha_i \cdot Foreground_i + (1 - \alpha_i) \cdot Background_i, \quad \alpha_i \in [0, 1], \tag{1}$$

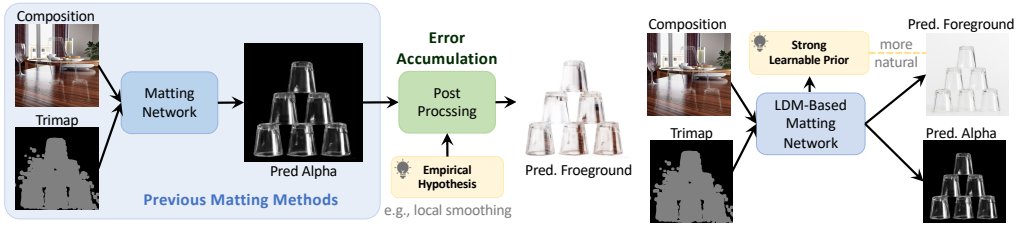
where $i$ denotes the index of a pixel. Here, all quantities on the right-hand side are unknown, and the prediction of the alpha matte $\alpha$ and foreground color represents an ill-posed problem.

In the past decade, advances in deep learning have significantly pushed the boundaries of image matting, rapidly becoming the mainstream direction in this field [2, 4–7]. Despite their impressive performance, two challenges remain unresolved in this domain: **(i) high-quality foreground color prediction**. As illustrated in Fig. 1, most matting methods consist of two stages: namely, alpha prediction with neural networks and foreground isolation via post-processing. These methods typically struggle to generalize and recover high-fidelity foregrounds due to the accumulated errors in alpha prediction and post-processing. **(ii) accurate prediction of semi-transparent objects**. When the target to be predicted contains large areas of semi-transparency (*e.g.*, a water glass) or

*Corresponding author: Yi Yang.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).


https://doi.org/10.52202/079017-2537




(a) Previous Methods (b) Drip (ours)

Figure 1: **(a)** Matting methods [20, 4–7] commonly predict the alpha matte and then infer the foreground color by post-processing [21], which often relies on empirical assumptions such as local smoothing of the foreground and background, leading to the accumulation of errors. **(b)** In contrast, our joint prediction approach estimates both the foreground and alpha simultaneously. By leveraging LDM's [8] powerful natural vision prior, our predicted foreground is closer to natural images.

semi-transparent regions with high-frequency details (*e.g.*, patterned semi-transparent fabric), existing methods still struggle to predict the alpha matte accurately.

High-quality semitransparent matting data is difficult to annotate on a large scale. Therefore, *how can we enhance the algorithm's generalization capability for semi-transparent objects and achieve high-quality foreground prediction with limited training data?* Recently, with the emergence of large pre-trained generative models [8–12], their data priors learned from billions of images (*e.g.*, LAION [13]) are found to be useful for various downstream tasks [14–18]. Back to the challenging image matting, we posit that the data priors learned from tons of natural images are also intuitively beneficial. Hence, our key insight is to unleash the data priors from the large pre-trained generative models (LDM) to estimate alpha and foreground simultaneously.

To achieve this, we propose `Drip`, the unleashing Diffusion priors for joint foreground and alpha prediction method, which follows the diffusion paradigm and jointly generates the foreground and alpha map conditioned on the image and trimap input. Specifically, we wisely design a **cross-domain switcher** that leverages domain-aware embedding to unify the foreground and alpha generations in a single-diffusion model. This design facilitates mutual information exchange and ensures high consistency between foreground image and alpha. Besides, the pre-trained VAE compresses the image into a compact latent space, significantly reducing training consumption while inevitably missing detailed information. To narrow the errors caused by VAE, we introduce an auxiliary **latent transparency decoder**, which is implemented by inserting the features from early layers in the encoder into the decoder with several learnable zero-conv layers [19]. This latent transparency decoder significantly contributes to high-fidelity foreground image and alpha prediction and also effectively adapts the pre-trained LDM into image matting.

We extensively evaluate the performance of our method through extensive experiments and comparisons. The results demonstrate that our approach achieves state-of-the-art performance on the Composition-1k test set and exhibits stronger generalizability on other benchmark datasets. Remarkably, `Drip` outperforms all the previous methods in the mainstream benchmark, Composition-1k, where `Drip` improves the SAD metric of alpha prediction by $\mathbf{3.3}\%$ and foreground by $\mathbf{12.1}\%$ and MSE metric of alpha by $\mathbf{6.1}\%$ and foreground by $\mathbf{28.33}\%$. In summary, the key contributions of this paper are as follows.

- To our best knowledge, we introduce the first LDM-driven matting method, `Drip`, which effectively unleashes the data priors learned from LDM into image matting.
- To enable joint prediction of foreground and alpha, we propose a switcher and a cross-domain attention mechanism, facilitating mutual information exchange and ensuring high consistency.
- To mitigate the inherent reconstruction errors of the LDM's VAE decoder, we propose a latent transparency decoder to align the RGBA prediction with the input image.

## 2 Related Work

**Image Matting** is aimed to extract the foreground objects from arbitrary natural images [22, 2]. Traditional methods always need the auxiliary user input like trimap [23, 4] and scribble [24, 25]. These methods basically only leverage low-level color or structure features, which limits their ability

to distinguish foreground details from images. With the success of deep learning, researchers have begun to use deep convolutional neural networks (CNNs) to predict the alpha map in an end-to-end fashion [26, 20]. One type takes images and auxiliary trimap or scribble as input and outputs the alpha map [5]. In order to alleviate the demand for trimap, trimap-free methods [27, 6, 28] are proposed to directly predict alpha mattes from the input image, which increases efficiency while sacrificing performance. Although these existing methods achieve impressive results in alpha prediction, accurately predicting foreground and background colors remains an essential yet challenging task for high-quality matting. Tang *et al.* [29] and Aksoy *et al.* [30] firstly proposed to address color estimation by sequentially or directly predicting the background and foreground colors before alpha prediction. Furthermore, recent method [31] unifies foreground, background, and alpha matte into an end-to-end framework. While, another line of works [32, 28] focuses on foreground human extraction and alpha matte prediction. However, these methods are limited by the lack of high-quality labeled data. Meanwhile, the explosion of generative models shows immense potential in providing priors in different tasks. In this work, we explore unleashing diffusion priors within stable diffusion [8] to improve the performance of image matting.

**Diffusion Models** have emerged as a powerful class of generative models, which learn a reverse denoised process from the Gaussian noise to natural images [33]. In the vanilla DDPMs [33], the sampling process is time-consuming due to the Markovian property. To speed up the sampling, DDIMs [34] is proposed to provide a non-Markovian shortcut. Furthermore, LCM [35] just formulate the diffusion process as one-step denoising via an ODE. Besides the speed, a series of works [36, 10, 37, 38] focus on increasing the controllability of diffusion models. For instance, Controlnet [19] fine-tunes a Stable Diffusion model with zero convolutions, which proves to be effective in adapting the pre-trained diffusion models to different tasks by adding different conditions.

**Diffusion Priors in Visual Perceptive Tasks** are prevalent and hot topics. A series of works leverage the diffusion priors in segmentation [17], image enhancement [39], depth estimation [40] and 3D vision [41, 42]. In the context of image matting, Xu *et al.* [43] propose to formulate alpha prediction as a denoised process, and train a condition generation model in DDPMs fashion. However, the vanilla DDPMs have not been scaled-up training due to their expensive computational cost. On the contrary, LDM [8] proposes to compress the features into a compact latent space, which obviously reduces the computational cost. And based on it, Stable Diffusion is largely trained on the large-scale dataset [13]. However, LDM generates the image features in the latent space encoded by pretrained VAE. In order to alleviate the domain gap, Marigold [16] finetunes UNet backbone of diffusion models to perform affine invariant monocular depth estimation and exhibit strong generalization capability. Inspired by this, we carefully discuss and propose a novel method to unleash the diffusion priors within stable diffusion to improve the performance of image matting while preserving high-fidelity details.

## 3 Drip

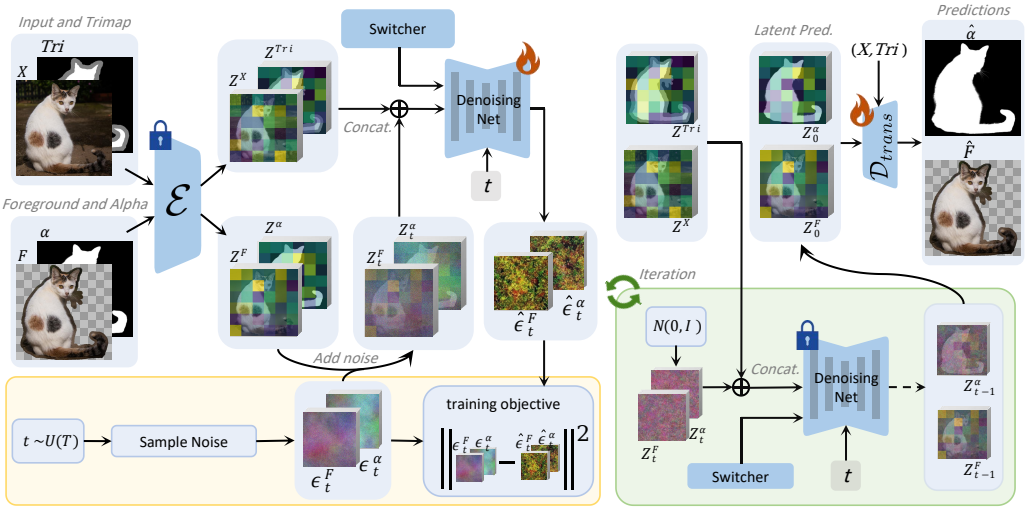
**Overview**. Drip is an LDM-based matting model designed to predict both foreground and alpha values while ensuring high consistency between these two representations. Given an input image ($X$) and a trimap ($Tri$) indicating the object to be matted, our goal is to estimate its corresponding Foreground ($F$) and alpha ($\alpha$). Initially, we explore the problem using the diffusion paradigm (see Sec. §3.1). Subsequently, we present our LDM-based matting model (see Sec. §3.2). This model employs a cross-domain switcher to simultaneously generate the foreground color and alpha map using a single diffusion model. Moreover, through mutual information exchange, the model effectively enhances boundary and texture consistency. To address the challenge of missing high-frequency information caused by VAE compression, the model incorporates an auxiliary latent transparency decoder (see Sec. §3.3). An overview of Drip is provided in Fig. 2.

### 3.1 Problem Formulation

The task of foreground and alpha estimation is to model the mapping $f(\cdot) : (X, Tri) \rightarrow (F, \alpha)$, where $F \in R^{H\times W\times 3}$ represents the foreground and $\alpha \in R^{H\times W}$ represents the alpha map. The input conditions are an RGB image $X \in R^{H\times W\times 3}$ and a trimap $Tri \in R^{H\times W}$, which consists of three values indicating the foreground, unknown, and background regions, respectively. However, unlike prior works that adopt CNN or transformer as architecture, we employ a diffusion-based scheme $f(\cdot)$ to model the joint foreground and alpha distribution $p(F, \alpha)$.

Diffusion Probabilistic Models [44, 33] define a forward Markov chain that progressively transits the sample $x$ drawn from data distribution $p(x)$ into noisy versions $x_t \in (1, T)|x_t = \alpha_t x_0 + \sigma_t \epsilon$, where


https://doi.org/10.52202/079017-2537




(a) Training (b) Inference

Figure 2: **Overview of** `Drip`. **(a)** During training, the input image $X$, trimap $Tri$, ground-truth foreground $F$, and ground-truth alpha map $\alpha$ are first encoded into latent representations $Z^X$, $Z^{Tri}$, $Z^F$, and $Z^\alpha$ respectively using the original Stable Diffusion VAE encoder $\mathcal{E}$. After adding noise to $Z^F$ and $Z^\alpha$, all the latents are fed into a U-Net, which generates the output in the foreground or alpha domain guided by a switcher (§3.2.1). The U-Net is then fine-tuned by optimizing the standard diffusion objective(§3.2.3). **(b)** After executing the T-step denoising schedule, the resulting latents $Z_0^F$ and $Z_0^\alpha$ are decoded by a transparent latent decoder (§3.3).

$\epsilon \sim N(0, I)$, $T$ is the timestep, $\alpha_t$and $\sigma_t$ are the noisy scheduler terms that control sample quality. In the reverse Markov chain, it learns a denoising network $\epsilon_\theta(\cdot)$ parameterized by $\epsilon$ usually structured as U-Net [45] to transform $x_t$ into $x_{t-1}$ from an initial Gaussian sample $x_T$ through iterative denoising.

For the joint foreground and alpha distribution $p(F, \alpha)$, given a conditional input image $X$ with its corresponding trimap $Tri$, the foreground $F$ and the alpha map $\alpha$ can be obtained by the generative formulation in Markov probabilistic form:

$$f(X, Tri) = p\left(\hat{F_T}, \hat{\alpha_T}\right) \prod_{t=1}^{T} p_\theta(\hat{F_{t-1}}, \hat{\alpha_{t-1}} | \hat{F}_t, \hat{\alpha}_t, X, Tri), \quad \hat{F_T}, \hat{\alpha_T} \sim N(0, I). \tag{2}$$

To enhance computational efficiency and generate higher-resolution images, Stable Diffusion [8] employs the latent diffusion model, where the diffusion steps are performed in the low-dimensional latent space instead of directly operating on the original data. The latent space is formed within the bottleneck of VAE [46], which is trained separately from the denoiser. This design allows latent space compression and facilitates perceptual alignment with the data space.

To translate our formulation (Eq. 2) into the latent space, we obtain the corresponding latent code for a given image using an encoder: $z^{(i)} = \mathcal{E}(i)$, where $i \in X, Tri, F, \alpha$. It's worth noting that we triplicate the single-channel trimap and alpha map into three channels. Moreover, the denoiser $\epsilon_\theta(\cdot)$ is subsequently trained in the latent space. To obtain the desired outputs, given latent codes $z_F$ and $z_\alpha$, the foreground and alpha can be reconstructed using the decoder $\mathcal{D}$: $\hat{F} = \mathcal{D}(z^F)$ and $\hat{\alpha} = \mathcal{D}(z^\alpha)$. It is worth noting that in the Matting task if the reconstructed image is obtained directly from the latent representation of the foreground and alpha without making any modifications to the VAE, a significant error can occur.

### 3.2 LDM-Based Matting Model

We base our model on a pretrained text-to-image LDM (Stable Diffusion v2 [8]), which has learned strong and generalizable image priors from LAION-5B [13]. In order to accept a given image and trimap as conditions and simultaneously generate both foreground and alpha outputs, we quadruple the input of the original U-Net and employ a switcher mechanism to expand the capabilities of the original LDM model. Additionally, we incorporate cross-domain attention to enhance consistency.

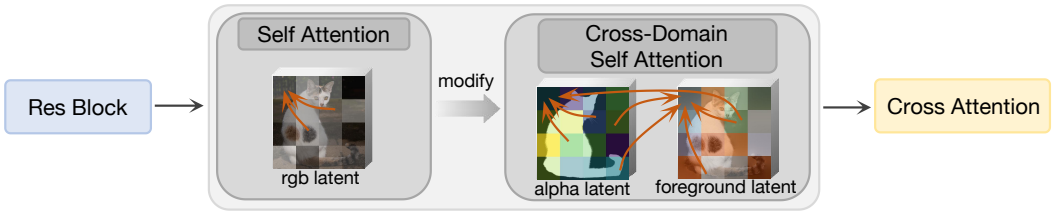



Figure 3: **Demonstration of Cross-Domain Attention**(§3.2.2). To enhance mutual guidance and ensure contextual consistency, we instead utilize a cross-domain self-attention mechanism instead of self-attention to associate the foreground and alpha latent.

### 3.2.1 Foreground and Alpha Switcher

Previous matting models [20, 4–6] have primarily focused on predicting the alpha value and utilizing post-processing methods, such as the local smoothness assumption, to estimate the foreground color [21, 47]. However, this two-stage approach often leads to error accumulation and suboptimal foreground estimation results, particularly when dealing with transparent objects. To address this limitation, we propose a novel approach that leverages an LDM-based method to predict the foreground and alpha values simultaneously. Our method generates more realistic foreground images by incorporating a strong natural image prior distribution learned from Stable Diffusion v2 [8].

To incorporate foreground and alpha estimation, one straightforward approach is to finetune two U-Nets separately to model their respective distributions. However, this method introduces additional parameters and fails to capture the inherent connections between foreground and alpha. Motivated by the work [48, 14], we propose a novel approach using a switcher that enables a single stable diffusion model to generate both foreground and alpha values based on indicators. Mathematically, the foreground and alpha values can be obtained as follows:

$$\hat{F} = f(X, Tri, s_F) = f(X, Tri, \text{PosEnc}(1)) \tag{3}$$

$$\hat{\alpha} = f(X, Tri, s_\alpha) = f(X, Tri, \text{PosEnc}(0)) \tag{4}$$

In the above equations, $s_F$ and $s_\alpha$ are one-dimensional vectors controlling the foreground and alpha domains, respectively. The switchers are encoded using low-dimensional positional encoding and combined with time embedding within the U-Net architecture.

### 3.2.2 Cross-Domain Attention

To further facilitate mutual-guided optimization, we introduce a modification to the self-attention layer in the U-Net architecture, transforming it into a cross-domain self-attention layer that encourages spatial alignment (refer to Fig 3). This operator enhances the geometric consistency between the foreground and alpha channels and accelerates convergence. The cross-domain attention operation, denoted as $AttCD(\cdot)$, is defined as follows:

$$AttCD(Q_i, K_i, V_i) = Att\left(W_q \cdot h_i, W_k \cdot (h_F \oplus h_\alpha), W_v \cdot (h_F \oplus h_\alpha)\right) \tag{5}$$

Here, $i = F, \alpha$, $h_F$ and $h_\alpha$ represent the latent embeddings of the foreground and alpha channels within the transformer blocks, respectively. The symbol $\oplus$ denotes the concatenation operation. $W_q$, $W_k$, and $W_v$ are the matrices of query, key, and value embeddings, respectively. Finally, $Att(\cdot)$ refers to the softmax attention mechanism.

### 3.2.3 LDM Loss Function

We adopt annealed multi-resolution noise noises [16] to preserve low-frequency details in the depth and normal maps, as similar values will frequently appear in local geometric regions. This deviation proves to be more efficient than a single-scale noise schedule. We perturb the two geometry branches with the same timestep scheduler to decrease the difficulty when learning more modalities. The canonical standard learning objective we utilize is defined as follows:

$$\mathcal{L} = \mathbb{E}_{X,Tri,F,\alpha,\epsilon,t}[\epsilon_\theta\left(F_t; X, Tri, s_F\right) - \epsilon_t^F \|_2^2 + \epsilon_\theta\left(\alpha_t; X, Tri, s_\alpha\right) - \epsilon_t^\alpha \|_2^2] \tag{6}$$

Here, $\epsilon_t^F$ and $\epsilon_t^\alpha$ are two Gaussian noises independently sampled from annealed multi-scale noise sets for the foreground and alpha, respectively.


https://doi.org/10.52202/079017-2537

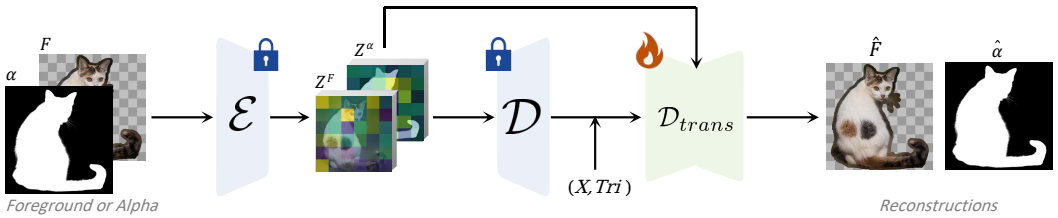



Figure 4: **Structure of Transparent Latent Decoder** (§3.3). Due to the non-negligible reconstruction loss introduced by the compression of the LDM's VAE, we employ a transparent latent decoder, which takes the output images of the LDM-based matting model and the corresponding latent as inputs, generating results that are more consistent with the details of the composite image.

### 3.3 Latent Transparency Decoder

The compression of the VAE [46] in LDM introduces a non-negligible reconstruction loss, indicating that $x \neq \mathcal{D}(\mathcal{E}(x))$. Previous discriminative tasks using the LDM [8, 12] have primarily focused on tasks such as segmentation and depth estimation. In comparison, matting is a task that places more emphasis on capturing fine details, making the errors introduced by the VAE more noticeable.

To address this challenge, we draw inspiration from LayerDiffusion [49] and propose a novel latent transparency decoder $\mathcal{D}_{trans}$. This decoder takes the output images of the LDM-based matting model 3.2 and the corresponding latent as inputs to generate results that are more consistent with the details of the composition image (refer to Fig 4). Mathematically, the expression is given by

$$(\hat{F}, \hat{\alpha}) = \mathcal{D}_{trans}(X, Tri, \mathcal{D}(\hat{z^F}), \mathcal{D}(\hat{z^{\alpha}}), \hat{z^F}, \hat{z^{\alpha}}) \tag{7}$$

where $\hat{z^F}$ and $\hat{z^{\alpha}}$ represent the foreground and alpha predictions in the latent space, respectively. Additionally, $\mathcal{D}$ represents the original VAE.

By incorporating this transparent latent decoder into our framework and training it with the matting loss, we aim to improve the fidelity of the predicted foreground and alpha outputs, ensuring that they capture the intricate details present in the composition image. This enhancement is particularly crucial for the task of matting, which relies heavily on preserving fine details and boundaries.

## 4 Experiment

### 4.1 Experimental Setup

#### 4.1.1 Datasets.

**Composition-1k dataset** [5] is a synthetic dataset consisting of 431 manually labeled foreground images for training and an additional 50 foreground images for evaluation. The training set is created by compositing each foreground image with 100 background images sourced from the COCO dataset [50]. This approach allows for the generation of a sufficient training set despite having a smaller number of unique foreground object images. Similarly, the test set is generated by synthesizing 50 test foreground images using 20 background images from VOC2012 [51], resulting in a total of 1000 test images.

**AIM-500 dataset** [6] is a benchmark for natural image matting that encompasses various object categories. It consists of 500 high-resolution real nature images, each with a minimum short side length of 1080 pixels. Unlike other natural matte datasets [52–54] that are often limited to specific classes of human and animal images, AIM-500 offers a more diverse range of objects. Evaluating the performance on the AIM benchmark helps assess the model's ability to generalize well to natural images rather than solely fitting the distribution of synthetic images. Therefore, the evaluations conducted on both the Composition-1k and AIM benchmarks complement each other, providing a comprehensive understanding of matte models in real-world scenarios.

#### 4.1.2 Implementation Details.

We employed Diffusers [55] with Stable Diffusion v2 [8] as the backbone for implementing our Drip. Text conditioning was disabled by providing empty text input. To accommodate the two additional conditions, the weights of the first layer of the UNet were copied three times as initialization. After training the LDM-based matting network, the obtained output and latent results were utilized as inputs for training the latent transparent decoder.

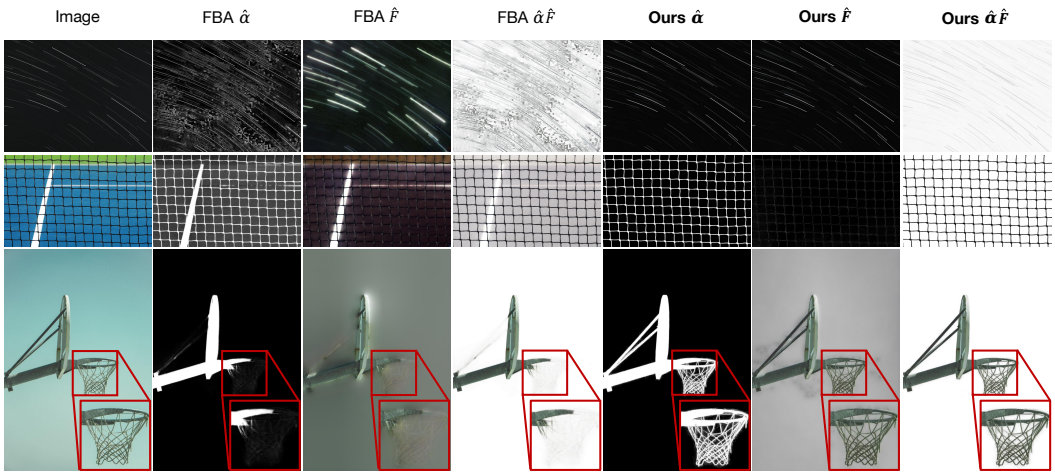



Figure 5: **Qualitative Result of Foreground.** The visual results compared with FBA [31] on AIM-500[6]. Please zoom in for the best view.

Table 1: Comparison of our Drip with State-of-the-Arts (SOTAs) on the synthetic dataset Composition-1k [5] and the natural dataset AIM-500 [6], focusing on four metrics for alpha prediction results.(§4.2).

| | Method | Publication | Composition-1k | | | | AIM-500 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | SAD | MSE | Grad | Conn | SAD | MSE | Grad | Conn |
| Alpha | DIM [5] | CVPR'17 | 50.4 | 14.0 | 31.0 | 50.8 | 49.3 | 14.7 | 29.3 | 47.1 |
| | IndexNet [56] | ICCV'19 | 45.8 | 13.0 | 25.9 | 43.7 | - | - | - | - |
| | FBA [31] | ArXiv'20 | 25.8 | 5.2 | 10.6 | 20.8 | - | - | - | - |
| | HATT [27] | CVPR'20 | 44.0 | 7.0 | 29.3 | 46.4 | 479.2 | 270.0 | 238.6 | 474.0 |
| | AIM [6] | IJCAI'21 | - | - | - | - | 43.9 | 16.1 | 33.1 | 43.2 |
| | GFM [32] | IJCV'22 | - | - | - | - | 52.7 | 21.3 | 46.1 | 52.9 |
| | MFormer [57] | CVPR'22 | 23.8 | 4.0 | 8.7 | 18.9 | - | - | - | - |
| | ViTMatte [7] | IF'23 | 21.5 | 3.3 | 7.2 | **16.2** | - | - | - | - |
| | DiffMat [43] | ArXiv'24 | 22.8 | 4.0 | 6.8 | 18.4 | - | - | - | - |
| | Ours | - | **20.8** | **3.1** | **6.8** | 17.8 | **17.3** | **1.5** | **5.4** | **14.7** |

To enhance the diversity of the dataset, we performed various data augmentation techniques during the training of the 2D image set. These included random horizontal flipping, cropping, and photometric distortion. Besides, in order to enhance the foreground, we followed LayerDiffusion [49] to fill the foreground image. Pixels with Alpha values equal to zero in the foreground have no impact on the appearance of the alpha-blended image when assigning any color. Nevertheless, since neural networks tend to produce high-frequency patterns surrounding image edges, avoiding unnecessary edges in the RGB channels prevents potential artifacts. To achieve this, we applied Gaussian blurring to regions of the foreground image where the Alpha value was strictly equal to zero.

During training, the DDPM noise scheduler [58] with 1000 diffusion steps was applied. At inference time, the DPM solver scheduler [59]was employed, and only 10 steps were sampled. The model was trained for 30,000 steps with a total batch size of 96, using an image size of 512 × 512 exclusively on the Composition-1k dataset [5]. The entire training procedure typically took approximately 2 days when executed on a cluster consisting of 4 Nvidia Tesla A100-80GB GPUs. For optimization, the Adam optimizer was used with a learning rate of $1 \cdot 10^{-5}$.

#### 4.1.3 Evaluation Metric.

We employ standard evaluation metrics to assess the quality of alpha predictions. Specifically, we report the Sum of Absolute Differences (SAD), Mean Square Error (MSE), Gradient loss (Grad), and Connectivity loss (Conn). A lower value for these metrics indicates a higher quality alpha matte.

Additionally, we follow the evaluation methodology of FBA matting [31] to evaluate the quality of foreground predictions. Since colors other than the foreground object region are not used, we only consider the region where the ground truth alpha, denoted as $\alpha_{gt}$, is located and use it as a weighting factor. We apply the SAD and MSE to evaluate the $\alpha_{gt}F$ predictions. A lower value for these metrics indicates a better quality foreground prediction.


https://doi.org/10.52202/079017-2537

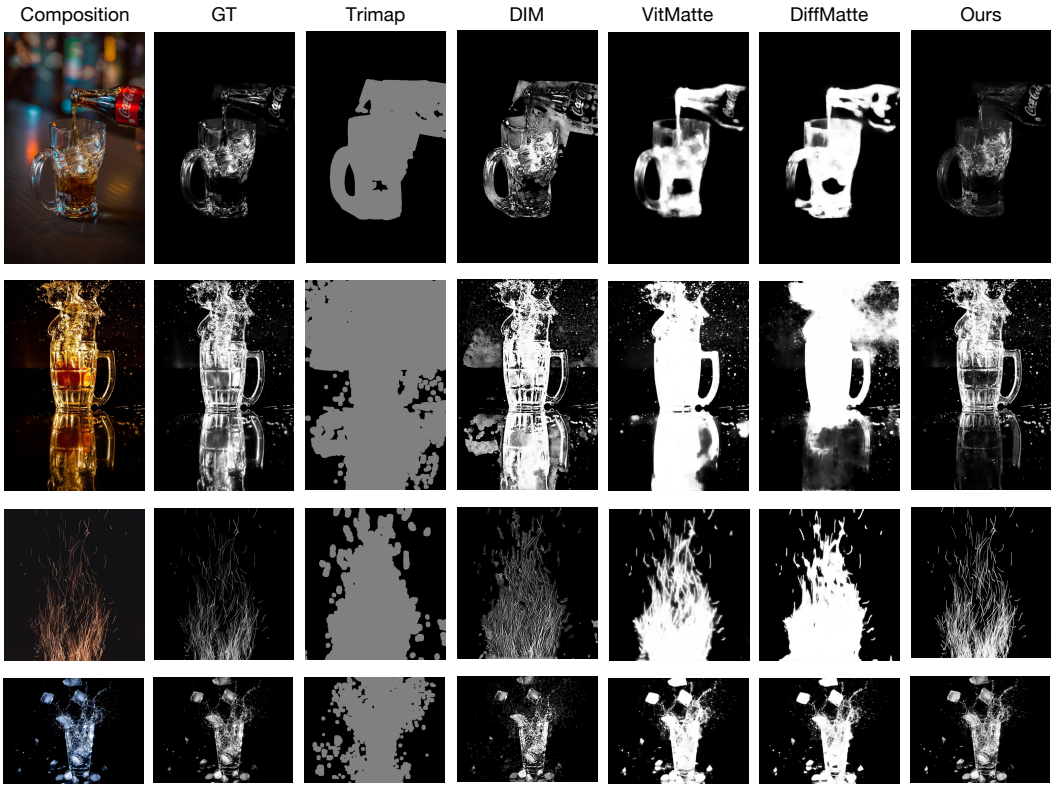



Figure 6: **Qualitative Result of Alpha.** The visual results compared with previous SOTA methods on AIM-500[6]. Please zoom in for the best view.

#### 4.1.4 Baselines.

For **alpha estimation**, we consider several state-of-the-art methods as baselines, including both trimap-based approaches and automated methods. Among the automated methods, HATT [27], AIM [6], and GFM [32] are included. On the other hand, the trimap-based methods consist of DIM [5], IndexNet [56], FBA [31], MFormer [57], and ViTMatte [7]. These trimap-based methods are all deep learning-based, with the backbone architecture gradually transitioning from convolutional networks to transformer-based networks. Furthermore, with the development of diffusion, the contemporary method DiffMat [43] is also based on diffusion. However, it is important to note that DiffMat [43] does not operate in the latent space, nor does it leverage the priori of the natural image distribution learned from diffusion.

For **foreground estimation**, we compare our proposed method with a limited set of baselines, as many matting algorithms primarily focus on alpha estimation rather than foreground estimation. In our comparison, we include Global-Matting [60] and KNN-Matting [61] as non-deep learning methods. We consider ContextAware-Matting [62] and FBAMatting [31] for deep learning-based methods. These approaches leverage deep neural networks to estimate the foreground and have demonstrated promising results in previous studies.

### 4.2 Quantitative Result & Qualitative Result

Table 2: Comparison with SOTAs(§4.2).

| | Method | | Publication | Composition-1k SAD | Composition-1k MSE |
|---|---|---|---|---|---|
| Foreground | Global | [60] | CVPR'11 | 220.39 | 36.3 |
| | KNN | [61] | TPAMI'13 | 281.9 | 36.3 |
| | CA | [62] | ICCV'19 | 61.72 | 3.24 |
| | LBS | [29] | CVPR'19 | 49.7 | 8.6 |
| | FBA | [31] | ArXiv'20 | 38.8 | 6.0 |
| | Ours | | - | **34.1** | **4.3** |

The results are shown in Table 1 for alpha. The results indicate that our method outperforms others by a large margin and achieves state-of-the-art (SOTA) performance. On the Composition-1k dataset, our method improves the SAD metric by 0.7 $(+3.3\%)$ and the MSE metric by 0.2 $(+6.1\%)$ compared to the ViTMatte [7] method. The performance improvements are even more noticeable on the AIM-500 dataset, where our method improves the SAD by 26.6 and the MSE by 14.6 compared to the AIM [6] approach. For the foreground metrics, as shown in Table 2, our method demonstrates significant improvements compared to the FBA-Matting [31]. Specifically, we improve the SAD metric by 4.7 $(+12.1\%)$ and the MSE metric by 1.7 $(+28.33\%)$.

https://doi.org/10.52202/079017-2537

Table 3: A set of ablative experiments about our proposed modules on the AIM-500 [6].(§4.3)

| Switcher | CDAttn | $\hat{\alpha}$ SAD | $\hat{\alpha}$ MSE | $\alpha\hat{F}$ SAD | $\alpha\hat{F}$ MSE | $\hat{\alpha}\hat{F}$ SAD | $\hat{\alpha}\hat{F}$ MSE |
|---|---|---|---|---|---|---|---|
| ✓ | | 18.1 | 1.7 | 21.1 | 3.8 | 23.7 | 5.2 |
| | ✓ | 17.8 | 1.5 | 21.3 | 4.1 | 26.7 | 5.9 |
| ✓ | ✓ | 17.3 | 1.5 | 20.6 | 3.7 | 23.3 | 4.9 |

(a) Joint prediction of foreground and alpha

| Decoder | $\hat{\alpha}$ SAD | $\hat{\alpha}$ MSE | $\alpha\hat{F}$ SAD | $\alpha\hat{F}$ MSE |
|---|---|---|---|---|
| | 21.3 | 2.1 | 25.3 | 4.8 |
| ✓ | 17.3 | 1.5 | 20.6 | 3.7 |

(b) Transparent latent decoder

Additionally, we visualize some qualitative results in comparison with other baselines. As demonstrated in Figure Fig. 5, our method produces foreground predictions that are more closely aligned with natural images, and the resulting RGBA outputs are more consistent with the original image in detail. As shown in Fig. 6, for natural datasets, our results are significantly closer to the true values, demonstrating the strong generalization ability of our method.

### 4.3 Ablation Study

We conduct ablation studies on AIM-500 [6] to investigate the contributions of the proposed joint prediction of foreground and alpha and the transparent latent decoder.

**Joint Prediction of Foreground and Alpha** We first investigate the effect of the proposed cross-domain attention mechanism on the joint foreground and alpha estimation(*cf.* §3.2). As shown in Table 3a, when the cross-domain attention module is removed, we observe a decrease in the prediction accuracies of both the foreground and alpha representations. Importantly, the accuracy of the final RGBA composition, obtained by multiplying the predicted foreground and alpha, decreases more significantly. This suggests that the information interaction between the two modalities, facilitated by cross-domain attention, increases the consistency of spatial information between the foreground and alpha representations. This finding verifies that cross-domain self-attention can effectively correlate the two representations, enabling them to benefit from each other's contextual cues mutually.

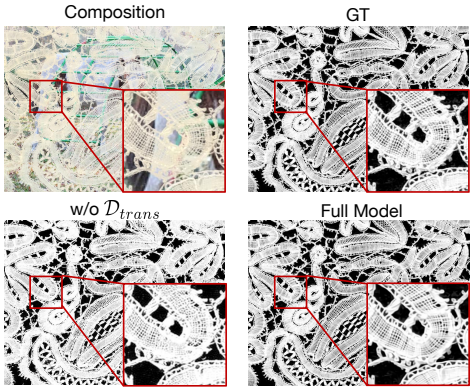



Figure 7: **Ablation Result.** When the transparent latent decoder is not used, the generated output exhibits significant differences in low-level details with the original input.

For the switcher, we also notice a clear reduction in performance, suggesting that giving additional embedding information rather than simply distinguishing between foreground and alpha with channel order is helpful in relation to the neural network's ability to distinguish and utilize information from the two modalities.

**Transparent Latent Decoder** Next, we ablate the effect of the $\mathcal{D}_{trans}$(*cf.* §3.3). As reported in Table 3b, using the original VAE decoder without $\mathcal{D}trans$ leads to a degradation in the prediction performance of both alpha and foreground. In addition, the qualitative results on composition-1k also clearly show that the low-level details correspond better after applying $\mathcal{D}_{trans}$, mitigating the errors induced by VAE compression.

## 5 Conclusion and Future Work

This work addresses two key challenges in image matting: high-quality foreground prediction and accurate semi-transparent object alpha estimation. To overcome these, we propose Drip method, which leverages data priors from large pre-trained generative models to jointly predict the foreground and alpha. Drip utilizes a switcher and a cross domain attention for consistent foreground-alpha generation and a latent transparency decoder to enhance fidelity. Extensive experiments demonstrate Drip achieves sota performance on Composition-1k and stronger generalization on other benchmarks.

While the Drip method demonstrates strong performance in image matting tasks, several key limitations warrant consideration: **i) Model Complexity and Deployment**. The incorporation of latent diffusion models (LDMs) substantially increases the architectural complexity of the approach. This added complexity may impact deployment and inference efficiency, particularly in real-time or


https://doi.org/10.52202/079017-2537

resource-constrained environments where computational overhead is critical. **ii) Inherited Biases from Generative Priors**. Drip's methodology relies on the extensive priors captured within pre-trained LDMs. Consequently, Drip inherits any biases present in the original generative model. These biases could adversely affect the method's performance on certain types of images or domains, such as those with highly complex lighting and shadows, or those containing numerous small texture details, thereby limiting its general applicability.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (U2336212) and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

## References

[1] Wenguan Wang, Yi Yang, and Yunhe Pan. Visual knowledge in the big model era: Retrospect and prospect. *Frontiers of Information Technology & Electronic Engineering*, 2024.

[2] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep image matting: A comprehensive survey. *arXiv preprint arXiv:2304.04672*, 2023.

[3] Zongxin Yang, Jiaxu Miao, Yunchao Wei, Wenguan Wang, Xiaohan Wang, and Yi Yang. Scalable video object segmentation with identification mechanism. *TPAMI*, 2024.

[4] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. *ACM Trans. on Graphics (TOG)*, 23(3):315–321, 2004.

[5] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*, 2021.

[7] Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pretrained plain vision transformers, 2023.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[10] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[11] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[12] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

[15] Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. Doraemongpt: Toward understanding dynamic scenes with large language models (exemplified as a video agent). In *ICML*, 2024.

[16] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. *arXiv preprint arXiv:2312.02145*, 2023.

[17] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[18] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[19] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[20] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[21] Thomas Germer, Tobias Uelwer, Stefan Conrad, and Stefan Harmeling. Fast multi-level foreground estimation. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2021.

[22] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):228–242, 2008.

[23] Jue Wang and Michael F Cohen. Optimized color sampling for robust matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[24] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):228–242, 2007.

[25] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[26] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *European Conference on Computer Vision (ECCV)*, 2016.

[27] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *AAAI Conference on Artificial Intelligence*, 2022.

[29] Jingwei Tang, Yagiz Aksoy, Cengiz Öztireli, Markus H. Gross, and Tunç Ozan Aydin. Learning-based sampling for natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3055–3063. Computer Vision Foundation / IEEE, 2019.

[30] Yagiz Aksoy, Tunç Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 228–236. IEEE Computer Society, 2017.

[31] Marco Forte and François Pitié. $f$, $b$, alpha matting. *arXiv preprint arXiv:2003.07711*, 2020.

[32] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision (IJCV)*, 130(2):246–266, 2022.

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

 https://doi.org/10.52202/079017-2537

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[35] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[36] Dewei Zhou, You Li, Fan Ma, Zongxin Yang, and Yi Yang. Migc++: Advanced multi-instance generation controller for image synthesis. *arXiv preprint arXiv:2407.02329*, 2024.

[37] Xiaolong Shen, Jianxin Ma, Chang Zhou, and Zongxin Yang. Controllable 3d face generation with conditional style code diffusion. In *AAAI*, volume 38, pages 4811–4819, 2024.

[38] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[39] D Zhou, Z Yang, and Y Yang. Pyramid diffusion models for low-light image enhancement. arxiv 2023. In *IJCAI*, 2023.

[40] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[41] Zechuan Zhang, Zongxin Yang, and Yi Yang. Sifu: Side-view conditioned implicit function for real-world usable clothed human reconstruction. In *CVPR*, pages 9936–9947, 2024.

[42] Yuanyou Xu, Zongxin Yang, and Yi Yang. Seeavatar: Photorealistic text-to-3d avatar generation with constrained geometry and appearance. *arXiv preprint arXiv:2312.08889*, 2023.

[43] Yangyang Xu, Shengfeng He, Wenqi Shao, Kwan-Yee K Wong, Yu Qiao, and Ping Luo. Diffusionmat: Alpha matting as sequential refinement learning. *arXiv preprint arXiv:2311.13535*, 2023.

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *The International Conference on Machine Learning (ICML)*, 2015.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (MICCAI)*, 2015.

[46] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[47] Marco Forte. Approximate fast foreground colour estimation. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2021.

[48] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

[49] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency, 2024.

[50] 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 (ECCV)*, 2014.

[51] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88:303–338, 2010.

[52] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xiansheng Hua. Boosting semantic human matting with coarse annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[53] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[55] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[56] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[57] Jiaojiao Li, Yihong Leng, Rui Song, Wei Liu, Yunsong Li, and Qian Du. Mformer: Taming masked transformer for unsupervised spectral reconstruction. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2023.

[58] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[59] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023.

[60] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[61] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35(9):2175–2188, 2013.

[62] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[63] 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 (WACV)*, 2022.

 https://doi.org/10.52202/079017-2537

# A Appendix

This appendix contains additional details for the Neurips 2024 submission, titled *"DRIP: Unleashing Diffusion Priors for Joint Foreground and Alpha Prediction in Image Matting"*. The appendix is organized as follows:

- §A.1 offers more implementation details of the multi-level annealed noise.
- §A.2 offers more implementation details of the augmentation to pad the foreground.
- §A.3 provides more information about the transparent latent decoder.
- §A.4 provides ablation results on the number of timesteps for the AIM [6] dataset.

## A.1 Annealed Multi-Resolution Noise

In order to make ldm handle the details better, we adopt the annealed multi-resolution noise used in [16, 14] .In crafting the standard multi-resolution noise, an array of Gaussian noise images is initially sampled to construct a pyramid with a gradient of resolutions, which are then merged using upscaling, weighted averaging, and normalization methods. Each level $i$ of this pyramid is assigned a weight following the formula $s^i$, where $s$ is a decimal fraction between 0 and 1, signifying the extent of the influence exerted by noise at reduced resolutions. To better align the resultant noise with the Gaussian distribution outlined in the foundational DDPM framework, the weights of the upper levels $i > 0$ are modified in accordance with a diffusion schedule. Specifically, at each time step $t$, the $i$-th level is endowed with a weight calculated as $(s^i/T)^t$, with $T$ representing the cumulative number of diffusion steps. As a result, the weight allocated to levels with diminished resolution is progressively reduced as the schedule nears its noise-free terminal point.

**Algorithm 1** annealed_pyramid_noise(x,timesteps,discount)

$b, c, w, h \leftarrow x.shape$ {Get the shape of the input tensor}
$w_{ori} \leftarrow w, h_{ori} \leftarrow h$ {Save the original dimensions}
$noise \leftarrow gen\_noise\_like(b * c * w_{ori} * h_{ori})$ {Create a noise tensor}
$i \leftarrow 0$
**repeat**
  $r \leftarrow rand() * 2 + 2$ {Generate a random scale factor}
  $w \leftarrow \max(1, \lfloor w_{ori}/r^i \rfloor)$
  $h \leftarrow \max(1, \lfloor h_{ori}/r^i \rfloor)$ {Compute the current feature map size}
  $temp\_noise \leftarrow gen\_noise\_like(b * c * w * h)$ {Generate a temporary noise tensor}
  **for** $j \leftarrow 0$ to $b * c * w_{ori} * h_{ori} - 1$ **do**
    $x_{idx} \leftarrow j\%w_{ori}$
    $y_{idx} \leftarrow \lfloor j/w_{ori} \rfloor$
    $new\_x \leftarrow \lfloor x_{idx} * (w_{ori}/w) \rfloor$
    $new\_y \leftarrow \lfloor y_{idx} * (h_{ori}/h) \rfloor$
    $new\_idx \leftarrow new\_y * w + new\_x$
    $noise[j] \leftarrow noise[j] + temp\_noise[new\_idx] * (timesteps/1000.0) * discount^i$
  **end for**
  $i \leftarrow i + 1$
**until** $i \geq 10$ or ($w \leq 1$ and $h \leq 1$) {If already reached minimum resolution, break out}
$result \leftarrow noise/noise.std$ {Normalize the noise tensor}
**return** *result*

## A.2 Padded Foreground

In order to enhance the foreground effect and avoid potential artifacts, we apply a fill technique [49] to the foreground image. Pixels in the foreground with an alpha value equal to zero, regardless of their color, do not affect the appearance of the final alpha-blended image. However, neural networks tend to produce high-frequency patterns at the edges of the image, so avoiding unwanted edges in the RGB channel helps prevent potential artifacts. To address this issue, we apply Gaussian blurring to regions of the foreground image where the alpha value is strictly equal to zero.

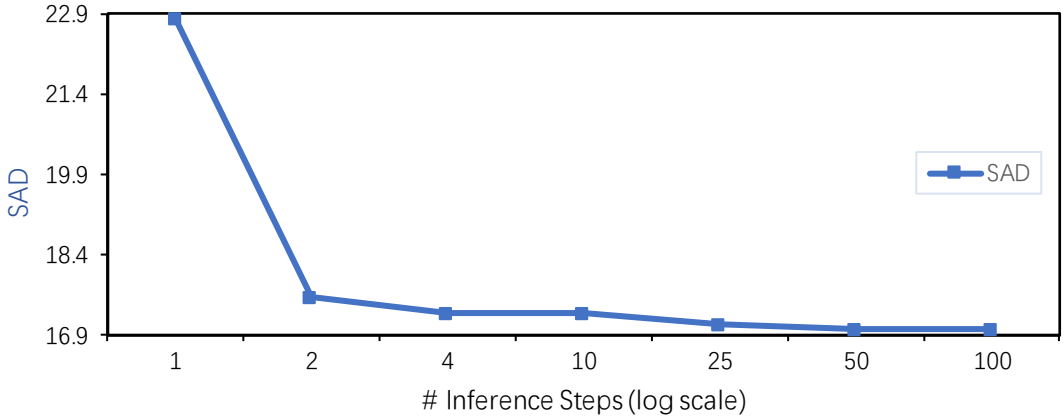



Figure 8: **Impact of Denoising Steps on Performance.** The performance of our method improves as the number of denoising steps increases, with diminishing marginal gains as the number of timesteps becomes larger.

**Algorithm 2** padded_fg(alpha, fg)

$h, w \leftarrow$ shape$(alpha)$ {Get the height and width of the input image}
mask $\leftarrow$ gen_mask$(alpha)$ {a mask where alpha = 0 are 1 and alpha > 0 are 0}
$i \leftarrow 0$
**while** $i < 64$ **do**
  filtered_image $\leftarrow$ gaussian_blur$(fg, (13, 13), 0)$ {apply gaussian blur to foreground}
  $j \leftarrow 0$
  **while** $j < h \times w$ **do**
    **if** mask$[j] == 1$ **then**
      $fg[j] \leftarrow$ filtered_image$[j]$ {update foreground if mask indicates transparent region}
    **end if**
    $j \leftarrow j + 1$
  **end while**
  $i \leftarrow i + 1$
**end while**
**return** $fg$ {Return the updated foreground image}

### A.3 Detail of Transparent Latent Decoder

The core component of our approach is the transparent latent decoder, which is implemented as a U-Net architecture. To train this module, we randomly sample the step number between 1 and 10 to get the output of latent matting network, which we then use as input. We utilize a loss function commonly used in matting algorithms to optimize the model. Specifically, we employ the combined matting loss, which includes separate components such as the $l_1$ loss [7, 63] , $l_2$ loss [63] , laplacian loss [62, 63] , and gradient penalty loss [7] for both alpha and foreground. The objective function becomes:

$$L_{mat} = L_{l_1}^{sp} + L_{l_2} + L_{lap} + L_{grad} \tag{8}$$

### A.4 Ablation Results on Timesteps

We conducted ablation studies on the number of diffusion timesteps used in our method on the AIM [6] dataset. As shown in Figure 8, increasing the number of timesteps generally improves the performance, though the gains diminish as the number of timesteps becomes larger.


https://doi.org/10.52202/079017-2537

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the key claims, contributions, and scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the conclusion section we discussed the limitation of our method.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In the implementation part of the experiment section we elaborate on the details needed for method reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?


https://doi.org/10.52202/079017-2537

Answer: [No]

Justification: Codes will be released after the paper is accepted. Data are available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In the implementation part of the experiment section we elaborate on the details needed for method reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: All experiments are conduct on fixed random seed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the implementation part of the experiment section we elaborate on the details needed for method reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in the paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.


https://doi.org/10.52202/079017-2537

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In the reference section we accurately cite the datasets, papers and codebases used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.


https://doi.org/10.52202/079017-2537