

TAME-RD: Text Assisted Replication of Image Multi-Adjustments for Reverse Designing

| | | | |
|---|---|--|---|
| Pooja Guhan* University of Maryland pguhan@umd.edu | Uttaran Bhattacharya Adobe Inc. ubhattac@adobe.com | Somdeb Sarkhel Adobe Inc. sarkhel@adobe.com | Vahid Azizi Adobe Inc. vazizi@adobe.com |
| Xiang Chen Adobe Inc. xiangche@adobe.com | Saayan Mitra Adobe Inc. smitra@adobe.com | Aniket Bera Purdue University aniketbera@purdue.edu | Dinesh Manocha University of Maryland dmanocha@umd.edu |

Abstract

Given a source and its edited version performed based on human instructions in natural language, how do we extract the underlying edit operations, to automatically replicate similar edits on other images? This is the problem of reverse designing, and we present **TAME-RD**, a model to solve this problem. **TAME-RD** automatically learns from the complex interplay of image editing operations and the natural language instructions to learn fully specified edit operations. It predicts both the underlying image edit operations as discrete categories and their corresponding parameter values in the continuous space. We accomplish this by mapping together the contextual information from the natural language text and the structural differences between the corresponding source and edited images using the concept of pre-post effect. We demonstrate the efficiency of our network through quantitative evaluations on multiple datasets. We observe improvements of 6–10% on various accuracy metrics and 1.01X–4X on the RMSE score and the concordance correlation coefficient for the corresponding parameter values on the benchmark GIER dataset. We also introduce **I-MAD**, a new two-part dataset: **I-MAD-Dense**, a collection of approximately 100K source and edited images, together with automatically generated text instructions and annotated edit operations, and **I-MAD-Pro**, consisting of about 1.6K source and edited images, together with text instructions and annotated edit operations provided by professional editors. On our dataset, we observe absolute improvements of 1–10% on the accuracy metrics and 1.14X–5X on the RMSE score. [Project Page]

1 Introduction

The prevalence of digital media in education (Haleem et al., 2022), healthcare (Ventola, 2014), business and entertainment (Fitzgerald et al., 2022) comes with the need for large-scale image

designing. It involves multiple image adjustment operations, ranging from image-level filtering (Mittal et al., 2021; Zhou et al., 2023; Jing et al., 2022; Krawczyk et al., 2007) to pixel-level manipulations (Steininger et al., 2023). This has opened up new research avenues focusing on making multi-adjustment image designing an intuitive and comfortable experience for editors. From the editors’ perspective, image designing can be broadly categorized into forward and reverse designing. Forward designing refers to editors planning and applying multi-adjustment edit operations on source images based on their specific end goals. Conventional image editing tools (Systems, 2002) are tailored for forward designing. Reverse design, by contrast, refers to extracting multi-adjustment edit operations applied between given pairs of source and edited images. The operations obtained can then be reapplied on other images in downstream pipelines (Xue et al., 2020; Rossler et al., 2019).

However, the realm of reverse designing remains relatively unexplored compared to forward designing. Complex many-to-many relationships in mappings from source to edited images contribute to this gap, making it challenging to establish optimization criteria solely based on factors like the total number of edit operations or specific parameter value ranges. Additionally, the lack of standardization in forward design practices among image editors further complicates this challenge. Reverse designing holds significant potential for generating comprehensive editing histories, facilitating the replication and reuse of large volumes of image assets. The availability of editing history proves advantageous across various applications including image editing revision control (Rinaldi et al., 2023), automatic tutorial generation (Grabler et al., 2009), editing visualization (Feng et al., 2023), quality control (Jiang et al., 2021), and forensic analysis (Rossler et al., 2019). However, relying only on the pairs of source and edited images for

*Work partly done as an intern at Adobe Inc.

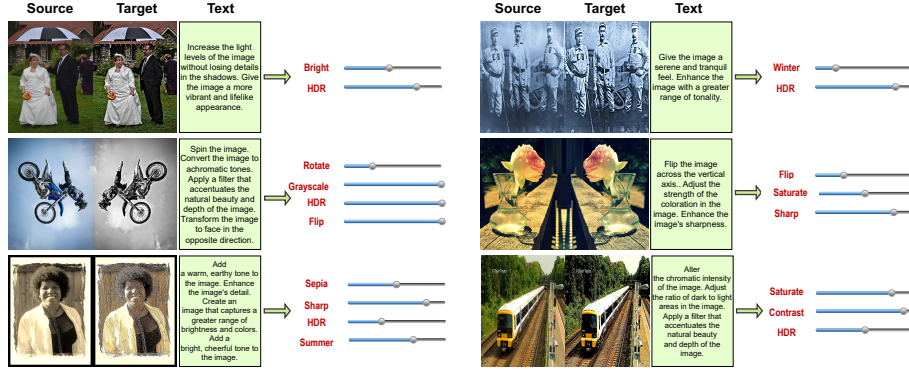


Figure 1: **Image Multi-Adjustments.** Given a source image, its edited version, and contextual information in natural language, we estimate the set of underlying edit operations and the corresponding parameter values that map the source image to the edited image. We show such edit operations estimated with our approach on six image pairs. We also present a new dataset called **I-MAD** to enable further research.

reverse designing may be insufficient. This limitation arises from the need to interpret the creative decisions and goals of the editor in the reverse design process. To address this, we propose a more promising approach that involves incorporating an additional modality to provide contextual information. We integrate the editors’ intent expressed through natural language text. This might elucidate the forward design mechanism they employed or even discern specific reasons or details associated with the editing process that transformed the source image into the edited image. Documenting machine-level micro-instructions that can be directly fed to image editing softwares is not scalable, particularly when dealing with a large number of images. Consequently, any textual information available for reverse designing purposes will primarily be high-level, vague, and in natural language. Our multimodal approach seamlessly integrates image pairs with text-based contextual cues, enabling a more nuanced understanding of the editing process that led to obtaining the edited image from the source image. The inclusion of a text component proves invaluable in complementing visual information, particularly in instances of noise or insufficient clarity. This holistic strategy endeavors to address the gap in reverse design research by prioritizing the interplay between visual and language-based sources of information.

Main Contributions. We introduce a multimodal learning method tailored for reverse design in image editing. The novel components include:

1. **Image Multi-Adjustment Prediction.** We present **TAME-RD**, a multimodal multitask learning model designed to predict a sequence of image editing operations along with their

parameter values. By blending structural and semantic correlations in the pixel space through our use of the *pre-post effect*, and integrating it with the accompanying textual information, **TAME-RD** can predict the underlying micro-instructions for the editing process. This sets the foundation for a paradigm shift in our capability to learn editing meta-data from image pairs.

2. **Image Multi-Adjustment Dataset.** We introduce a two-part dataset named **I-MAD** to advance research in reverse design and related tasks. It consists of a set of 100K images edited through an automated pipeline (**I-MAD-Dense**), and a set of 1.6K image pairs edited by professional editors (**I-MAD-Pro**). Different from existing image editing datasets such as GIER (Shi et al., 2020), our dataset contains rich annotations of edit operations and their parameter values from professional editors.
3. **Controllable Reverse Design.** To the best of our knowledge, we present the first controllable pipeline for reverse design, *i.e.*, estimating the *specific edit parameter values* in addition to the edit operation names. Our approach equips editors with a complete set of machine-level operations shaping edited images, offering unparalleled customization. This stands in stark contrast to the current state of the art, which, at best, can only estimate image edit operations.

We show quantitative evaluations on two datasets, GIER (Shi et al., 2020), and **I-MAD**, on the joint tasks of (a) multi-label classification and multi-valued regression, (b) multi-class classification and single-valued regression, and (c) multi-class classification and multi-valued regression. On the GIER

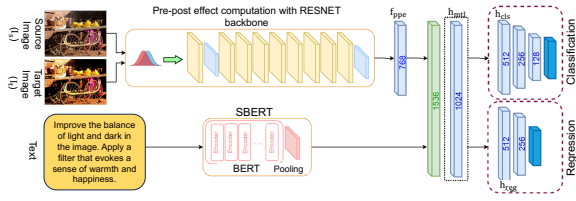


Figure 2: **TAME-RD**: I_s (source image), I_t (edited image), and c (text instructions as a sequence of words, optional) are the inputs to our network. We extract features f_{ppe} that capture the structural differences between the two images I_s and I_t based on the pre-post effect (Sec. 3.2) and f_c corresponding to the semantically encoded features of c (Sec. 3.3). Our fused latent features h_{mtl} incorporate both the structural and the contextual information to enable the prediction of edit operations that are both numerically accurate and plausible to human users. Our predictions consist of categorical edits paired with the corresponding parameter values. We achieve this by separating h_{mtl} into h_{cls} and h_{reg} , which we use for our classification and regression branches, respectively.

dataset, compared to the closest available baselines, we report absolute improvements of 1 – 10% on the relevant evaluation metrics of accuracy, precision, and F1 score. We also report RMSE and concordance correlation coefficient (CCC) improvements of 1.01X – 4X compared to the baselines. On our proposed dataset **I-MAD**, compared to the baselines, we report absolute improvements of 1 – 10% on the same evaluation metrics, and 1.1X – 5X improvement in RMSE and CCC.

2 Related Work

We provide a brief overview of recent works on understanding the relationship between two images, and datasets containing comparisons between two images.

Forward Design. Current image editing approaches, including those using language-based instructions (Jiang et al., 2021; Shi et al., 2021; Fu et al., 2022), excel at forward design by automatically applying specific image edits. However, they lack the key capability for reverse design, which is extracting edit operations given source and edited images. Style transfer (Jing et al., 2019) and image-to-image translation (Kim et al., 2022) methods directly apply changes from one image onto another. However, they do not provide control over the multi-adjustments that underlie the entire transformation as required for reverse design.

Relating Two Images. Reverse design requires an understanding of how a source and an edited image are related. From this perspective, we note recent methods relating two images. Methods such as (Yan et al., 2021; Guo et al., 2022) take in an

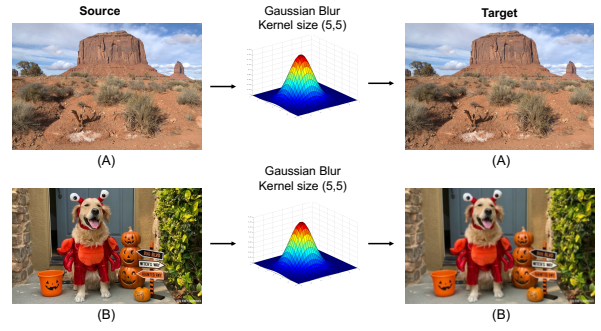


Figure 3: Source (A) and (B) undergo the exact same operation to give edited (A) and (B), respectively. However, the blur is more apparent in edited (B) than in edited (A) because of more foreground elements closer to the camera.

image pair and output a sentence describing their similarities and differences. Other methods focus on computing bounding boxes for image elements such as objects that were altered or removed between a source and an edited image (Sachdeva and Zisserman, 2023). These methods focus entirely on the low-level image structures and do not consider any higher-level semantic relationships. Closer to the reverse design pipeline, Tan et al. (2019) predict categorical edit operations performed between image pairs but do not estimate the corresponding parameter values that are necessary to close the loop for reverse design. Jiang et al. (2021) leverages this work (Tan et al., 2019) to check the quality of its forward design-based generated output. There don’t seem to be any recent attempts close enough to explore reverse design.

Paired Image Datasets. Image datasets are available aplenty, but only a handful of them (Jhamtani and Berg-Kirkpatrick, 2018; Suhr et al., 2018) include image pairs semantically connected by human-readable descriptions. There are also datasets for language-guided image editing (Fu et al., 2022), performing style transfer (Fu et al., 2020), adding, altering, and removing objects (Sun et al., 2021; Park et al., 2022), and storing edit operations performed between image pairs (Shi et al., 2020). However, these datasets lack tool-level details on how source images were edited, and are, therefore, not sufficient to train supervised reverse design pipelines.

3 TAME-RD: Our Approach

We present our algorithm, **TAME-RD**, to detect image multi-adjustments and close the loop for reverse image design. We formally state the problem in Sec. 3.1, and explain all the components of our

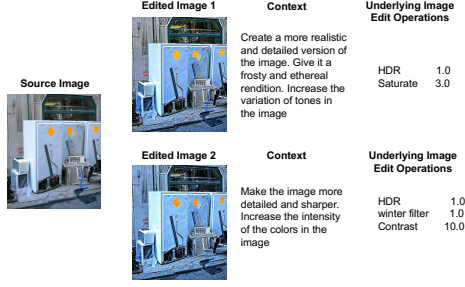


Figure 4: For edited images that look similar to each other, there can be ambiguity in the edit operations without any additional context such as from textual instructions.

approach in Secs. 3.2-3.4. Fig. 2 gives an overview of the proposed approach.

3.1 Problem Formulation

We consider two images, source I_s and edited I_t . They are related as follows:

$$I_t = \mathbf{E}(I_s; \theta) \quad (1)$$

where \mathbf{E} represents the set of edit operations done on I_s and $\theta \in \mathbb{R}^d$ represents the flattened vector consisting of all the corresponding parameter values. Our goal in this work is to identify \mathbf{E} and θ given I_s and I_t . This problem is non-trivial because of two primary reasons:

1. When we consider two or more edit operations, a one-to-one mapping \mathbf{E} between any given image pair I_s and I_t generally does not exist. Different combinations of individual edit operations performed with different parameter values on I_s can result in similar I_t (Fig. 4), which can only be resolved with additional context. Conversely, the same set of edit operations performed with the same parameter values on two different images I_{s1} and I_{s2} can result in vastly different-looking edited images I_{t1} and I_{t2} (refer Fig. 3).
2. Given I_s and I_t , we can compute the structural differences between the two. However, that, by itself, does not reveal any semantic relationship between the two, which can make a particular solution pair (\mathbf{E}^*, θ^*) more plausible than the rest to human users (refer Fig. 5).

To approach these issues, we first consider a finite set of N categorical operations $\{E_1, \dots, E_N\}$ to represent the function space of \mathbf{E} . We note that images in practice are constrained to lie within a finite range in the pixel space (most commonly

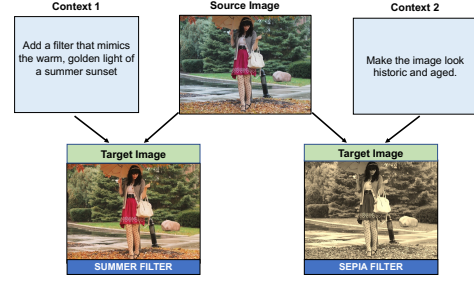


Figure 5: Given different contexts (as textual instructions in our case), the plausible edit operations on the same source image can change significantly.

between the values 0 and 255 for each pixel); therefore, the parameter values θ also lie in a finite range by design. Without loss of generality, we consider $\theta \in \mathbb{R}^{(0,1]^{d_i N}}$, where d_i denotes the dimensionality of the parameter values θ_i corresponding to E_i , and $\theta_i = 0$ denotes E_i is absent from the particular instance of \mathbf{E} . Using this representation, we address the first issue by learning \mathbf{E} and θ given a large training corpus of image pairs (I_s, I_t) and their corresponding edit operations. To address the second issue, we note that textual descriptions of the edit operations, if created in the forward design process, can provide the relevant context for the edits and help understand the semantic relationships between I_s and I_t . To this end, we can rewrite the relationship between I_s and I_t as

$$I_t = \mathbf{E}(I_s, c; \theta), \quad (2)$$

where c denotes the textual instructions as a sequence of words in a dictionary. Consequently, our goal becomes identifying \mathbf{E} and θ using I_s, I_t and c . Since the data available in this case is multimodal in nature, we propose a multimodal learning network consisting of two streams. We design one stream (Stream 1) to capture the information available from the image pair (I_s, I_t) , and the other stream (Stream 2) to learn semantic information from the text c that complements the information learned in Stream 1. Since we consider the function space of \mathbf{E} to be a finite set of categorical edit operations, we can model the problem of solving for \mathbf{E} as a multilabel classification problem. Correspondingly, solving for θ becomes a multi-value regression task. Our goal is to jointly solve for \mathbf{E} and θ . Therefore, we propose a multimodal multitask learning (MML) network, which we call **TAME-RD**. Next, we discuss the different components of **TAME-RD**'s MML network.

3.2 Stream 1: Pre-Post Effect From Images

We design this stream based on the concept of a specific type of treatment effect referred to as *pre-post effect*. It is widely used to understand the effectiveness of interventions introduced in fields like economics, medicine and healthcare, and education (Cuijpers et al., 2017; Estrada et al., 2019). Pre-post effect (δ) essentially compares the outcomes of a group of individuals before (O_0) and after (O_1) receiving treatment or intervention.

$$\delta = O_1 - O_0 \quad (3)$$

While it may reveal some characteristics of the treatment, it primarily reflects the treatment’s impact on the subjects’ outcomes. In our problem setup, we have access to an edited image that was the outcome of treatment (edit operations) received by the source image. Consequently, we use the pre-post effect to capture the distribution underlying the changes between the two images, to understand the nature (edit operation types) and the extent (corresponding parameter values) of the change. Following Eqn. 3, we define δ_I as

$$\delta_I = I_t - I_s \quad (4)$$

and use it as the input to this mode. Now we use ResNet-18, which is a variant of the ResNet architecture, that has been pre-trained on the ImageNet dataset (Deng et al., 2009), to encode the visual features from δ_I to get f_{ppe} . This will allow us to effectively capture the information available in δ_I despite its big range of values compared to the actual images I_s and I_t (0, 255). Therefore,

$$f_{ppe} = RESNET_{18}(\delta_I) \quad (5)$$

3.3 Stream 2: Context From Language

In certain situations, the difference between the edited and original images, denoted as δ or the pre-post effect, can be sufficient to determine the specific editing operation due to the distinctive characteristics displayed by δ . However, as mentioned earlier, it is common for δ to lack distinct uniqueness, or the inference of the editing operation might be influenced by various factors or conditions. To address this issue, we utilize textual information (c) as a valuable source to correctly interpret the significance of δ . This textual information is presented as natural language sentences, which offer insights into the changes that may not be directly discernible from the images themselves.

We employ pre-trained Siamese BERT networks (SBERT) (Reimers and Gurevych, 2019) to encode these sentences. SBERT embeddings are designed to capture semantic similarities rather than a verbatim representation of the sentence. This approach ensures that the model’s behavior remains consistent even with changes in the wording of the text, as long as the underlying semantics of the text remain unchanged. Therefore, we can define the textual features as:

$$f_c = SBERT(c) \quad (6)$$

3.4 Multitask Network

The features extracted from stream 1, denoted as f_{ppe} , and the features obtained from stream 2, denoted as f_c , are combined through concatenation. These concatenated features are then fed into a fully connected layer, which, therefore, produces an embedding capable of capturing the relationships between the two modalities. Now in order to solve the multitask part of the problem, we choose the first half (h_{cls}) of this embedding to represent the feature corresponding to the classification task while the second half (h_{reg}) represents the features needed for the regression task. Each of these embeddings are then passed into equal number of fully connected neural network layers to learn their respective tasks. Therefore,

$$h_{mtl} = FC(concat(f_{ppe}, f_c)) \quad (7)$$

$$h_{cls} = h_{mtl}[:, e_{mtl}] \quad (8)$$

$$h_{reg} = h_{mtl}[e_{mtl}, :] \quad (9)$$

In this work, we have taken $e_{mtl} = \frac{1}{2} \text{len}(h_{mtl})$

For training the multitask network, we use the following loss function

$$L_{net} = L_{mlcls} + \lambda L_{reg}, \quad (10)$$

where L_{mlcls} is the loss function for learning the multilabel classification task, and L_{reg} is the loss function used for learning the multi-valued regression task. We consider the cross-entropy loss function for L_{mlcls} and the ℓ_1 loss for L_{reg} .

4 Datasets

We provide details on the datasets we experiment with: the benchmark GIER dataset (Shi et al., 2020) and our proposed dataset I-MAD.

4.1 GIER Dataset

The Grounded Image Editing Request (GIER) dataset (Shi et al., 2020) comprises 6,179 source and edited image pairs with pixel- and image-level edit operations. The dataset includes 23 edit operations and natural language instructions for each image pair. On average, each image has about 3.21 edit operations. However, GIER is curated from public sites like Reddit.com and Zhopped.com, and as such, does not provide parameter values for the edit operations. In our work, we augmented the GIER dataset with approximately computed parameter values based on edit operation definitions used by (Shi et al., 2020) for language-based image editing. We provide the details in the appendix, acknowledging the difficulty in guaranteeing identical values used by the creators for obtaining edited images. Given multiple potential operations for each image pair (hence multi-label), we associate a parameter value with each operation for every image pair.

4.2 I-MAD: Our Dataset

GIER is the closest available dataset for our use case. However, crucially for reverse design processes, the exact parameter values associated with the edit operations are unavailable. To address this gap, we introduce a novel dataset named the Image Multi-Adjustment Dataset (I-MAD). I-MAD consists of two separate parts: I-MAD-Dense and I-MAD-Pro. Both parts contain paired source images, edited images, and natural language text instructions or contextual information underlying the edit operations. The triplets are annotated by the set of operations performed on the source image to obtain the edited image and also their corresponding parameter value. We only consider the set and not the sequence of the operations as non-commutative operations (such as brightness and contrast) incur negligible errors when changing the sequence, within the operating ranges computed from GIER and our dataset. I-MAD-Dense and I-MAD-Pro mainly differ in the preparation methods to obtain the triplets. We discuss the highlights for each of these subsets of I-MAD.

I-MAD-Dense. This dataset segment comprises 100K triplets generated in an uncontrolled setting. We obtained the source images from COCO (Lin et al., 2014), which has been vetted to have non-harmful content (Ha et al., 2024). COCO contains everyday objects and humans in natural scenes in a

variety of styles, including paintings and cartoons. We applied 5 (without repetition) edit operations to each source image, selected randomly from a pool of 12 operations in total. We provide more details of the operations in the appendix. For the text descriptions in the triplets, we gathered 20 to 25 vaguely phrased descriptions in English for each of the 12 edit operations. However, these descriptions do not provide information about the parameter values associated with the edit operations, nor do they contain image-specific details. They are generic descriptions, such as "Create an image that has a more balanced and natural appearance" or "Add a filter that mimics the warm and golden light of a summer sunset." For each of the 5 randomly selected operations, we randomly chose a sentence from their set of text descriptions. We also choose the parameter values for the edit operations randomly within predetermined ranges. Despite its random nature, this approach holds significance. A network capable of accurately identifying image edit operations would inherently grasp their underlying distribution space, avoiding mere memorization of coincidental operation sequences. From a real-world perspective, providing high-level natural language instructions for edit operations is quicker and more straightforward for individuals compared to crafting image-specific instructions. I-MAD-Dense’s intentional lack of one-to-one correspondence and the inclusion of randomness in edit operations forces the model to comprehend the individual contributions of each image edit operation rather than relying on memorized sequences. Additionally, I-MAD-Dense also benefits from being created at a fraction of the cost of human-curated datasets.

I-MAD-Pro. To gather this segment of I-MAD, we hired professional editors, at a payment rate in agreement with them. They were tasked with editing images using their unique creative processes and then supplying us with both the source and the edited images. Additionally, they were required to submit detailed textual instructions in English outlining their workflows, along with a breakdown of the editing operations conducted between the source and the edited images. This included specifying the parameter values associated with each operation. Given the intensive nature of this data collection process, we have collected a set of 1,674 data samples spanning 30 edit operations at about 4.55 operations per image pair. The textual instructions provided by editors

are intentionally high-level, similar to the GIER dataset. However, different from GIER, which collected edited images from public platforms with no explicit guardrails for quality, our edited images were vetted for quality by domain experts. Our editors also vetted the images in our dataset to ensure there was no identifiable personal or private content, as well as no offensive or harmful content. Any images they flagged with such content were filtered out of the dataset. The editors were given samples from the COCO dataset (but these samples are different than the ones used to create I-MAD-Dense) following the usage terms in its public license. For the edit operations and natural language instructions our editors provided, they agreed to transfer us the ownership as part of the hiring process.

For our experiments, we individually normalized the parameter values for each edit operation based on their observed ranges. This approach ensured consistent representation across all the operations.

5 Experiment Results and Analyses

We discuss the hyperparameter details and metrics in Sec. 5.1, baselines and quantitative evaluations in Sec. 5.2, and ablation experiments in Sec. 5.3.

5.1 Training Details and Evaluation Metrics

We test our model on three datasets, namely, I-MAD-Dense, I-MAD-Pro, and GIER. To train our model on each of the two segments of I-MAD (Dense and Pro), we adhere to conventional practices by partitioning the dataset into three subsets. In GIER, we use the standard train, validation, and test split ratios provided in the dataset. We use a batch size of 32 for both datasets. The models are trained for 300 epochs. We use Adam optimizer with a learning rate of 0.0001. All our results are generated on NVIDIA GeForce GTX 2080 Ti GPU and all codes were implemented using Pytorch (Paszke et al., 2019). We evaluate our multi-label classification tasks using standard metrics: accuracy, average precision (AP), and F1. For multi-valued regression, we employ root mean squared error (RMSE) and concordance correlation coefficients (CCC).

5.2 Quantitative Comparisons

Given the absence of prior methods for our problem (Sec. 3.1), we compare with the most relevant

state-of-the-art method of Tan et al. (2019). For comprehensiveness and completeness, we also investigate various baseline approaches.

1. **Tan et al. (2019).** Tan et al. (2019) introduce a transformer-based model with a dynamic relational attention mechanism to compute alignment scores between source and edited image features during each decoding step, preventing information loss. While originally designed for text generation, we re-purpose their model and retrain it to generate sequences of edit operations given source and edited images. However, their approach cannot estimate parameter values. Table 1 (rows 3, 15, 28) compares the performance of Tan et al. (2019)’s approach with ours. We note that their approach of defining image dissimilarity based on learned feature alignment fails to adapt to scenarios where multiple edit operations with diverse parameter values are applied to source images. By contrast, our approach explicitly models observed differences between images to learn the fine-grained changes and learns the continuous parameter space of the edit operations, leading to substantial performance gains.
2. **Modification of Tan et al. (2019).** For completeness, we also compare with a modified version of Tan et al. (2019), where we replace their LSTM layer, which generates text sentence outputs, with an MLP to perform multi-label classification. We report the performance of this modified model in Table 1 (rows 4, 16, 29). This change does not significantly alter the overall performance of Tan et al. (2019).
3. **CLIP-Based Backbone.** We replace the ResNet-18 backbone with CLIP (Radford et al., 2021) in Stream 1 by computing differences between CLIP-derived features for source and edited images. Also, in Stream 2, we replace the SBERT features with CLIP features. Importantly, we maintain consistent feature sizes from the pre-trained models of both CLIP and ResNet-18. This ensures that the overall parameter count remains unchanged in the CLIP-based architecture. We report the performance of this architecture in Table 1 (rows 6, 18, 31). With the same parameter count, the performance differences between the CLIP and ResNet-18 variants highlight the

advantages of our Stream 1 design.

4. **Dual Image Encoders with ResNet in TAME-RD.** To understand the usefulness of the features we learn from image differences, we experiment with a variation that uses individual image encoders for the source and the edited images. We complete this pipeline by concatenating the features obtained from each of the image encoders to form f_{img} , unlike f_{ppe} in TAME-RD. We keep the rest of the network the same as TAME-RD. This architecture leads to significant performance drops (Table 1, rows 7, 19, 32) due to its focus on the individual image features rather than a nuanced understanding of the image differences. This further corroborates the non-trivial nature of learning from image differences.
5. **TAME-RD with Shallow CNN backbone.** We replace our ResNet-18 backbone with a shallow CNN (Table 1, rows 5, 17, 30) to investigate the contributions of the additional parameter load in our proposed ResNet backbone. The ResNet backbone indeed leads to significant gains over a shallow CNN.
6. **Dual Image Encoders with shallow CNN in TAME-RD.** We also use shallow CNN architectures to encode the source and the edited images separately and then combine them with the text features to learn the image edit operations and the parameter values. We report this baseline in Table 1 (rows 8, 20, 33).
7. **GPT-4 Predictor.** We prompt GPT-4 (Achiam et al., 2023) to act as an expert editor and ask it to solve our specific task by providing it with source and edited images and corresponding text descriptions (Table 1, rows 2, 14, 27). We report the full prompt in our appendix. While GPT-4 performs relatively well in I-MAD-Dense, it fails to make any predictions for 95 and 327 samples in I-MAD-Pro and GIER.
8. **Most Frequent Operations.** To establish a performance lower-bound, we use a brute-force approach that simply assigns the top five most frequent operations and the corresponding expected parameter values for all the image pairs in the datasets (Table 1, rows 1, 26). Since I-MAD-Dense was a result of random

Table 1: **Quantitative Results.** We show the quantitative performances on our proposed I-MAD and GIER (Shi et al., 2020). Bold denotes **best**, underline denotes second-best. pt = pretrained ResNet18, fs = ResNet18 trained from scratch. We observe an overall state-of-the-art performance of our proposed method. The higher AP and F1 of the ResNet50 backbone are due to overfitting. * GPT-4 values only computed for samples where it could provide results. It failed to provide any parameter values for I-MAD-Pro.

| Data | Method | Evaluation Metrics | | | | |
|--|----------------------------|--------------------|---------------|---------------|-------------------|----------------|
| | | Acc. \uparrow | AP \uparrow | F1 \uparrow | RMSE \downarrow | CCC \uparrow |
| 1 | Most Freq. | 0.210 | 0.451 | 0.398 | 0.270 | 0.191 |
| | GPT-4 Pred.* | 0.017 | 0.585 | 0.383 | × | × |
| | Tan et al. (2019) | 0.026 | 0.001 | 0.003 | — | — |
| | Modified Tan et al. (2019) | 0.650 | 0.001 | 0.002 | — | — |
| I-MAD-Pro (~ 1K samples) | Shallow CNN backbone | 0.877 | 0.998 | 0.998 | 0.095 | 0.877 |
| | CLIP backbone | 0.809 | 0.999 | 0.999 | 0.086 | 0.889 |
| | Dual Enc. ResNet18 | 0.802 | 0.998 | 0.998 | 0.087 | 0.886 |
| | Dual Enc. Shallow CNN | 0.879 | 0.998 | 0.998 | 0.100 | 0.883 |
| | Only Stream 1 | 0.889 | 0.999 | 0.999 | 0.086 | 0.893 |
| | Only Stream 2 | 0.894 | 0.944 | 0.933 | — | — |
| | ResNet50 backbone | 0.903 | 0.999 | 0.999 | 0.077 | 0.917 |
| | TAME-RD (Ours, pt) | 0.889 | <u>0.999</u> | <u>0.999</u> | <u>0.084</u> | <u>0.906</u> |
| | TAME-RD (Ours, fs) | <u>0.896</u> | 0.999 | 0.999 | 0.090 | 0.882 |
| | GPT-4 Pred. | 0.951 | 0.960 | 0.971 | 19.30 | 0.001 |
| | Tan et al. (2019) | 0.453 | 0.002 | 0.002 | — | — |
| | Modified Tan et al. (2019) | 0.448 | 0.002 | 0.002 | — | — |
| I-MAD-Dense (100K samples) | Shallow CNN backbone | 0.780 | 0.944 | 0.966 | 0.261 | 0.440 |
| | CLIP backbone | 0.709 | 0.765 | 0.875 | 0.398 | 0.047 |
| | Dual Enc. ResNet18 | 0.747 | 0.736 | 0.794 | 0.377 | 0.044 |
| | Dual Enc. Shallow CNN | 0.544 | 0.950 | 0.966 | 0.345 | 0.229 |
| | Only Stream 1 | 0.783 | 0.827 | 0.874 | 0.386 | 0.098 |
| | Only Stream 2 | 0.794 | 0.818 | 0.866 | — | — |
| | ResNet50 backbone | 0.799 | <u>0.995</u> | <u>0.997</u> | 0.294 | 0.415 |
| | TAME-RD (Ours, pt) | 0.841 | 0.855 | 0.896 | <u>0.258</u> | <u>0.447</u> |
| | TAME-RD (Ours, fs) | <u>0.878</u> | 0.998 | 0.999 | 0.182 | 0.775 |
| | Most Freq. | 0.660 | 0.303 | 0.646 | 32.58 | 0 |
| | GPT-4 Pred.* | 0.510 | 0.458 | 0.71 | 125.2 | 0.096 |
| | Tan et al. (2019) | 0.151 | 0.007 | 0.007 | — | — |
| | Modified Tan et al. (2019) | 0.149 | 0.007 | 0.007 | — | — |
| GIER (Shi et al., 2020) (~ 6K samples) | Shallow CNN backbone | 0.707 | <u>0.902</u> | <u>0.940</u> | 4.516 | 0.406 |
| | CLIP backbone | <u>0.737</u> | 0.524 | 0.570 | 4.614 | 0.061 |
| | Dual Enc. ResNet18 | 0.463 | 0.763 | 0.842 | 4.562 | 0.236 |
| | Dual Enc. Shallow CNN | 0.562 | 0.858 | 0.911 | 4.542 | 0.332 |
| | Only Stream 1 | 0.656 | 0.691 | 0.787 | 4.576 | 0.095 |
| | Only Stream 2 | 0.534 | 0.568 | 0.673 | — | — |
| | ResNet50 backbone | 0.727 | 0.686 | 0.732 | 4.599 | <u>0.414</u> |
| | TAME-RD (Ours, pt) | 0.745 | 0.749 | 0.834 | 4.514 | 0.447 |
| | TAME-RD (Ours, fs) | 0.700 | 0.915 | 0.949 | <u>4.515</u> | 0.413 |

assignment of operations, we only consider I-MAD-Pro and GIER for this experiment.

5.3 Ablation Experiments

We perform the following two ablations.

1. **Contribution of each stream.** To understand the contributions of Streams 1 and 2, we run TAME-RD on both GIER and I-MAD by removing one stream at a time. Table 1 (rows 9, 21, 34, and 10, 22, 35) shows that the performance of either stream is comparable to the other, but when combined, they improve the end-to-end performance by a big margin. This corroborates our discussions in Sec. 3.1. Interestingly, our Stream 1 also outperforms the state-of-the-art baseline of Tan et al. (2019),







| | Source | Request | Target | Ground Truth | Our Predictions |
|---|---|--|---|----------------|-----------------|
| A |  | Clean up and make colors pop? This was my grandfather on Father's Day around 1956. |  | Brightness 1.3 | Brightness 1.2 |
| | | | | Contrast -33.6 | Contrast -0.1 |
| | | | | Hue 1.0 | Hue 0.75 |
| | | | | Lightness 1.3 | Lightness 0.91 |
| | | | | Saturation 1 | Saturation 1.2 |
| | | | | Sharpness 0.3 | Sharpness 0 |
| | | | | Tint 1.5 | Tint 1.16 |
| | | | | | |
| B |  | Enhance the image with a wintry, icy feel. Create it's warm and inviting version. |  | Summer 1 | Summer 0.4 |
| | | | | Winter 1 | Winter 1.32 |
| C |  | Give the image a vintage look. Duplicate the image in a flipped orientation. |  | Flip 1 | Flip 1.19 |
| | | | | Sepia 1 | Sepia 0.97 |

Figure 6: **Qualitative Results.** We show some qualitative results of our method. Row A is from the GIER dataset (Shi et al., 2020), and rows B and C are from our dataset *I-MAD-Dense*. We note that our method matches the ground truth for a large variety of edit operations.

which similarly uses only images to predict the edit operations. This further highlights the contribution of the pre-post effect in the learning process.

2. **Increasing *TAME-RD*'s parameters.** We compare two variants, one with a ResNet-18 backbone and the other with a ResNet-50 backbone, to investigate whether increasing the model parameters leads to better performance (Table 1, rows 11, 23, 36). We note that more parameters do not necessarily improve performance. *TAME-RD* with the ResNet-50 backbone has poorer overall performance compared to ResNet-18 due to overfitting, indicating our task cannot be solved by merely increasing the network complexity.

5.4 Qualitative Results

We also show some qualitative results of our method on GIER and *I-MAD-Dense* in Fig. 6.

6 Conclusion

We have presented *TAME-RD*, a method to automatically detect *fully-specified* edit operations given pairs of source and edited images, and optionally, contextual information in the form of natural language text. Our fully-specified edit operations consist of categorical edits paired with corresponding parameter values. We have also proposed a

corresponding dataset *I-MAD*, consisting of 100K tuples (source image, edited image, text) annotated with fully-specified edit operations between the image pairs, to enable further research on this task as well as downstream image editing tasks.

7 Limitations and Future Work

Our work also has some limitations. Our multi-modal network does not explicitly attend to the individual components in the images, such as objects and scene segments, that can provide additional context for the edit operations. To this end, the features learned in the image stream in our network can be combined with current techniques for object detection and scene segmentation to improve prediction performance. Further, while we are constrained to define a finite set of categorical edits for practical viability, there is scope to expand the number of edits in our dataset for more fine-grained learning. Lastly, we note that while a mathematically optimal set of edit operations may not exist between source and edited image pairs, especially given the non-linearity of the edit operations, we can fine-tune our network on more diverse samples, as well as incorporate negative samples (mismatched contextual information and edited images) to keep improving performance.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Gary Bradski. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123.
- Pim Cuijpers, E Weitz, IA Cristea, and J Twisk. 2017. Pre-post effect sizes should be avoided in meta-analyses. *Epidemiology and psychiatric sciences*, 26(4):364–368.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Konstantinos G Derpanis. 2010. Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3.
- Eduardo Estrada, Emilio Ferrer, and Antonio Pardo. 2019. Statistics for evaluating pre-post change: Relation between change in the distribution center and change in the individual scores. *Frontiers in psychology*, 9:2696.
- Yingchaojie Feng, Xingbo Wang, Bo Pan, Kam Kwai Wong, Yi Ren, Shi Liu, Zihan Yan, Yuxin Ma, Huamin Qu, and Wei Chen. 2023. Xnli: Explaining and diagnosing nli-based visual data analysis. *IEEE Transactions on Visualization and Computer Graphics*.
- Kaitlin Fitzgerald, Zhiying Yue, Jody Chin Sing Wong, and Melanie C Green. 2022. Entertainment and social media use during social distancing: Examining trait differences in transportability and need for social assurance. *Psychology of Popular Media*, 11(3):305.
- Tsu-Jui Fu, Xin Eric Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. 2020. Sscr: Iterative language-based image editing via self-supervised counterfactual reasoning. *arXiv preprint arXiv:2009.09566*.
- Tsu-Jui Fu, Xin Eric Wang, and William Yang Wang. 2022. Language-driven artistic style transfer. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 717–734. Springer.
- Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. In *ACM SIGGRAPH 2009 papers*, pages 1–9.
- Zixin Guo, Tzu-Jui Julius Wang, and Jorma Laaksonen. 2022. Clip4idc: Clip for image difference captioning. *arXiv preprint arXiv:2206.00629*.
- Eungyeom Ha, Heemook Kim, and Dongbin Na. 2024. Hod: New harmful object detection benchmarks for robust surveillance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 183–192.
- Abid Haleem, Mohd Javaid, Mohd Asim Qadri, and Rajiv Suman. 2022. Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers*.
- Ronghang Hu and Amanpreet Singh. 2021. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449.
- Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. 2018. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17.
- Amila Jakubović and Jasmin Velagić. 2018. Image feature matching and object detection using brute-force matchers. In *2018 International Symposium ELMAR*, pages 83–86. IEEE.
- Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. *arXiv preprint arXiv:1808.10584*.
- Wentao Jiang, Ning Xu, Jiayun Wang, Chen Gao, Jing Shi, Zhe Lin, and Si Liu. 2021. Language-guided global image editing via cross-modal cyclic mechanism. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2115–2124.
- Junfeng Jing, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun. 2022. Recent advances on image edge detection: A comprehensive review. *Neurocomputing*.
- Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385.
- Soohyun Kim, Jongbeom Baek, Jihye Park, Gyeongnyeon Kim, and Seungryong Kim. 2022. Instaformer: Instance-aware image-to-image translation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18321–18331.
- Grzegorz Krawczyk, Rafal Mantiuk, Dorota Zdrojewska, and Hans-Peter Seidel. 2007. Brightness adjustment for hdr and tone mapped images. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pages 373–381. IEEE.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

- David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110.
- Trisha Mittal, Vishy Swaminathan, Somdeb Sarkhel, Ritwik Sinha, David Arbour, Saayan Mitra, and Dinesh Manocha. 2021. Bohance: Bayesian optimization for content enhancement. In *2021 IEEE International Symposium on Multimedia (ISM)*, pages 17–24. IEEE.
- Jihye Park, Soohyun Kim, Sunwoo Kim, Jaejun Yoo, Youngjung Uh, and Seungryong Kim. 2022. Lanit: Language-driven image-to-image translation for unlabeled data. *arXiv preprint arXiv:2208.14889*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Eduardo Rinaldi, Davide Sforza, and Fabio Pellacini. 2023. Nodigit: Diffing and merging node graphs. *ACM Transactions on Graphics (TOG)*, 42(6):1–12.
- Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11.
- Ragav Sachdeva and Andrew Zisserman. 2023. The change you want to see. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.
- Jing Shi, Ning Xu, Trung Bui, Franck Deroncourt, Zheng Wen, and Chenliang Xu. 2020. A benchmark and baseline for language-driven image editing. In *Proceedings of the Asian Conference on Computer Vision*.
- Jing Shi, Ning Xu, Yihang Xu, Trung Bui, Franck Deroncourt, and Chenliang Xu. 2021. Learning by planning: Language-guided global image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13590–13599.
- Daniel Steininger, Andreas Trondl, Gerardus Croonen, Julia Simon, and Verena Widhalm. 2023. The cropandweed dataset: A multi-modal learning approach for efficient crop and weed manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3729–3738.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2018. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*.
- Jianxin Sun, Qi Li, Weining Wang, Jian Zhao, and Zhenan Sun. 2021. Multi-caption text-to-face synthesis: Dataset and algorithm. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2290–2298.
- Adobe Systems. 2002. *Adobe Photoshop 7.0*. Adobe Press.
- Hao Tan, Franck Deroncourt, Zhe Lin, Trung Bui, and Mohit Bansal. 2019. Expressing visual relationships via language. *arXiv preprint arXiv:1906.07689*.
- C Lee Ventola. 2014. Social media and health care professionals: benefits, risks, and best practices. *Pharmacy and therapeutics*, 39(7):491.
- Yang Wu, Zijie Lin, Yanyan Zhao, Bing Qin, and Li-Nan Zhu. 2021. A text-centered shared-private framework via cross-modal prediction for multimodal sentiment analysis. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 4730–4738.
- Hongwei Xue, Haomiao Liu, Jun Li, Houqiang Li, and Jiebo Luo. 2020. Sed-net: Detecting multi-type edits of images. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.
- An Yan, Xin Eric Wang, Tsu-Jui Fu, and William Yang Wang. 2021. L2c: Describing visual differences needs semantic understanding of individuals. *arXiv preprint arXiv:2102.01860*.
- Jingchun Zhou, Lei Pang, Dehuan Zhang, and Weishi Zhang. 2023. Underwater image enhancement method via multi-interval subhistogram perspective equalization. *IEEE Journal of Oceanic Engineering*.

Appendix

We discuss additional details related to preparing the datasets for our work.

A Ethical Considerations and Risks

The GIER dataset is freely accessible to the public. In the context of this research, we have only inferred additional information from the dataset and have not made any alterations to the provided data itself. The data collection process for I-MAD does not include any personal, private or sensitive information, and was deemed exempt from an ethics review. Also, we used images from the publicly available COCO dataset following the usage terms of its public license. For the edit operations and natural language instructions our editors provided, they agreed to transfer us the ownership as part of the hiring process.

While there are no immediate risks associated with our work, it is essential to emphasize that the practice of reverse designing in general should be approached with careful consideration of intellectual property and ethical concerns.

B GIER Dataset

GIER (Shi et al., 2020) contains triplets of (source image, edited image, natural language request), together with annotations of the categorical edit operations performed between the source and the edited images. However, these edit operations are not *fully specified* in that the corresponding parameter values are not available. Therefore, for the purposes of this work, we annotate the dataset with the parameter values corresponding to the edit operations where feasible and detail our process below.

Contrast. Following the definition of contrast given in (Shi et al., 2020; Hu et al., 2018), we first compute the luminance of the source image I_s as

$$Lum(I_s) = 0.27I_{sr} + 0.67I_{sg} + 0.06I_{sb}, \quad (11)$$

where I_{sr}, I_{sg}, I_{sb} correspond to the RGB channels of I_s . We then write the enhanced luminance (EnLum) as

$$EnLum(I_s) = \frac{1}{2} (1 - \cos(\pi Lum(I_s))). \quad (12)$$

This allows us to compute the image with the enhanced contrast (EnCon) as

$$EnCon(I_s) = \frac{EnLum(I_s)}{Lum(I_s)} I_s. \quad (13)$$

We can define the edited image I_t as the combination of the enhanced contrast and the original image as

$$I_t = (1 - p) I_s + p EnCon(I_s), \quad (14)$$

which allows us to solve for the contrast control parameter p .

Sharpness. We solve for sharpness control parameter p using the equation

$$I_t = I_s + p \Delta^2 I_s. \quad (15)$$

Flip. We checked if the edited image has been flipped left to right or top to bottom using OpenCV library (Bradski, 2000).

$$\text{flip value} = \begin{cases} 1 & \text{if image flipped left to right} \\ 2 & \text{if image flipped top to bottom} \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

We use this to also identify flip actions for the “flip obj” edit operation in the GIER dataset.

Rotation. For rotation, the parameter is the angle of rotation from the source to the edited images w.r.t. an axis perpendicular to the plane of the images. We estimate this by aligning both the source and the edited images according to the following procedure:

1. Use SIFT (Lowe, 2004) to find the key points and descriptors of both the source and the edited image.
2. Match the descriptors in both images using the BFMatcher algorithm (Jakubović and Velagić, 2018).
3. Use the matched points to estimate the homography matrix between the two images using the RANSAC algorithm (Derpanis, 2010).
4. Extract the angle of rotation from the homography matrix.

We use this method to identify the angle of rotation for both “rotate” and “rotate obj” edit operations in the GIER dataset.

Saturation. We convert the source and the edited images to the HSV color spaces. We then compute the mean saturation value of each of the source and the edited images and compute the saturation factor p using

$$S(I_t) = p S(I_s) \quad (17)$$

where $S(\cdot)$ refers to the saturation computation function available in OpenCV (Bradski, 2000).

Brightness and Lightness. For brightness, we compute the scaling factor p between the intensities of the source and the edited images as

$$B(I_s) = pB(I_t), \quad (18)$$

where $B(\cdot)$ is the function defined to compute the intensity of the image available in OpenCV (Bradski, 2000). We also use this method to compute the parameter values for the "lightness" edit operation in the GIER dataset, following the fact that both operations are similarly designed (Shi et al., 2020).

Hue. We convert the source and the edited images to the HSV color spaces. We then compute the mean hue value of each of the source and the edited images and compute the factor p as

$$H(I_t) = pH(I_s), \quad (19)$$

where $H(\cdot)$ refers to the hue computation function available in OpenCV (Bradski, 2000).

Tint. To understand variations in the image tint, we represent both the source and the edited images in the LAB color space and compute the mean color of each channel ('a' and 'b') in the LAB color space: 'a' represents color in the red-to-green axis, and 'b' represents color in the blue-to-yellow axis. We can then compute the tint factor p as the ratio of the mean values of the 'a' and 'b' channels of the tinted image and the original image in LAB color space.

C I-MAD: Additional Details

We describe additional details related to our new dataset.

C.1 I-MAD-Pro: Instructions Provided to Professional Editors

This is the full instruction sent with the recruitment email:

We are creating a dataset of adjustment operations performed on images satisfying creative user needs. We are conducting the data collection using Adobe Photoshop, and as a result, only freelancers with active Photoshop licenses are eligible for this project. To collect the dataset, we provide freelancers with a set of images. We ask them to provide a one- or two-line comment on what can

be adjusted in the image from their creative perspective (for example, "This image is a bit too dark and has some background clutter"), then perform the desired edit operations in Photoshop and tabulate them (for example, brightness: +5%, object removal: between pixels (a, b) and (c, d)), and provide the edited image. To better organize the data collection process, we also provide a "universal" set of around 20 edit operations, so that for each image, freelancers can start off by thinking of possible adjustments to the image using edits within that universal set. Freelancers are, of course, free to think of adjustments beyond the universal set, in which case we will add the new edit operations to the universal set. To summarize the deliverables: Given a set of source images and a universal set of around 20 edit operations, please provide, for each source image,

1. *A one- or two-line comment in English on how to improve the image. This comment need not specifically mention edit operation names but only convey a general sense of what can be improved.*
2. *The edited image, created using a particular version of Adobe Photoshop.*
3. *A tabulated entry of the exact edits performed (detailed instructions on how to tabulate will be provided in a readme along with the source images). We typically expect between 2 and 4 edit operations per image.*

C.2 I-MAD-Pro Distribution

We report the edit operations and the corresponding number of samples in the I-MAD-Pro dataset in Table 3. To prepare the dataset, we collected all edit operations performed by all the editors and filtered out the samples that contained edit operations appearing less than 10 times across all the editors.

C.3 I-MAD-Dense Distribution

For I-MAD-Dense, we maintain a list of 12 edit operations described in Table 2. For the corresponding text descriptions, we collected 20 – 25 ways of describing each edit operation at a high level in English. To create the dataset, we randomly selected 30,000 images from the COCO dataset (Lin et al., 2014). For each image we selected, we chose a random sequence of 5 unique edit operations out of those 12, and applied them one after the other. For every selected edit operation, we also randomly

Table 2: **I-MAD-Dense**. We provide the list of categorical edits in our proposed dataset, together with the corresponding total number of tuples (source image, target image, text) and the ranges of the parameter values. For some edit categories, such as grayscale, HDR, flip, and pre-defined filters such as sepia, summer and winter, there are no associated parameter values.

| Sl. | Edit Operation | # Samples | Range of Values |
|-----|---------------------------|-----------|-----------------|
| 1 | grayscale | 8772 | — |
| 2 | HDR | 8732 | — |
| 3 | brightness | 8722 | 10 – 100 |
| 4 | summer (instagram filter) | 8689 | — |
| 5 | sepia (instagram filter) | 8639 | — |
| 6 | rotate | 8638 | 90, 180, 270 |
| 7 | winter (instagram filter) | 8615 | — |
| 8 | flip | 8385 | — |
| 9 | sharpness | 8383 | 10 – 100 |
| 10 | contrast | 8094 | 50 – 150 |
| 11 | gaussian filter | 7922 | 25 – 125 |
| 12 | saturate | 6409 | 5 – 30 |

selected a sentence from its set of text descriptions. The randomness in the choice of images and edit operations ensures a wide variety of variations observed due to the different edit operations. However, the random choice also implies some edited images may have poor visual quality or not appear close to how humans would edit. Therefore, we enforced the following constraints to remove such images from the dataset.

1. **Complex Wavelet Structural Similarity Index Metric (CW-SSIM)**. This is capable of handling a wide range of geometric distortions without compromising its ability to measure the similarity between the source-edited images. We use this to ensure that the edited image is not the same as the source image, even after when various geometric distortions are involved in editing.
2. **Color Histogram**. We compare the color histograms of the source and the edited images to ensure that the edited image is not an overexposed or underexposed version of the source image.

Our choice of edit operations for **I-MAD-Dense** was influenced by two factors:

1. Frequency of edit operations reported by GIER. The top 5 most frequently used operations are all single-parameter based, taking up 67%. Overall, single-parameter operations make up 83% of all operations.
2. For our baseline approach, we wished to consider image edit operations that need single pa-

rameter values (Table 2) to completely specify them. **I-MAD-Pro** (Table 3) considers more complex operations such as object removal, cropping, flipping objects in the images, noise addition, and different camera filters.

C.4 I-MAD-Dense Quality Control

We report evaluation metrics to both benchmark the **I-MAD-Dense** dataset and show the performance of **TAME-RD**.

Per-Class Average Precision We report the per-class average precision (AP) on both the GIER and the **I-MAD-Dense** datasets for all the variations of our method we experiment with. In the GIER dataset (Fig. 7), we notice that all the variations perform poorly for some of the edit operations, such as blurring, denoising, rotation, and flip, but perform well for all other edit operations. We hypothesize that this may be due to our method overfitting to the training data because of a combination of two factors: few training samples for those edit operations and the structural differences between the source and the edited images for many of those edit operations not being sufficient for our method to learn representative features. We also note that the ResNet-18 variation of our method generally performs better than the ResNet-50 variation, indicating that more parameters do not lead to better performance on the smaller GIER dataset.

By contrast, in **I-MAD-Dense**, we have sufficient and diverse samples for all the edit operations, resulting in reasonably high average precisions for all of them. We also note that the ResNet-50 variation of our method generally performs better than the ResNet-18 variation, indicating that more parameters lead to some performance benefits in the larger dataset **I-MAD-Dense**.

C.5 Qualitative Results

We also show some qualitative examples where our method doesn’t perform as expected on GIER and **I-MAD-Dense** in Fig 9.

D Additional Experiment Details

We report the full prompt we used for the GPT-4 Predictor (baseline #7 in Sec. 5.2), and report the results of additional experiments.

D.1 Prompt for GPT-4 Predictor.

The text prompt provided for experiments on all datasets had the following template: *Perform the*

Table 3: **I-MAD-Pro Edit Operations**. We provide the list of categorical edits in our proposed I-MAD-Pro dataset, together with the corresponding total number of tuples (source image, target image, text) in the dataset.

| Sl. | Edit Operation | # Samples |
|-----|--------------------------|-----------|
| 1 | brightness | 752 |
| 2 | contrast | 748 |
| 3 | crf_dehaze | 582 |
| 4 | crf_temperature | 561 |
| 5 | crf_exposure | 515 |
| 6 | crf_tint | 463 |
| 7 | saturate (w/o colorize) | 462 |
| 8 | crop | 278 |
| 9 | sharpness | 255 |
| 10 | noise_strength | 222 |
| 11 | noise_preserve_details | 219 |
| 12 | hue (w/o colorize) | 184 |
| 13 | noise_reduce_color_noise | 141 |
| 14 | noise_sharpen_details | 135 |
| 15 | vibrance_vibrance | 131 |
| 16 | vibrance_saturate | 127 |
| 17 | camera_filter_shadow | 125 |
| 18 | lightness(w/o colorize) | 109 |
| 19 | shadow_gamma_correction | 69 |
| 20 | vibrance_exposure | 66 |
| 21 | remove_object | 65 |
| 22 | camera_filter_highlights | 59 |
| 23 | bw_R | 46 |
| 24 | bw_Y | 46 |
| 25 | bw_C | 45 |
| 26 | bw_G | 44 |
| 27 | exposure_offset | 44 |
| 28 | bw_B | 42 |
| 29 | bw_M | 41 |
| 30 | radial_filter | 35 |
| 31 | rotate_deg | 30 |
| 32 | bw_saturate (w/o tint) | 23 |
| 33 | color_cyan | 20 |
| 34 | color_yellow | 20 |
| 35 | color_magenta | 19 |
| 36 | saturate(w/ colorize) | 17 |
| 37 | bw_hue (w/o tint) | 15 |
| 38 | bw_saturate (w/ tint) | 14 |
| 39 | hue (w/ colorize) | 14 |
| 40 | bw_hue (w/ tint) | 13 |
| 41 | brush_blur_strength | 12 |
| 42 | lightness(w/ colorize) | 12 |

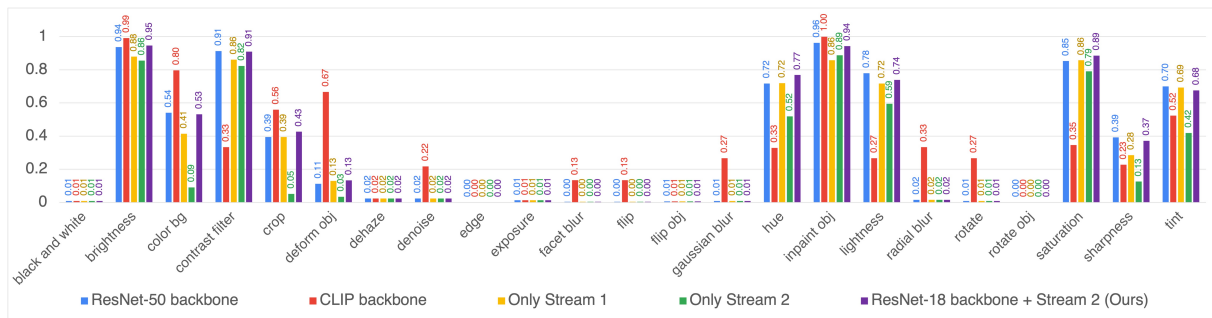


Figure 7: **Per-class Average Precision on GIER (Shi et al., 2020).** We show the average precisions for all variations of our method. We note poorer performance of all the variations of our method on some of the classes compared to others. We also note that the ResNet-18 variant with fewer parameters generally performs better than ResNet-50 on this smaller dataset.

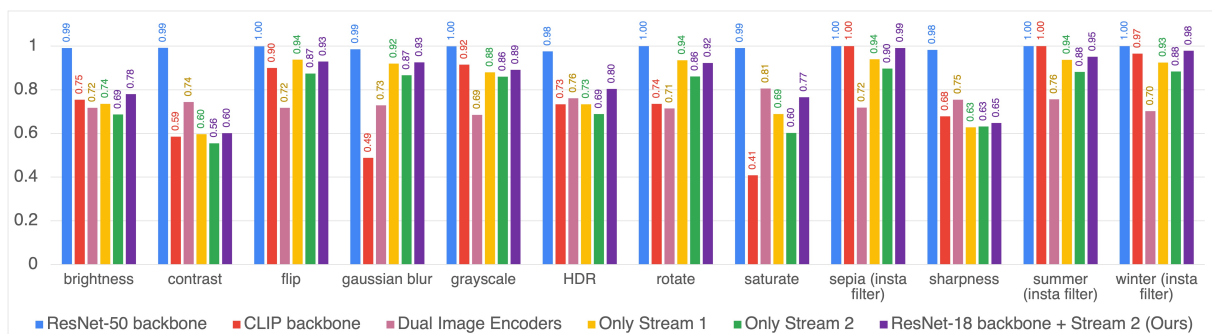


Figure 8: **Per-class Average Precision on I-MAD-Dense.** We show the average precisions for all variations of our method. We note the consistent performance of all the variations of our method on all the classes. We also note that the ResNet-50 variant with more parameters generally performs better than ResNet-18 on this larger dataset.

role of a professional editor for images. You are given two images and a text description. The first image is the source image and the second image is its corresponding edited image. The text description is a high-level description of underlying the edit operations between the first and the second images.

The following is the full set of available edit operations: <ENTER EDIT OPERATION NAMES>.

Consider only these edit operations and nothing else.

Your task is to determine the edit operations along with the exact parameter values such that applying those edit operations with those parameter values on the first image transforms it to the second image. Your task is also to ensure that the edit operations are consistent with the text descriptions. For example:

text description: nedit operations: n<5 EXAMPLES IN NEW LINES> nThink step by step to determine the edit operations and their parameter values from the given first image, the second image, and the text description. Ensure that the edit operations and the corresponding parameter values that you determine indeed transform the first image to the second image while being consistent with the text

description. Determine the edit operations and their parameter values again if you are not absolutely sure. Once you are sure of the edit operations and their parameter values, provide your response in the following format:

nedit operations: provide the comma-separated list of edit operations that you have determined.

nparameter value: provide the comma-separated list of parameter values in the same order as the edit operations.

nRespond with only these two lists and nothing else.

nBelow is the actual given text description for the given first and second images.

text description: <ACTUAL TEXT DESCRIPTION>.

For obtaining actual results on the different datasets, we replace the angular brackets <5 EXAMPLES IN NEW LINES> with 5 examples from the dataset we are testing on, and replace the angular brackets <ACTUAL TEXT DESCRIPTION> with the actual descriptions accompanying the image pairs in the dataset.







| | Source | Request | Target | Ground Truth | Our Predictions |
|---|---|--|---|--------------------------------------|------------------|
| A |  | Transform the image into shades of gray. Make the image more detailed. |  | Grayscale 1 Sharpness 0.5 | Contrast 0.01 |
| B |  | Please remove everything but the row of band members |  | Color_bg 1 Crop 1 Deform_obj 1 | Inpaint_obj 0.19 |
| C |  | Can someone replace the white and black with yellow and purple? While adding in a background color that suits? |  | Hue 1.5 Saturation 3.5 | Color_bg 0.13 |

Figure 9: **Failure cases.** We also show some failure cases of our method. Row A is from our dataset [I-MAD-Dense](#), and rows B and C are from the GIER dataset ([Shi et al., 2020](#)). We note that our method can be different from the ground truth for cases where there are potential sets of edit operations that are more minimal (such as in row A), or when edit operations are ambiguous even with the textual context (such as in row C).

Table 4: [TAME-RD](#) with different feature fusion strategies. ResNet18 has been trained from scratch for all cases shown.

| | Data | Method | Evaluation Metrics | | | | |
|---|--|---------------------------------|--------------------|---------------|---------------|-------------------|----------------|
| | | | Acc. \uparrow | AP \uparrow | F1 \uparrow | RMSE \downarrow | CCC \uparrow |
| 1 | I-MAD-Pro (\sim 1K samples) | Additive Fusion | 0.893 | 0.999 | 0.999 | 0.0944 | 0.785 |
| 2 | | Multiplicative Fusion | 0.884 | 0.999 | 0.999 | 0.095 | 0.787 |
| 3 | | End-to-End Concatenation (Ours) | 0.896 | 0.999 | 0.999 | 0.090 | 0.882 |
| 4 | GIER (Shi et al., 2020) (\sim 6K samples) | Additive Fusion | 0.627 | 0.667 | 0.818 | 4.52 | 0.386 |
| 5 | | Multiplicative Fusion | 0.669 | 0.701 | 0.840 | 4.523 | 0.384 |
| 6 | | End-to-End Concatenation (Ours) | 0.700 | 0.915 | 0.949 | 4.515 | 0.413 |

D.2 Additional Ablation Experiments

D.2.1 Fusion Strategy

According to Equation 7, we have opted for an end-to-end concatenation strategy. Unlike other available fusing methods, such as component-wise addition, multiplication, or other forms of cross-correlation, we do not assume any pairwise correlations in the features between the two streams. We let the network learn the importance of the individual feature components through the mapping using the fully connected (FC) layer. As a result, the fusion is a generalized combination of the two streams that encompasses special cases of pairwise correlations. Concatenation-based fusion is a popular fusion strategy in the multimodal learning literature ([Wu et al., 2021](#); [Hu and Singh, 2021](#)) because of its generalizability, which has inspired our choice. Nevertheless, for the sake of completeness,

we report performance numbers when using additive and multiplicative fusions in [TAME-RD](#) on the [I-MAD-Pro](#) and GIER datasets (Table 4).

D.2.2 Changing λ in Equation 10

λ is a scalar hyper-parameter to control the relative weightings of the classification and the regression losses, L_{mlcs} and L_{reg} respectively. If $\lambda = 0$, then L_{reg} has no effect on training. Similarly, if $\lambda \rightarrow \infty$, then L_{mlcs} has a negligible effect on training (we note that some influence of L_{mlcs} always remains in the training by design, as the parameter value depends on the edit categories). In our case, we consider the classification and regression losses to be equally important in training and hence choose $\lambda = 1$. Experimentally, this also turns out to be the optimal balance between the two loss terms to achieve the best model performance. To highlight this, we report our observations for different metrics for $\lambda = 0.5$ (classification weighed twice as much as regression) and $\lambda = 2$ (regression weighed twice as much as classification) in Table 5

D.3 Additional Results.

We performed further experimental analysis on performing channel-wise concatenation of I_s and I_t as an alternative to computing δ_I (Eqn. 4). The results show an accuracy of 57%, RMSE of 0.332,

Table 5: **TAME-RD** with different λ . ResNet18 has been trained from scratch for all cases shown.

| | Data | Method | Evaluation Metrics | | | | |
|---|--|----------------------|--------------------|---------------|---------------|-------------------|----------------|
| | | | Acc. \uparrow | AP \uparrow | F1 \uparrow | RMSE \downarrow | CCC \uparrow |
| 1 | I-MAD-Pro (\sim 1K samples) | $\lambda = 0.5$ | 0.896 | 0.999 | 0.999 | 0.0919 | 0.884 |
| 2 | | $\lambda = 2$ | 0.867 | 0.563 | 0.984 | 0.0932 | 0.881 |
| 3 | | $\lambda = 1$ (Ours) | 0.896 | 0.999 | 0.999 | 0.090 | 0.882 |
| 4 | GIER (Shi et al., 2020) (\sim 6K samples) | $\lambda = 0.5$ | 0.717 | 0.909 | 0.841 | 4.52 | 0.42 |
| 5 | | $\lambda = 2$ | 0.715 | 0.907 | 0.844 | 4.513 | 0.416 |
| 6 | | $\lambda = 1$ (Ours) | 0.700 | 0.915 | 0.949 | 4.515 | 0.413 |

and CCC of 0.266. This indicates a drop in performance compared to learning features based on pixel-wise discrepancies. We also investigate whether, given the source image I_s and the text c , the edited image I_t is actually beneficial for predicting edit operations together with parameter values. The corresponding experiment yields an accuracy of 55%, RMSE of 0.343, and CCC of 0.197. These results establish that reverse designing is most effectively solved when both the source and the edited images are available. This also mirrors how humans cannot precisely deduce edit operations and parameters without observing the final edited image.