Supplementary Material for COIN-Matting: Confounder Intervention for Image Matting

Zhaohe Liao[1], Jiangtong Li[2], Jun Lan[3], Huijia Zhu[3], Weiqiang Wang[3], Li Niu[⋆1,4], and Liqing Zhang[*1]

[1] Shanghai Jiao Tong University
{zhaoheliao, ustcnewly, zhang-lq}@sjtu.edu.cn
[2] Tongji University
jiangtongli@tongji.edu.cn
[3] Ant Group
{yelan.lj, huijia.zhj, weiqiang.wwq}@antgroup.com
[4] miguo.ai

In this document, we provide additional background and details to support the main submission. In Appendix A, we provide additional background on the causal graph and causal inference. In Appendix B, we give more detailed derivations and explanations for the equations presented in the main submission. In Appendix C, we elaborate on the implementation details of our COIN matting framework. In Appendix D, we present additional qualitative comparisons to illustrate the effectiveness of our COIN matting framework. In Appendix E, we discuss the existence of the biases in real-world matting datasets. In Appendix F, we discuss the limitations of our framework and the directions for future work.

A Causal Graph and Intervention

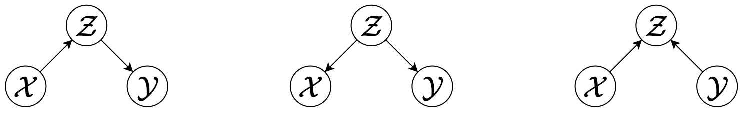
In this section, we introduce the fundamental concepts of causal graphs and causal interventions, which serve as the main tools used in our paper.

A.1 Causal Graph and Elemental Structures

Causal graph [3, 4] is a directed acyclic graph (DAG) whose edges indicate the relationship among variables. Formally, a causal graph is a DAG $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ denotes the variables, and the edge set $\mathcal{E}$ denotes the **causal relationships** among variables. An edge from node $\mathcal{X}$ to node $\mathcal{Y}$ (denoted as $e = \mathcal{X} \rightarrow \mathcal{Y}$) signifies that variable $\mathcal{X}$ is a direct cause of variable $\mathcal{Y}$. It is worth noting that such directed edges differ from those in Bayesian networks, as they possess strong causal semantics, whereas the latter may lack specific meaning and can even be antitemporal. For instance, while the graph $\mathcal{X} \rightarrow \mathcal{Y}$ is equivalent to $\mathcal{Y} \rightarrow \mathcal{X}$ in terms of probability graphs, their interpretations can be entirely different in causal graphs [1].

As shown in Fig. 1, there are three elemental structures in the causal graph.

[⋆] The corresponding authors.



(a) Chain structure. **(b)** Fork structure. **(c)** Collider structure.

Fig. 1: Three elemental structures in causal graph.

- **Chain structure** (Fig. 1a) is represented as $\mathcal{X} \rightarrow \mathcal{Y} \rightarrow \mathcal{Z}$, where $\mathcal{Z}$ acts as an intermediary variable between $\mathcal{X}$ and $\mathcal{Y}$. This structure implies that knowing the value of $\mathcal{Z}$ renders $\mathcal{X}$ and $\mathcal{Y}$ independent as $\mathcal{X}$ provides no additional information about $\mathcal{Y}$ in this context. Consequently, an intervention on $\mathcal{Z}$ would block the causal path from $\mathcal{X}$ to $\mathcal{Y}$.
- **Fork structure** (Fig. 1b) is represented as $\mathcal{X} \leftarrow \mathcal{Z} \rightarrow \mathcal{Y}$, with $\mathcal{Z}$ serving as the common cause (*i.e.*, the **confounder**) of $\mathcal{X}$ and $\mathcal{Y}$. This configuration leads to a spurious correlation between $\mathcal{X}$ and $\mathcal{Y}$ due to their shared cause, despite the absence of a direct causal link between them. For instance, let $\mathcal{X}$ denote the sales of umbrellas and $\mathcal{Y}$ denote the number of car accidents and $\mathcal{Z}$ as the *rainfall*. From the causal perspective, there exists no direct causal relationship between $\mathcal{X}$ and $\mathcal{Y}$. However, the correlation between $\mathcal{X}$ and $\mathcal{Y}$ can be induced by $\mathcal{Z}$, as rainfall can cause both the increase in umbrella sales and the increase in car accidents. Moreover, an intervention on $\mathcal{Z}$ shall block the correlation between $\mathcal{X}$ and $\mathcal{Y}$ and make them independent. In the aforementioned example, if we consider only the days with (or without) rainfall, the correlation between $\mathcal{X}$ and $\mathcal{Y}$ shall be blocked and the sales of umbrellas and number of car accidents shall be independent.
- **Collider structure** (Fig. 1c) is denoted as $\mathcal{X} \rightarrow \mathcal{Z} \leftarrow \mathcal{Y}$, where $\mathcal{Z}$ is the common effect of $\mathcal{X}$ and $\mathcal{Y}$. In this structure, $\mathcal{X}$ is naturally independent of $\mathcal{Y}$. However, if conditioned on $\mathcal{Z}$, $\mathcal{X}$ and $\mathcal{Y}$ could be correlated. Therefore, for analyzing the causal effect from $\mathcal{X}$ to $\mathcal{Y}$, we should leave the common effect $\mathcal{Z}$ unconditioned.

These three elemental structures are the basic modules of a causal graph. Moreover, their aforementioned properties provide us with the basic tools for analyzing the causal effects among variables and for blocking the paths between variables. In conclusion, to block the causal effect from $\mathcal{X}$ to $\mathcal{Y}$, we should intervene on $\mathcal{Z}$ in both the chain structure and the fork structure, while doing nothing for the collider structure.

A.2 The Confounder and do-calculus

As described in Appendix A.1, the correlation between two variables $\mathcal{X}$ and $\mathcal{Y}$ does not necessarily imply a direct causal relationship between them. In example, $\mathcal{X} \leftarrow \mathcal{Z} \rightarrow \mathcal{Y}$ also indicate a correlation between $\mathcal{X}$ and $\mathcal{Y}$. A simple proof would involve noting the fact that:

$$\begin{aligned} P(\mathcal{Y}|\mathcal{X}) - P(\mathcal{Y}) &= \sum_{\boldsymbol{z}} P(\mathcal{Y}|\mathcal{X}, \boldsymbol{z}) P(\mathcal{X}|\boldsymbol{z}) - \sum_{\boldsymbol{z}} P(\mathcal{Y}|\boldsymbol{z}) P(\boldsymbol{z}) \\ &= \sum_{\boldsymbol{z}} P(\mathcal{Y}|\boldsymbol{z}) P(\mathcal{X}|\boldsymbol{z}) - \sum_{\boldsymbol{z}} P(\mathcal{Y}|\boldsymbol{z}) P(\boldsymbol{z}) \\ &= \sum_{\boldsymbol{z}} P(\mathcal{Y}|\boldsymbol{z}) [P(\mathcal{X}|\boldsymbol{z}) - P(\boldsymbol{z})]. \end{aligned} \tag{12}$$

Due to the direct causal relation between $\mathcal{X}$ and $\mathcal{Z}$, typically $P(\mathcal{X}|\boldsymbol{z}) \neq P(\boldsymbol{z})$. And that would lead to $P(\mathcal{Y}|\mathcal{X}) - P(\mathcal{Y}) \neq 0$ which indicates $\mathcal{A}$ and $\mathcal{X}$ are not independent. Such a common cause like $\mathcal{Z}$ is called "**confounder**" in causal inference, and usually induces spurious correlation between the variables it causes, as illustrated int the fork structure of Appendix A.1.

Since the correlation does not mean causation, our goal is to model the actual causal effect between variables. One approach is to use an ideal intervention: **do-calculus** [3,4]. The do-calculus in $P(\mathcal{Y}|do(\mathcal{X} = x))$ (abbreviated as $P(\mathcal{Y}|do(\mathcal{X})))$ represents that we actively assign value x to variable $\mathcal{X}$ without any immediate effect instead of passively observing it as in $P(\mathcal{Y}|X = x))$. As we have assigned the value to $\mathcal{X}$, $\mathcal{X}$ shall not be influenced by its parent. Therefore, the do-calculus intervenes in the causal effect of the parent of $\mathcal{X}$ in the causal graph, and such do-calculus-involved probability is also called **intervention query**. For example, in Fig. 1a, calculating $P(\mathcal{Y}|do(\mathcal{X}))$ representing we set the variable $\mathcal{X}$ to be x and ignore the influence from its parent (*i.e.*, $\mathcal{Z}$). That is, we are cutting off all edges ending at $\mathcal{X}$ while calculating $P(\mathcal{Y}|do(\mathcal{X}))$. Therefore, the probability $P(\mathcal{Y}|do(\mathcal{X}))$ shall represent the causal effect from $\mathcal{X}$ to $\mathcal{Y}$.

By definition, the calculation of $P(\mathcal{Y}|do(\mathcal{X}))$ is to physically intervene on $\mathcal{X}$ and observe the change of $\mathcal{Y}$. That is, we shall collect (the possibly counterfactual) data by forcing $\mathcal{X}$ to be x without causing any other effect and observe the probability distribution of $\mathcal{Y}$. For example, if $\mathcal{X}$ represents the *mutation of specific gene* and $\mathcal{Y}$ represents the *probability of having cancer*, calculating $P(\mathcal{Y}|do(\mathcal{X}))$ involves collecting data by force the gene to mutate or not for all collected samples without having other immediate effect and observe the probability of having cancer conditioned on the existence of gene. Such an operation is usually impractical. However, we are still capable of reformulating the calculation of do-calculus into observational probabilities as shown in Appendix A.3.

A.3 Backdoor Path and Backdoor Adjustment

A backdoor path from $\mathcal{X}$ to $\mathcal{Y}$ represents a path that contains both causal edges pointing to $\mathcal{X}$ and to $\mathcal{Y}$. It could introduce spurious correlations to $\mathcal{X}$ and $\mathcal{Y}$ without direct causal relations. To remove this spurious correlation, we use **backdoor adjustment** to calculate the actual causal effect $P(\mathcal{Y}|do(\mathcal{X}))$.

The backdoor adjustment formula claims that, if $\boldsymbol{\mathcal{Z}}$ is a set of variables that blocks all backdoor paths from $\mathcal{X}$ to $\mathcal{Y}$ and has no node in the descendant of $\mathcal{X}$, then we may reformulate the intervention query $P(\mathcal{Y}|do(\mathcal{X}))$ as

$$P(\mathcal{Y}|do(\mathcal{X})) = \sum_{\boldsymbol{z}} P(\mathcal{Y}|\mathcal{X}, \boldsymbol{z})P(\boldsymbol{z}). \tag{13}$$

Proof. We may use standard probabilistic rules to conclude that

$$P(\mathcal{Y}|do(\mathcal{X})) = \sum_{\boldsymbol{z}} P(\mathcal{Y}|do(\mathcal{X}), \boldsymbol{z})P(\boldsymbol{z}|do(\mathcal{X})). \tag{14}$$

For the former terms, since $\boldsymbol{\mathcal{Z}}$ blocks all backdoor paths, $\mathcal{X}$ and $\mathcal{Y}$ are independent given $\mathcal{Z}$ on the graph where the outgoing edges of $\mathcal{X}$ are cut. Therefore, all information from $\mathcal{X}$ to $\mathcal{Y}$ goes through the outgoing edges from $\mathcal{X}$. Since the do-calculus only affects the incoming edges of $\mathcal{X}$, whether the value of $\mathcal{X}$ is assigned by do-calculus or observed form data does not affect the distribution of $\mathcal{Y}$. Therefore, we conclude that

$$P(\mathcal{Y}|do(\mathcal{X}), \boldsymbol{z}) = P(\mathcal{Y}|\mathcal{X}, \boldsymbol{z}). \tag{15}$$

For the second term, as no node in $\boldsymbol{\mathcal{Z}}$ is a descendant of $\mathcal{X}$, we can conclude that the intervention on $\mathcal{X}$ shall not influence the distribution of $\boldsymbol{\mathcal{Z}}$. Formally,

$$P(\boldsymbol{z}|do(\mathcal{X})) = P(\boldsymbol{z}). \tag{16}$$

By combining these formulas, we have

$$P(\mathcal{Y}|do(\mathcal{X})) = \sum_{\boldsymbol{z}} P(\mathcal{Y}|\mathcal{X}, \boldsymbol{z})P(\boldsymbol{z}). \tag{17}$$

B Detailed Proofs for Equations

In this section, we give more detailed proofs and explanations for the equations in the main submission using the background detailed in Appendix A.

B.1 Proof of Equation (3) and (4)

In Equation (3) and (4), we simplify the intervention query $P(\mathcal{A}|do(\mathcal{I}, \mathcal{F}, \mathcal{B}))$ into observational probabilities. In the main submission, we formulated:

$$\begin{aligned}
P(\mathcal{A}|do(\mathcal{I}, \mathcal{F}, \mathcal{B})) &= \sum_{\boldsymbol{t}} \sum_{\boldsymbol{c}} P(\mathcal{A}|do(\mathcal{I}, \mathcal{F}, \mathcal{B}), \boldsymbol{t}, \boldsymbol{c})P(\boldsymbol{c}, \boldsymbol{t}|do(\mathcal{I}, \mathcal{F}, \mathcal{B})) \\
&= \sum_{\boldsymbol{t}} \sum_{\boldsymbol{c}} P(\mathcal{A}|\boldsymbol{i}, \boldsymbol{f}, \boldsymbol{b}, \boldsymbol{t}, \boldsymbol{c})P(\boldsymbol{c}, \boldsymbol{t}) \\
&= \sum_{\boldsymbol{t}} \sum_{\boldsymbol{c}} P(\mathcal{A}|\boldsymbol{f}, \boldsymbol{t}, \boldsymbol{c})P(\boldsymbol{t}, \boldsymbol{c}) = \mathbb{E}_{\boldsymbol{t},\boldsymbol{c}}[P(\mathcal{A}|\boldsymbol{f}, \boldsymbol{t}, \boldsymbol{c})]
\end{aligned} \tag{18}$$

Since $\mathcal{C}$ and $\mathcal{T}$ blocks all backdoor paths $\mathcal{A} \leftarrow \mathcal{F} \leftarrow \mathcal{C} \rightarrow \mathcal{B}$, $\mathcal{B} \leftarrow \mathcal{I} \leftarrow \mathcal{C} \rightarrow \mathcal{F} \rightarrow \mathcal{A}$, and $\mathcal{A} \leftarrow \mathcal{F} \leftarrow \mathcal{T} \rightarrow \mathcal{I} \rightarrow \mathcal{B}$, we may apply the backdoor adjustment formula, resulting in the first two steps. The former step corresponds to Eq. (14) and the latter step corresponds to Eq. (17). The third step involves removing redundant variables in the condition set using the condition independence property of the causal graph in Figure 3(b) of the main submission. Note all input edges of $\mathcal{I}, \mathcal{F}$ and $\mathcal{B}$ are removed since we have intervened their value, and their direct causes have nothing to do with this. Formally, we derive the following formulas:

$$\begin{aligned}
&\sum_{\boldsymbol{t}}\sum_{\boldsymbol{c}} P(\mathcal{A}|\boldsymbol{i},\boldsymbol{f},\boldsymbol{b},\boldsymbol{t},\boldsymbol{c}) \\
=&\sum_{\boldsymbol{t}}\sum_{\boldsymbol{c}} P(\mathcal{A}|\boldsymbol{f},\boldsymbol{i},\boldsymbol{t},\boldsymbol{c})P(\boldsymbol{t},\boldsymbol{c}) \quad \cdots\cdots \quad \mathcal{A} \perp\!\!\!\perp \mathcal{B}|\mathcal{I},\mathcal{F},t,c \\
=&\sum_{\boldsymbol{t}}\sum_{\boldsymbol{c}} P(\mathcal{A}|\boldsymbol{f},\boldsymbol{t},\boldsymbol{c})P(\boldsymbol{t},\boldsymbol{c}) \quad \cdots\cdots \quad \mathcal{A} \perp\!\!\!\perp \mathcal{B}|\mathcal{F},t,c.
\end{aligned} \tag{19}$$

The last simplification is the definition of expectation.

B.2 Proof of Equation (5)

In Equation (5) of the main submission, we formulated:

$$\begin{aligned}
P(\mathcal{F}|do(\mathcal{I},\mathcal{B})) &= \sum_{\boldsymbol{t}}\sum_{\boldsymbol{c}} P(\mathcal{F}|\boldsymbol{i},\boldsymbol{b},\boldsymbol{t},\boldsymbol{c})P(\boldsymbol{c},\boldsymbol{t}) \\
&= \sum_{\boldsymbol{t}}\sum_{\boldsymbol{c}} P(\mathcal{F}|\boldsymbol{i},\boldsymbol{t},\boldsymbol{c})P(\boldsymbol{c},\boldsymbol{t}) = \mathbb{E}_{\boldsymbol{c},\boldsymbol{t}}[P(\mathcal{F}|\boldsymbol{i},\boldsymbol{t},\boldsymbol{c})].
\end{aligned} \tag{20}$$

As $\mathcal{C}$ and $\mathcal{T}$ blocks all backdoor paths $\mathcal{F} \leftarrow \mathcal{C} \rightarrow \mathcal{B}$, $\mathcal{B} \leftarrow \mathcal{I} \leftarrow \mathcal{C} \rightarrow \mathcal{F}$, and $\mathcal{F} \leftarrow \mathcal{T} \rightarrow \mathcal{I} \rightarrow \mathcal{B}$, we apply the backdoor adjustment formula and result in the first step. The second step comes from the fact that $\mathcal{B}$ and $\mathcal{A}$ are independent given $\mathcal{F}$ in Figure 3(b) of the main submission. The last step is the definition of expectation.

B.3 Proof of Equation (7)

Equation (7) in the main submission is formulated as

$$P(\mathcal{A}|do(\mathcal{I},\mathcal{F},\mathcal{B})) = \mathbb{E}_{\boldsymbol{z}}P(\mathcal{A}|\boldsymbol{f},\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z}}\ \sigma(g(\boldsymbol{f},\boldsymbol{z})) \approx \sigma(g(\boldsymbol{f},\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}])), \tag{21}$$

The first step comes from Equation (6) of the main submission. The second step denotes the implementation of producing alpha matte of the matting framework. In detail, it denotes that for every scale, we concatenate the foreground feature $\boldsymbol{f}$ and confounder representation $\boldsymbol{z}$ to produce the alpha representation via a linear transformation $g(\cdot)$, then a sigmoid function $\sigma(\cdot)$ is applied to normalize the output. The third step involves the Normalized Weighted Geometric Mean (NWGM) approach [5,6]. As the sigmoid function can be viewed as a degradation of the softmax function in the binary classification, we prove the more generalized softmax form in the following. Consider $\mathbb{E}_{\boldsymbol{z}}[\text{softmax}(g(f,z))]$ By applying the Weighted Geometric Mean (WGM) [5], we approximate the expectation as

$$\mathbb{E}_{\boldsymbol{z}}[\text{softmax}(g(f,z))] \approx \text{WGM}[\text{softmax}(g(f,z))]. \tag{22}$$

Then, we move the expectation to the feature level. Given $\text{Softmax}(g(f,z)) \propto \exp[g(f,z)]$, we have

$$\begin{aligned}\text{WGM}[\text{softmax}(g(f,z))] &= \prod_{\boldsymbol{z}} \exp[g(f,z)]^{P(z)} \\ &= \exp\left[\sum_{\boldsymbol{z}} P(z)g(f,z)\right] = \exp\left[\mathbb{E}_{\boldsymbol{z}}[g(f,z)]\right].\end{aligned} \tag{23}$$

Therefore, to normalize the distribution, we apply the softmax function to the expectation, resulting in the Normalized Weighted Geometric Mean (NWGM) approach as

$$\text{NWGM}[\text{softmax}(g(f,z))] = \frac{\exp[\mathbb{E}_{\boldsymbol{z}}[g(f,z)]]}{\sum_{\boldsymbol{z}} \exp[\mathbb{E}_{\boldsymbol{z}}[g(f,z)]]} = \text{softmax}(\mathbb{E}_{\boldsymbol{z}}[g(f,z)]). \tag{24}$$

Further, as $g(\cdot)$ is a linear transformation, we can move the expectation into it, resulting in the feature level expectation as

$$\mathbb{E}_{\boldsymbol{z}}[g(f,z)] = g(f, \mathbb{E}_{\boldsymbol{z}}[z]). \tag{25}$$

By combining Eqs. (22) to (25), and degrading the softmax function in the binary classification scene, we conclude the third step of Equation (7) in the main submission.

B.4 Proof of Equation (8)

Equation (8) in the main submission is formulated as

$$P(\mathcal{F}|do(\mathcal{I},\mathcal{B})) = \mathbb{E}_{\boldsymbol{z}} P(\mathcal{F}|\boldsymbol{i},\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z}}\ g(\boldsymbol{i},\boldsymbol{z}) = g(\boldsymbol{i}, \mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}]). \tag{26}$$

The first step comes from Equation (6) of the main submission, and the second step implements the feature transformation from the (multi-scale) image feature to the foreground feature via a linear transformation. The third step performs the move-in of an expectation, which can be explained as $g(\cdot)$ is a linear transformation.

C Implementation Details

In this section, we provide more implementation details about our COIN matting framework. For the implementation of the confounder spaces, the dimension of space representation for each level corresponds to the number of channels in the feature map extracted by the feature extractor. To produce the alpha matte, we

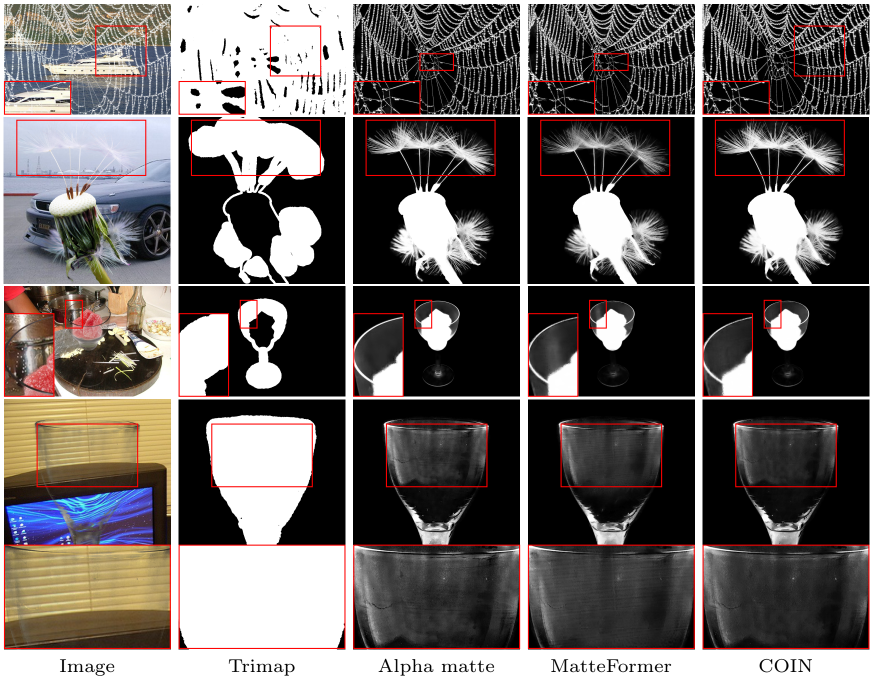


Fig. 2: The visualized comparison of applying our COIN framework on MatteFormer [2]. Images are best viewed when **zoomed in**.

follow [2,7] which applies a progressive refinement strategy. Such a strategy uses the multiscale alpha mattes to generate the final alpha matte in both training and testing and is also capable for the multiscale feature space in our framework. The training procedures are consistent with the baselines. Typically, the losses include alpha loss, compositional loss, and Laplacian loss. Further, we supervise the attention scores, which act as the probability in calculating the expectations, with the ground truth contrast level and transparency level. To address the continuity semantic of the contrast level and transparency level, we smooth the label with $\mathcal{N}(0, 1.2)$ Gaussian kernel for a more smoothed representation. During training, we employ the same data preprocessing, learning rate schedule, and optimizer as the baselines.

.

D More Qualitative Results

In Fig. 2, we give more qualitative comparisons to illustrate the effectiveness of our COIN matting framework. The first two rows show two examples of contrast bias. The areas located by the red boxes have low contrast, and the baseline predicts lower even zero alpha values on it. In contrast, after applying our COIN

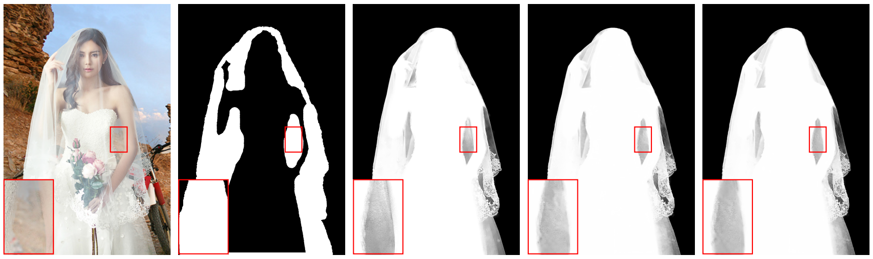
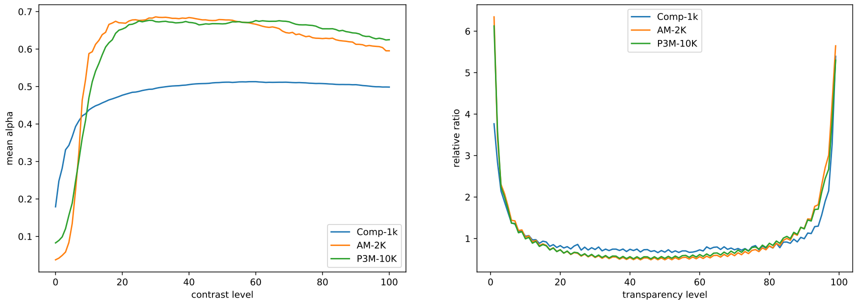


Image Trimap Alpha matte MatteFormer COIN

Fig. 3: An example of the failure case.



(a) Contrast bias. **(b)** Transparency bias.

Fig. 4: The contrast bias and transparency bias in datasets.

framework, such bias is relieved and the alpha matte is more consistent with the ground truth. The lower two rows show two examples of transparency bias. The areas located by the red boxes have medium transparency, and the texture of the background (*i.e.*, the reflection light in the third row and the texture of blinds in the fourth row) mistakenly appears in the alpha matte of the baseline. By applying out COIN framework, the transparency bias is reduced by intervening the confounder variable, resulting in less background texture in the alpha matte.

E Biases on the Real-World Matting Datasets

To address such concern on whether the proposed biases can generalize to real-world matting scenarios, we firstly verified the existence of the transparency bias and contrast bias in the training datasets, as shown in Figure 4. The contrast and transparency bias in the real-world dataset (*i.e.*, AM-2K and P3M-10K) are both similar to those in the synthetic dataset (*i.e.*, Compositional-1K). This suggests that the contrast bias and transparency bias are generally exist, no matter in real-world dataset or in synthetic dataset. Therefore, our framework is able to generalize to real-world matting datasets and effectively reduce the biases in them. To further demonstrate that the improvement of our framework

Model	AM-2K		P3M-10K		RWP-636	
	SAD ↓	MSE ↓	SAD ↓	MSE ↓	SAD ↓	MSE ↓
MatteFormer	5.72	4.65	4.73	9.87	21.37	51.85
+ COIN	5.11 $^{-0.61}$	4.22 $^{-0.43}$	4.18 $^{-0.55}$	8.96$^{-0.91}$	19.21 $^{-2.16}$	46.31 $^{-5.54}$

Table 1: The improvement of COIN framework on real-world matting datasets. Metrics are computed on the unkown area of trimap.

can generalize to real-world scenarios, we conduct experiments on multiple real-world matting datasets, including AM-2K, P3M-10K and RealWorldPortrait-636 (RWP-636). For the RWP-636 dataset, since it only contains test set, the models are trained on Composition-1K dataset and tested on RWP-636 dataset to evaluate the generalization ability of models from synthetic scene to real-world scene. For the AM-2K and P3M-10K dataset, we estimate the foreground and background based on image and alpha with Closed-Form Matting to compute the contrast and compositional loss during initialization and training. The results are listed in Table 1. As shown in Table 1, we could observe that our COIN framework still significantly improves the performance of the matting backbone on real-world matting datasets, which further proves the generalization ability of our framework to real-world scenarios. In detail, our framework achieves improvements from 9.21% to 11.62% across the real-world datasets in various metrics. Such improvements indicates that, since there still exist similar contrast and transparency bias in the real-world datasets, the COIN framework can still effectively reduce the bias in them.

F Limitation and Future Work

Our COIN framework is a general framework for addressing the confounding bias in the matting task, and can efficiently reduce the observed contrast bias and transparency bias. However, due to the full disentangling of all confounder variables being impractical, other biases may still exist, causing the background still correlated to the predicted alpha matte via other biases. Therefore, exploring more confounder variables or using other causal tools to reduce the biases may be a future work.

Moreover, while implementing the intervention, we applied several approximations (*i.e.*, in Equation (7) and the condition of the feature-level expectations). These approximations prevent our framework from completely eliminating the contrast biases and transparency biases, but can only significantly reduce them. An example is shown in Fig. 3. The area highlighted with the red box is semi-transparent. Although the most apparent background texture (*i.e.*, the shape of the right part of the stone and its boundary line with the sky) disappears, there still slightly exists the texture of the left part of the stone.

References

1. Koller, D., Friedman, N.: Probabilistic Graphical Models - Principles and Techniques. MIT Press (2009)
2. Park, G., Son, S., Yoo, J., Kim, S., Kwak, N.: Matteformer: Transformer-based image matting via prior-tokens. In: CVPR (2022)
3. Pearl, J., Glymour, M., Jewell, N.P.: Causal inference in statistics: A primer (2016)
4. Rubin, D.B.: Essential concepts of causal inference: a remarkable history and an intriguing future. Biostatistics & Epidemiology (2019)
5. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. (2014)
6. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML (2015)
7. Yu, Q., Zhang, J., Zhang, H., Wang, Y., Lin, Z., Xu, N., Bai, Y., Yuille, A.L.: Mask guided matting via progressive refinement network. In: CVPR (2021)