Language-based Photo Color Adjustment for Graphic Designs

ZHENWEI WANG*, City University of Hong Kong, China
NANXUAN ZHAO*, Adobe Research, USA
GERHARD HANCKE, City University of Hong Kong, China
RYNSON W.H. LAU, City University of Hong Kong, China

(a) Input (b) Recolored results -- case 1 (c) Recolored result -- case 2

Fig. 1. Language-based photo recoloring of a graphic design. Given a graphic design containing an inserted photo (a), our model recolors the photo automatically according to the given language-based instruction. To facilitate the expression of various user intentions, our model supports multi-granularity instructions for describing the source color(s) (selected from the design elements) and the region(s) to be modified (selected from the photo). For example, the user may provide a coarse-grained instruction ("background" in (b)) to refer to multiple source colors or a fine-grained instruction ("yellow shape" in (c), *i.e.,* using the yellow shape at the top left) to specify a source color. For visualization, we highlight the source colors predicted by our model at the bottom (colors inside the circles).

Adjusting the photo color to associate with some design elements is an essential way for a graphic design to effectively deliver its message and make it aesthetically pleasing. However, existing tools and previous works face a dilemma between the ease of use and level of expressiveness. To this end, we introduce an interactive language-based approach for photo recoloring, which provides an intuitive system that can assist both experts and novices on graphic design. Given a graphic design containing a photo that needs to be recolored, our model can predict the source colors and the target regions, and then recolor the target regions with the source colors based on the given language-based instruction. The multi-granularity of the instruction allows diverse user intentions. The proposed novel task faces several unique challenges, including: *1) color accuracy* for recoloring with exactly the same color from the target design element as specified by the user; *2) multi-granularity instructions* for parsing instructions correctly to generate a specific result or multiple plausible ones; and *3) locality* for recoloring in semantically meaningful local regions to preserve original image semantics. To address these challenges, we propose a model called *LangRecol* with two main components: the language-based source color prediction module and the semantic-palette-based photo recoloring module. We also introduce an approach for generating a synthetic graphic design dataset with instructions to enable model training. We evaluate our model via extensive experiments and user studies. We also discuss several practical applications, showing the effectiveness and practicality of our approach. Please find the code and data at https://zhenwwang.github.io/langrecol.

CCS Concepts: • **Computing methodologies** → *Computational photography*; **Computational photography**; **Image processing**; • **Human-centered computing** → **Text input**.

Additional Key Words and Phrases: data-driven graphic design, photo recoloring, language-guided

ACM Reference Format:
Zhenwei Wang, Nanxuan Zhao, Gerhard Hancke, and Rynson W.H. Lau. 2023. Language-based Photo Color Adjustment for Graphic Designs. *ACM Trans. Graph.* 42, 4, Article 101 (August 2023), 16 pages. https://doi.org/10.1145/3592111

*Both authors contributed equally to this research.

Authors' addresses: Zhenwei Wang, zhenwwang2-c@my.cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China; Nanxuan Zhao, nanxuanzhao@gmail.com, Adobe Research, California, USA; Gerhard Hancke, gp.hancke@cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China; Rynson W.H. Lau, rynson.lau@cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China.

0730-0301/2023/8-ART101 $15.00
https://doi.org/10.1145/3592111

1 INTRODUCTION

Graphic designs (*e.g.,* posters, webpages, slides and advertisements) have become a prevailing communication tool nowadays. As photos play an essential role in graphic designs, they are often recolored to associate and harmonize with other elements (*e.g.,* text, background and shape) in the graphic designs in real applications [GraphicsZoo 2020; Huang et al. 2018; Jordá-Albiñana et al. 2009; Mcguire 2019]. This helps the graphic design deliver the message effectively, engage viewers and evoke emotions. Prior works have been conducted to help automate this task [Cohen-Or et al. 2006; Kim and Suk 2018; Nguyen et al. 2017; Zhao et al. 2021].

However, when used by novices, existing methods and commercial software (*e.g.,* Photoshop, Affinity Photo, and GIMP) often face a dilemma between the ease of use and level of expressiveness. Commercial software allows users to indicate arbitrary color modifications, but requires them to have design knowledge and rich experience. The work of Cohen-Or *et al.* [2006] allows users to adjust the image color to match with the color of the context elements through different harmonic schemes, but it requires users to understand these schemes and it is often difficult for them to control the color change within the specified local regions. The framework from Nguyen *et al.* [2017] allows users to recolor a group of images based on the theme color of a design (*e.g.,* brochure), but the color can be changed in undesirable ways (*e.g.,* skin tone turns green) without considering the semantic contents of the images. Methods such as Kim and Suk [2018] and Zhao *et al.* [2021] allow users to recolor a photo with source colors from the design by a single click, but these methods can only generate deterministic results that may not be the desired outcome.

In this work, we aim to seek an interactive approach that is user friendly and has a broad range of expressiveness. We follow the basic setting of existing works [Kim and Suk 2018; Zhao et al. 2021] by recoloring the target local region(s) in the photo with a source color extracted from the design. However, we take advantage of the recent success of language-based interactions used in various computer vision tasks [Bahng et al. 2018; Chen et al. 2018; Jiang et al. 2021; Lüddecke and Ecker 2022; Ma et al. 2018; Weng et al. 2022; Zou et al. 2019], and present a language-based system for photo color adjustment in the context of graphic designs (Fig. 1). This system interaction comes natural to people and is intuitive to use. It can also be combined with voice input and is effective to the literacy education for children [Zou et al. 2019]. The system allows users to simply state their goals via concise verbal terms, without the need to learn a new interface or hunt through menus [Laput et al. 2013; Woolfe et al. 2007; Zou et al. 2019].

Although there are some works on language-based image editing and colorization, designing such a system specifically for our task has several unique challenges. 1) **Color Accuracy:** Rather than recoloring using an arbitrary color, the source color should be exactly the same as the one obtained from the design itself, as specified by the user. Existing works [Li et al. 2020a; Liu et al. 2020; Zou et al. 2019] that rely on specifying a vague color attribute (*e.g.,* "blue") with a wide range of possible mapped values may not be accurate enough. How to interpret the instruction to extract the source colors from the correct elements of the graphic design may not be straightforward. 2) **Multi-granularity Instructions:** To allow users to express their objectives in a diverse way, the input language-based instruction should be in multi-granularity, *i.e.,* from an obscure one (*e.g.,* "text") to a more specific one (*e.g.,* "orange text" or "subtitle"). The model needs to parse different instructions precisely, while some instructions may lead to multiple plausible results. 3) **Locality:** To preserve the image semantics, the color editing operation should be limited to local regions. The model needs to understand image content semantically and propagate the source color into the local regions naturally to produce a high-quality result.

To tackle the above three challenges, we present a novel language-based recoloring framework, called *LangRecol*, for photo color adjustment in the context of graphic designs. Given a graphic design with an inserted photo and a language-based instruction, our LangRecol system automatically adjusts the color of the photo following the instruction. The language-based instruction generally contains two parts, one for indicating where to obtain the source colors from the design elements, and the other for indicating where the target local regions are in the photo for color adjustment. The key idea of our framework is to take the language-based instruction as a bridge to model the relationship between graphic designs and the inserted photos in order to solve two main problems: predicting the specified source color(s) from the multi-granularity instruction by parsing the design correctly, and recoloring the target local region(s) of the photo semantically using the predicted source color(s).

To solve the first problem, we design a multi-task method to jointly conduct granularity recognition and source color prediction. The granularity recognition branch aims to recognize the granularity of the instruction to associate it with different types of target design element. As the color distributions of different types of element (*e.g.,* text and background colors) are different, our source color prediction branch is customized based on the type of target element. To facilitate model training, we propose an approach to synthesize a graphic design dataset following some basic graphic design principles [Aland 2017; ANL [n. d.]; Mcguire 2019] and knowledge obtained from real-world graphic designs. To solve the second problem, we design a semantic-palette-based method to constrain the color editing step to be within semantically meaningful local regions. To do this, we first predict an initial region mask of the user specified object(s) by parsing the input instruction with the image content. We then convert the initial region mask to refined coherent soft semantic color layers by leveraging both semantic features [Aksoy et al. 2018] and color features [Tan et al. 2018; Wang et al. 2019]. Finally, we obtain the recoloring results by changing the color of the target layers to the predicted source colors.

We evaluate the effectiveness of our method via extensive qualitative and quantitative experiments and user studies, which show that our pipeline outperforms existing automatic image color editing and language-based image color editing approaches. In summary, our main contributions are:

- We design a novel tool called, *LangRecol*, for photo color adjustment of graphic designs according to the multi-granularity language-based instruction.

- We propose a multi-task model for parsing graphic design elements and understanding multi-granularity instructions through granularity recognition, while predicting accurate source colors.
- We introduce a semantic-palette-based approach for language-based photo recoloring, predicting the target regions in a coarse-to-fine manner and generating high-fidelity recolored results.
- We develop a method to synthesize a plausible graphic design dataset based on real-world design knowledge and principles, to enable model training.

2 RELATED WORK

To the best of our knowledge, the work in [Zhao et al. 2021] has the most similar context with ours. While their goal is to generate deterministic recoloring results based on the user given colors, our work aims to adjust the photo color based on the input language-based instruction. This previous work needs explicit color assignment and limits the results to be within the regions predicted by the model without any user control. Instead, our work allows users to specify arbitrary target region(s) and source color(s) through an instruction. The rest of this section reviews prior works that are most related to ours.

Scribble-based Image Recoloring. Scribble-based approaches require users to draw color strokes to interactively provide local color hints for recoloring. The propagation from the local color strokes to all pixels is usually based on low-level similarity metrics by optimization [Huang et al. 2005; Levin et al. 2004; Li et al. 2008; Luan et al. 2007; Xu et al. 2009; Yatziv and Sapiro 2006] or learning [Endo et al. 2016]. Recently, deep learning approaches are introduced to support sparse color strokes [Zhang et al. 2017b] or to solve color-bleeding problems [Kim et al. 2021]. While these works can produce appealing results, they often rely on extensive strokes. Instead, our model can generate recoloring results with a single instruction, supporting editing on design collections.

Palette-based Image Recoloring. Palette-based approaches recolor images by manipulating the extracted color palette, which involves two critical problems: palette extraction and palette-based image decomposition. The palette of a given image can be extracted using k-means methods [Chang et al. 2015; Zhang et al. 2017a], convex hull simplification [Tan et al. 2018, 2016] or a physically-based method [Aharoni-Mack et al. 2017]. For image decomposition, the soft color layers with alpha channels can be extracted by optimization over the alpha blending model [Tan et al. 2016], additive color mixing models [Aksoy et al. 2017; Zhang et al. 2017a] or geometry-based method [Tan et al. 2018]. Recent deep-learning-based approaches [Afifi et al. 2021b; Wang et al. 2022] utilize a color palette or histogram features to globally control the colors of images generated by Generative Adversarial Network (GAN) models [Goodfellow et al. 2014]. Afifi *et al.* [2021a] uses a color palette to segment the image and apply color-aware multi-style transfer. Inspired by these methods, we propose a semantic-palette-based photo recoloring approach, taking both color continuity in the palette and semantic features into consideration, so that only the colors within the target region(s) specified by the instruction are adjusted properly.

Example-based Image Recoloring. Example-based approaches require users to provide reference images to guide the recoloring process. Traditional methods [Arbelot et al. 2017; Bae et al. 2006; Chang et al. 2005; HaCohen et al. 2013; Reinhard et al. 2001] transfer colors from a reference image to a target image based on low-level feature correspondences, without considering high-level semantic relationships. Recent deep-learning-based approaches [He et al. 2018; Luan et al. 2017] take the semantics of a scene into consideration when transferring the colors between image pairs. As users only need to provide a reference image, these methods can reduce user efforts. However, as they recolor the whole image based on this reference, without maintaining the necessary color features of non-modifiable regions, they are not suitable for solving our task.

Language-based Image Recoloring. Recent breakthroughs in NLP [Brown et al. 2020; Radford et al. 2021; Vaswani et al. 2017] show its power as a promising interaction paradigm for image editing, allowing natural and accessible control by succinct sentences or phrases as instructions. An increasing number of works relying on pretrained image generator and text encoder for general image editing tasks have been proposed [Nichol et al. 2021; Patashnik et al. 2021; Ramesh et al. 2022; Xia et al. 2021], with high-quality results. However, these works regenerate each pixel without preserving the original object structure, and thus cannot be used to address our problem. For image recoloring, early works [Cheng et al. 2014; Laput et al. 2013] take pre-defined instructions and semantic labels as inputs for rule-based control, which may not be easily adapted to practical scenarios. Recently, some works [Chen et al. 2018; Cheng et al. 2020; Zou et al. 2019] train GAN models with either paired data (*i.e.,* source images, language instructions and recolored images) or unpaired data [Dong et al. 2017; Nam et al. 2018; Zhu et al. 2017], but are applicable to only a specific domain, such as fashion, sketch, flower and bird.

Closer to our problem setting, there is another direction of works that focuses on dealing with images across different classes and editing the appearance-related attributes without changing the object structure. Li *et al.* [2020a] propose to selectively manipulate description-relevant regions, taking a trade-off between reconstruction and editing. To achieve both high-quality images and effective editing at the same time, Liu *et al.* [2020] recolor images by vector arithmetic over visual and textual features, and adopt a sample-specific optimization. Although these methods can semantically recolor parts of an image based on the input text description they often fail to constrain color propagation within local semantic regions. More recently, Khodadadeh *et al.* [2021] propose a GAN-based method to recolor images given predefined color and object tags, which can constrain color changes to a local region but suffers from the color bleeding problem with imperfect region masks. Kawar *et al.* [2022] propose a semantic image editing tool based on a pretrained text-to-image diffusion model [Saharia et al. 2022]. In addition, the vague color instructions used in all these works are not suitable to deal with our requirement of using the accurate source colors extracted from the design itself for recoloring.

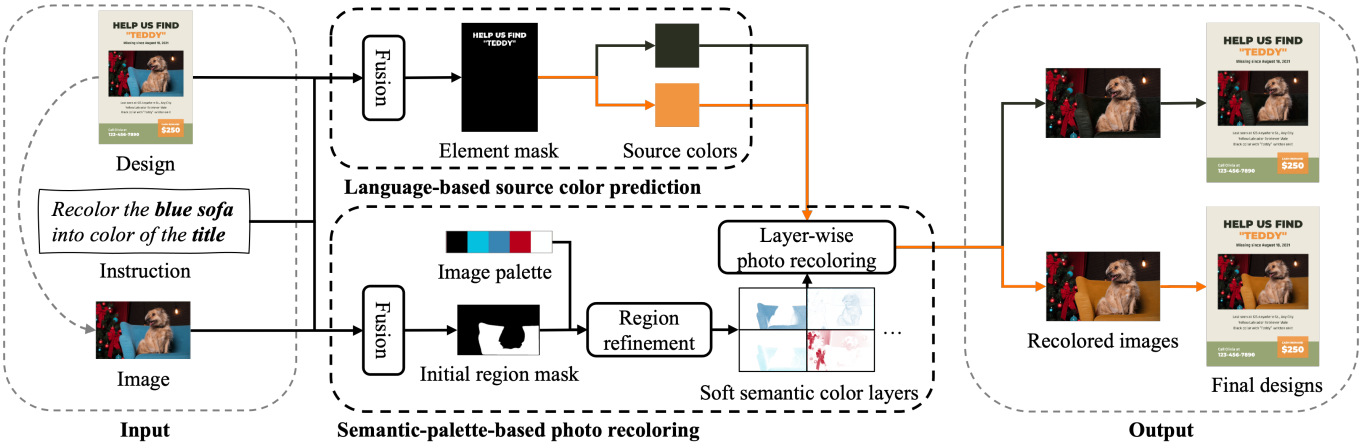


Fig. 2. Overview of our pipeline. There are two key components in our pipeline: (1) **language-based source color prediction** aims to predict the source colors from the user-specified design elements (*i.e.,* "title"); (2) **semantic-palette-based photo recoloring** aims to recolor the user-specified local regions (*i.e.,* "blue sofa") in the photo with the predicted source colors.

3 OVERVIEW

Our goal in this work is to construct a tool for recoloring the photos in graphic designs based on the given language-based instructions, which is accessible to even novices without any design experiences. We mainly focus on the single-page graphic design consisting of a small number of image, text and graphical elements, as it is widely used in our daily lives (*e.g.,* poster, advertisement and flyer), and studied in recent works [O'Donovan et al. 2014; Zhao et al. 2021]. As design elements are not always available in vector format (*e.g.,* a poster uses an inseparable image as background), we take pixel-format graphic designs as input. Given a pixel-format graphic design containing a photo that needs to be recolored, and a language-based instruction, our model parses the input instructions and extracts the source color(s) from the design elements while locating and recoloring the target region(s) in the photo.

We show the pipeline of our model in Fig. 2, which contains two key components: language-based source color prediction module and semantic-palette-based photo recoloring module. As a graphic design usually contains diverse elements in the form of a hierarchy (*e.g.,* a text element may cover title, subtitle, and plain text), the input instruction should support multi-granularity in order to identify the source color(s) properly. For example, the user may use a coarse-grained description, such as "background", to refer to all colors of a specific type of element, as shown in Fig. 1(b); and a fine-grained color description, such as "yellow shape", to refer to a specific color (*i.e.,* the color of the yellow shape, not the orange shape shown in Fig. 1(c)). The language-based source color prediction module aims to recognize the granularity and predict the source colors accordingly (see Sec. 5.1). However, as high-quality graphic designs require professionals to create, collecting a large-scale dataset with designs, instructions and source colors for training is impractical. To mitigate this problem, we introduce an approach to synthesize graphic designs for training (see Sec. 4). This allows us to obtain unlimited amount of data with ground truths easily.

To find the target regions in the image based on the instruction, there are two major requirements: semantically correct and locally compact. We take advantage of a large-scale dataset PhraseCut [Wu et al. 2020] with semantic labels and annotated segmentation masks for learning. However, their annotations are not accurate and are rough near the boundaries. Recoloring with such imperfect masks (*e.g.,* a straightforward way is to predict a binary mask and apply the recoloring network from [Zhao et al. 2021]) can generate severe and noticeable artifacts. Thus, we propose a coarse-to-fine method by taking this imperfect mask as an initial one, and then refining it by the *semantic-palette-based region refinement* process with the help of both deep semantic features and color continuity from palette extraction. (see Sec. 5.2).

4 SYNTHETIC GRAPHIC DESIGN DATASET WITH INSTRUCTIONS

In this section, we introduce a method to synthesize our graphic design dataset, containing pairs of instructions and source color(s). Before introducing the model details, we first define the basic design elements with their hierarchy. We then discuss how we synthesize different design elements with multi-granularity instructions.

Basic Design Elements. There are many types of elements in graphic designs. We start by collecting an initial list of frequently used design elements (except photos) from design books and blogs [ANL [n. d.]; Mcguire 2019; West 2020]. The initial list contains 16 types of design elements, covering three different categories: background, text (*e.g.,* title, tagline, branding and call-to-action), and shape (*e.g.,* logo, rectangle, polygon and line). To provide an accessible tool also for novices without any design experience, we remove elements that are not easily recognizable based on existing commercial tools

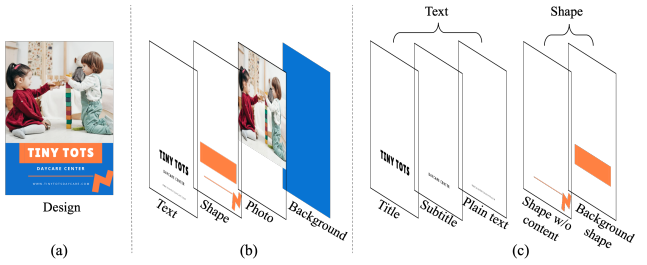


Fig. 3. Hierarchy of eight types of basic design elements of a design in (a). The hierarchy consists of (b) three main categories (except photo), *i.e.,* background, text, and shape, and (c) five fine-grained classes including title, subtitle, plain text, shape w/o content, and background shape.

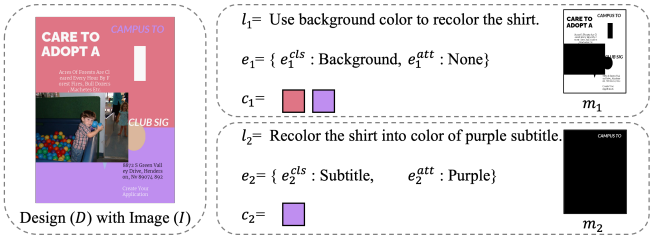


Fig. 4. Example of our synthetic graphic design dataset. For each design, we generate multiple language-based instructions referring to different target elements, with ground truth annotations (*e.g.,* color and mask).

(*e.g.,* Canva[1]) and a pilot study[2]. We obtain a final list of eight design elements with hierarchy, as shown in Fig. 3.

4.1 Design Elements Generation and Layout

Based on the above definition on design elements, we synthesize a design $D \in \mathbb{R}^{H\times W\times 3}$ by combining several design elements with a photo image I. For each synthetic design, we generate several language-based instructions $L = \{l_i\}$ that refer to different target elements $E = \{e_i\} = \{e_i^{cls}, e_i^{att}\}$ with class labels (*e.g.,* title) and color attributes (*e.g.,* blue). For each target element e_i, we denote its RGB color values as c_i and binary mask as m_i.

Rather than randomly generating and arranging design elements, we design our synthesis method based on both graphic design principles (*e.g.,* colors [Aland 2017; ANL [n. d.]; Mcguire 2019; West 2020] and fonts [Canva [n. d.]]), and design knowledge (*e.g.,* layout and appearance) summarized from real-world graphic designs on Canva. This can reduce the distribution gap between synthetic designs and the real ones, to better adapt our model to real design cases. An example of our synthetic graphic design dataset is shown in Fig. 4. Refer to the supplementary material for more details.

4.2 Multi-granularity Instruction Generation

The language-based instruction generally contains two parts: one for identifying the source color(s) from the design elements, and the other for locating the target region(s) in the inserted image. As for the first part, we categorize RGB color values into 11 color attributes (*i.e.,* blue, brown, green, orange, pink, purple, red, yellow, black, grey, and white) followed by a linguistic study on basic color terms [Berlin and Kay 1991]. We randomly assign either a specific element (*e.g.,* "subtitle", "blue shape") with its color as the ground truth, or a type of design element (*e.g.,* "all text") with multiple colors as the ground truth. We find that adding the color attribute of a design element to help indicate the specific source color is user-friendly and an effective way of mitigating ambiguity.

As for the second part, we use a large-scale dataset called PhraseCut [Wu et al. 2020], containing phrases of object regions with annotated segmentation masks. The multi-granularity phrases has already contained, such as "walking people", "black shirt", and "short deer". Similarly, we can use the object phrase directly (*e.g.,* "clothes") or in combination with its original color (*e.g.,* "blue clothes") for narrowing it down to a specific target region. The corresponding masks can be directly taken as the ground truths. Finally, we use the predefined instruction patterns, such as "recolor [region description] into [color description]" or "use [color description] to recolor [region description]", to combine the region and color descriptions into a complete instruction. Note that the order of these two parts can be random to allow more free-form instructions during testing. The conjunction words are randomly selected from a list of candidate words (*e.g.,* use "edit/change/adjust the color of" to replace "recolor"), to mimic diverse human expression habits. Refer to the supplementary material for more details.

5 OUR APPROACH

Given a design D with an inserted image I, LangRecol generates a recolored image I' according to the input language-based instruction l. As shown in Fig. 2, we first encode the design and the image separately using a ResNet101 [Zagoruyko and Komodakis 2016], and the instruction using Bi-GRU [Cho et al. 2014], to extract their features. These features are then fused into two different groups before being sent to the two main modules of our framework. More specifically, the features derived from the design and the instruction are fused for language-based source color prediction, while the features derived from the image and the instruction are fused for semantic-palette-based photo recoloring.

We use an encoder fusion operation $\mathcal{F}$, which progressively inserts the features of the instruction into the last three res-blocks through a co-attention module. This has been validated as an effective operation for fusing multi-modal features in multi-level without increasing additional computational burden [Feng et al. 2021]. We denote the fusion features from the design and the instruction as f_{dl} and fusion features from the image and the instruction as f_{il}: $f_{dl} = \mathcal{F}(ResNet(D), BiGRU(l))$, $f_{il} = \mathcal{F}(ResNet(I), BiGRU(l))$.

5.1 Language-based Source Color Prediction Module

Given the fusion features f_{dl}, we aim to predict the source color(s) from specified design elements. As there may be more than one source color, we design the model to predict a set of colors $\{c_i\}$ with confidence scores $\{s_i\} \in [0, 1]$, and we only consider the predicted colors with high confidence (*e.g.,* > 0.5) as the source

[1]https://www.canva.com/templates/

[2]We invited several graduate students from a local Computer Science department for this pilot study. We asked them whether they knew the meaning of each design element type and whether they would use it in language-based instructions.

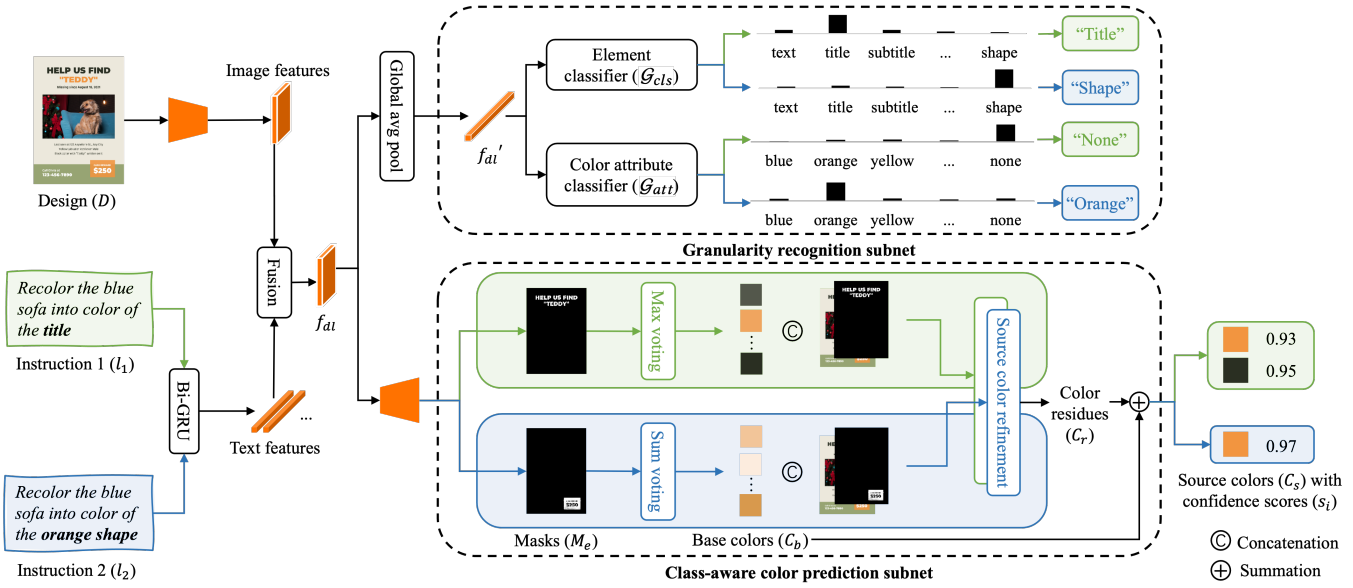


Fig. 5. Language-based source color prediction module. After fusing the features from the design and the instruction, the model conducts multiple tasks by recognizing the granularity of the instruction while predicting the source color(s) with confidence score(s). To better visualize the class-aware manner of our color prediction, we show two instructions referring to different types of target elements (*i.e.,* green branch for text-based elements, and blue branch for filled-color-based elements). During inference, which branch the model takes depended on the class type predicted by the element classifier above.

colors C_s during test time. Taking the challenges of multi-granularity in instructions and diverse appearances among design elements, we design a multi-task network for source color prediction, as shown in Fig. 5. It contains two sub-networks: the granularity recognition subnet, and the class-aware color prediction subnet. The element class type predicted by the granularity recognition subnet would help determine the way of color prediction in the class-aware color prediction subnet.

5.1.1 Granularity Recognition Subnet. This subnet aims to distinguish the granularity of the instructions. As the type and color attribute of elements determine the granularity of the instruction on the source color(s), we convert the problem into two classification tasks by recognizing the element class and color attribute of the target design element in the instruction. Note that the color attribute can be "None" as it is an optional choice. Formally, given the fusion features f_{dl}, we first pass them through a global average pooling layer (GAP) for flattening into 1D features $f_{dl}{}' = \text{GAP}\,(f_{dl})$, and then send them into two different classifiers $\mathcal{G}_{cls}$ and $\mathcal{G}_{att}$ to obtain:

$$p_{cls} = \mathcal{G}_{cls}\,(f_{dl}{}'), \quad p_{att} = \mathcal{G}_{att}\,(f_{dl}{}'), \tag{1}$$

where p_{cls} and p_{att} are the predicted probabilities of element class and color attribute, respectively.

5.1.2 Class-aware Color Prediction Subnet. Directly predicting the source colors in RGB values is not adequate as a slight change of RGB values can cause a huge difference in visual perception, which contradicts to our goal of extracting accurate color(s) from the design element. We thus design an approach by voting from the colors within the regions of the target element. We obtain the region mask of element e, $M_e \in \mathbb{R}^{H\times W\times 1} \in [0,1]$, through an element segmentation subnet, where a higher value indicates a higher probability that the RGB value of a pixel is the source color of the specified element.

Before conducting the voting, we observe that elements with distinct appearances can perform very differently. Text elements including titles, subtitles and plain text have string-like appearances, often overlapped with background colors and shapes, and their colors tend to be distributed in a sparse way. In contrast, a background color or shape often covers a large area of pure color in a filled region, and the color is distributed in a compact way. As such, we take different actions based on the class type: text-based elements, or filled-color-based elements. We introduce a class-aware voting strategy to obtain k base colors $C_b \in \mathbb{R}^{k\times 3}$ from the input design.

We first take the unique colors within the design pixels whose values in the predicted masks are larger than an adaptive threshold v (*i.e.,* the mean value of the element mask) to form the initial candidates of the base colors. As there may be many pixels with very similar colors (*e.g.,* noisy and gradient colors caused by image compression) and taking all these colors as candidates can make training very slow, we filter out pixels of similar colors and assign only unique colors into $b \times b \times b$ uniform histogram bins ($b = 6$) to form the final candidate sets. We then adopt max voting for text-based elements and sum voting for filled-color-based elements according to the predicted class type in Sec. 5.1.1. We regard the top-k unique colors ($k = 10$) as the base colors. More details can be found in the supplementary material.

We predict the final source colors by computing a color residual for each of the base colors. We represent each base color as a pure

color map $B_i \in \mathbb{R}^{H\times W\times 3}, i \in [1,k]$, and concatenate it with the element mask M_e and input design D. Based on this concatenated vector, we build a source color refinement subnet to obtain the color residuals C_r and confidence scores s_i. The final source colors C_s are computed as $C_b + C_r$. Note that the source color refinement subnet is also class-aware, and that we train two separate versions for text-based elements and filled-color-based elements (*i.e.,* same architecture but with different parameters).

5.1.3 Loss Functions. Our loss contains the following terms: *Granularity Loss.* We use the cross entropy loss to compute the predicted probabilities with the ground truth $\hat{p_{cls}}, \hat{p_{att}}$ defined as:

$$L_g = \mathrm{CE}(\hat{p_{cls}}, p_{cls}) + \mathrm{CE}(\hat{p_{att}}, p_{att}). \tag{2}$$

Segmentation Loss. We adopt the binary cross entropy loss to measure the difference between the predicted element mask M_e and the ground truth binary mask $\hat{M_e}$ as:

$$L_m = \hat{M_e} \log M_e + \left(1 - \hat{M_e}\right) \log\left(1 - M_e\right). \tag{3}$$

Source Color Loss. For source color prediction, we use a L_2 loss function to minimize the distance between the ground truth and predicted source colors, which is defined as:

$$L_c\left(\hat{C_s}, C_s\right) = \left\|\hat{C_s} - C_s\right\|_2^2. \tag{4}$$

Confidence Loss. We adopt a binary cross entropy loss for predicting the confidence scores of the source colors. We take the confidences of ground truth source colors as 1, and the others as 0, and define the loss term as:

$$L_s = \hat{s} log s + (1 - \hat{s}) log(1 - s). \tag{5}$$

In summary, the overall multi-task loss can be formulated as:

$$L = \lambda_1 L_g + \lambda_2 L_m + \lambda_3 L_c + \lambda_4 L_s, \tag{6}$$

where λ_i are coefficients to balance the loss terms.

5.2 Semantic-palette-based Photo Recoloring Module

Our objective of this module is to predict target region(s) based on the fused features f_{il}, and recolor the image within the predicted region(s) to obtain a visually natural and pleasing result. However, simply relying on a deep neural network to learn semantics and predict the region for recoloring [Zhao et al. 2021] can generate serious artifacts as we do not have large-scale well-annotated language-based segmentation dataset. Inspired by the recent success of palette-based recoloring methods [Chang et al. 2015; Tan et al. 2018, 2016], we take a hybrid way by combining the semantics-aware deep learning method and palette-based recoloring. The key idea is to mitigate the problem of imperfect region mask prediction by the network with color continuity of palette-based methods.

As shown in Fig. 6, the module takes several steps: (1) generating initial region mask; (2) refining the initial region based on both semantic features from the deep network and soft color layers from the palette; (3) recoloring the image in a layer-wise manner. To generate the initial region mask $M_r^{init} \in \mathbb{R}^{H\times W\times 1}$ and acquire the semantic knowledge, we train a language-based image segmentation subnet, sharing the same network architecture as the one in Sec. 5.1.2. The initial region mask is a hard one with binary values $M_r^{init} \in \{0,1\}$. We also obtain a color palette $\mathscr{P} \in \mathbb{R}^{N\times 3}$ using a geometry-based method [Tan et al. 2018, 2016] for the inserted image. We then show how we refine the initial mask based on the extracted palette and how we perform layer-wise photo recoloring.

5.2.1 Semantic-palette-based Region Refinement. This step aims to decompose the whole image into several soft semantic color layers, where each layer is a compact local region sharing the same semantics and color. We take a coarse to fine way to obtain the final layers. For semantics, we only distinguish between the target object (referred to as "foreground" in our context) and the other regions (as "background"). Given the initial region mask M_r^{init}, we first compute the refined soft background and foreground region masks $M_r^{soft} = \{m_f, m_b\} \in [0,1]$ through a semantic soft segmentation method S [Aksoy et al. 2018]. This method leverages texture, color, and semantic cues for conducting image soft segmentation. We set the number of target segments to two and regard our initial region mask as the semantic feature vector for computing the semantic affinity term in $\mathcal{S}$. This can encourage the model to group pixels belonging to the user specified target region(s) together. Formally,

$$M_r^{soft} = \{m_f, m_b\} = \mathcal{S}(M_r^{init}, I). \tag{7}$$

The palette-based image decomposition method can generate a set of soft color layers $\mathscr{L} = \{\ell_1, \ell_2, ..., \ell_N\}, \ell_i \in \mathbb{R}^{H\times W\times 4}$, each corresponding to a value in the color palette $\mathscr{P}$ with an opacity value $\alpha^p \in [0,1]$ for each pixel. The original image can then be restored by summing these soft color layers $I = \sum_i \ell_i$ and $\sum_i \alpha_i^p = 1$. Finally, we derive the soft semantic color layers $\mathscr{L}' = \{\ell'_1, \ell'_2, ..., \ell'_Q\}$ by multiplying the soft color layers $\mathscr{L}$ and refined soft region masks M_r^{soft} in a pairwise way (*i.e.,* $Q = 2N$), as shown in Fig. 6.

5.2.2 Layer-wise Photo Recoloring. With the benefit of linearly decomposing the input image as soft semantic color layers, we can directly recolor the image by changing the color of the target layers given the source color. We select the target layers based on the foreground soft region mask m_f. Specifically, we compute region overlap rates $O = \{o_1, o_2, ..., o_Q\}$ between the opacity value α_i of each soft semantic color layer and the foreground soft region mask m_f as:

$$o_i = \frac{\sum_p(\alpha_i \odot m_f)}{\sum_p(\alpha_i)}, \tag{8}$$

where $\sum_p(\cdot)$ denotes pixel-wise summation and $\odot$ denotes pixel-wise multiplication. As a larger o_i indicates a more dominant layer that contributes more color information within the target regions, we select the top-n layers ($n = 4$, empirically) as the target layers $\mathscr{L}'_t = \{\ell'_{t1}, \ell'_{t2}, ..., \ell'_{tn}\}$, where ℓ'_{t1} is the most dominant layer, and modify the color of these layers based on the source color C_s.

To preserve the original texture gradient in the target regions, instead of copying the source color to these layers directly, we follow the change of *lightness* (*i.e.,* the $\mathbb{L}$-channel of CIE Lab color space, which is more aligned with human perception) in the original regions. We define the original *lightness difference* between the i-th target layer ℓ'_{ti} and the most dominant layer ℓ'_{t1} as: $\Delta\mathbb{L}_i = \mathbb{L}_{ti} - \mathbb{L}_{t1}$. After assigning the source color to each of the target layers, we alter the *lightness* by adding back the above *lightness difference*

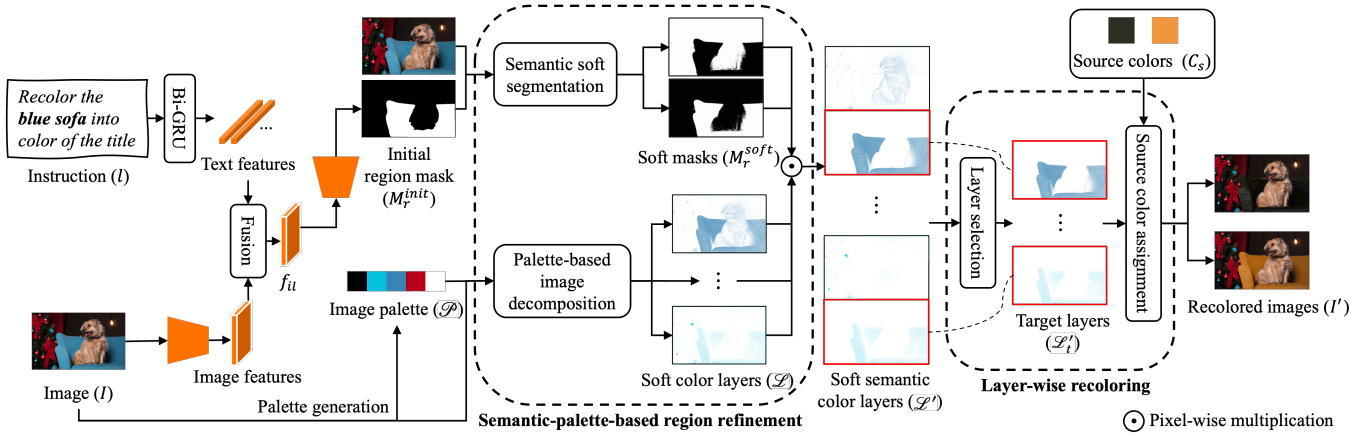


Fig. 6. Semantic-palette-based photo recoloring module. This module first predicts the target region(s) in the form of soft semantic color layers by taking advantages of both deep semantic features and color continuity from palette extraction. It then performs photo recoloring in a layer-wise manner by assigning the source color(s) to each of the selected target layers (*i.e.,* boxes marked in red).

$\hat{\mathbb{L}}_i = \mathbb{L}_{C_s} + \Delta\mathbb{L}_i$. By combining with unmodified layers, we obtain the final recolored image.

6 EXPERIMENTS

We show our language-based photo recoloring results on various graphic designs with diverse appearances in Fig. 7. We can see that our model can follow the instruction for recoloring, obtaining natural and pleasing results. The designs also look more visually harmonious after the photo color adjustment. In addition, our model allows users to specify different source colors and target regions through instructions to obtain multiple results (*e.g.,* rightmost column of Fig. 7). As shown in Fig. 1 and Fig. 8, if a coarse-granularity instruction (*e.g.,* "color of the shape") specifies multiple source colors, our model can also correctly generate multiple recoloring results. For the rest of this section, we first compare our results with those generated by the state-of-the-art methods and professional designers. We then conduct experiments to analyze the effect of different important design choices of our model.

Implementation details. We train the language-based source color prediction module (in Sec. 5.1) on the synthetic dataset (in Sec. 4) and the language-based image segmentation subnet (in Sec. 5.2) on the PhraseCut dataset. When training the language-based source color prediction module, we use a curriculum training strategy by pretraining the element segmentation subnet and the granularity recognition subnet first to avoid model collapse, and then train the whole model jointly. All the training and test processes are conducted on a PC with i7-10700 CPU and a single RTX3080 GPU. More training details can be found in the supplementary material.

Test Set. We have collected 71 realistic graphic designs from Canva, and assign several diverse language-based instructions to each design to obtain a test set with 165 design-instruction pairs. These collected graphic designs cover a broad range of themes and styles, such as event posters, business advertisements, holiday flyers, etc. The inserted photos also span a wide range, from indoors to outdoors, from street to fashion, and from food, human to animal. As there are no ground truth recoloring results, we asked three professional designers to manually extract the source colors and recolor the inserted photos following the instructions. We let them to use any tools that they preferred, *e.g.,* Photoshop. After two designers have finished all designs, the third designer chooses a preferred one in each design case as the ground-truth source colors and recoloring results.

6.1 Comparison with Exiting Methods

6.1.1 Baselines. There are no existing methods supporting language-based photo recoloring in graphic designs. We thus adapt several state-of-the-art language-based image editing methods in order for them to be applicable to our task:

- Four state-of-the-art text-guided image editing methods: Open-edit [Liu et al. 2020], ManiGAN [Li et al. 2020a], RecolGAN [Khodadadeh et al. 2021] and Imagic [Kawar et al. 2022], which support structure-preserved recoloring. To fit our problem, we manually convert our instructions to the prompts in a format suitable for these methods. For example, we use "blue sofa" for ManiGAN, "red sofa -> blue sofa" for Open-edit, "recolor flower to orange" for RecolGAN, and "a photo of a blue car" for Imagic.
- A state-of-the-art work for photo recoloring in graphic design [Zhao et al. 2021] (GDRecolor). This method directly predicts modifiable regions based on a specified color, and recolors these regions in an inpainting manner. As this method cannot process language-based input, we directly take the source colors predicted by our model to obtain its results.

6.1.2 Qualitative Results. We show the qualitative results in Fig. 9. We can see that even though the images are recolored following the given instructions to some extent, the two language-based baselines fail to produce high-quality recoloring results. ManiGAN tends to perform global color editing (*e.g.,* 1^{st} and 3^{rd} rows) and sometimes produces nearly unmodified results (*e.g.,* 2^{nd} and 4^{th} rows). This is mainly because of the tradeoff between image manipulation

Fig. 7. Our language-based photo recoloring results on graphic designs. For each design case, we show the original design on the left and our result(s) on the right, with the predicted source color and the input instruction below. We highlight the words related to the source colors and target regions in bold.

Fig. 8. Our recoloring results with coarse-granularity instructions. Left-bottom corner of each design case shows the predicted source colors.

and reconstruction during the ManiGAN training. While Open-edit can limit the editing to the target regions locally, it may generate unnatural contents. As the image editing is conducted in the visual-semantic embedding space, local details may be lost in the generated results (*e.g.*, human face in the 1^{st} and 4^{th} rows). Besides, by using the color words, *e.g.*, "red" and "green", their source colors may not be accurate enough, failing to achieve a more visually harmonious design (*e.g.*, hat in the 4^{th} row). Though GDRecolor can recolor in a local region, producing more natural and high-quality results, their method fails to follow the instructions and is less controllable. Users can only adjust the color in the model-predicted regions. For example, in the 3^{rd} row, with the green as the source color, the model prefers to adjust the grass color rather than the roof specified by the user. In contrast, our method allows flexible target region specification via instructions. Besides, our model can parse multi-granularity instructions efficiently by predicting accurate source colors and recoloring the image naturally, resulting in more aesthetically pleasing designs. Due to the unavailable code and the limited space, you may refer to the comparisons with RecolGAN and Imagic in the supplementary material.

Table 1. Recoloring results compared with language-based methods.

Methods	PSNR ↑	SSIM ↑
Open-Edit [Liu et al. 2020]	15.55	0.5230
ManiGAN [Li et al. 2020b]	16.91	0.6662
Ours	**24.27**	**0.8500**

6.1.3 Quantitative Results. We show the quantitative results in Tab. 1 on two different metrics: the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity index (SSIM). These metrics are commonly used in image editing and generation tasks [Bau et al. 2020; Zhang et al. 2017b]. Since GDRecolor cannot follow the instruction for locating the target regions, the quantitative comparison with this method becomes meaningless. We thus only compare with language-based baselines in the quantitative experiment as well as the user study to be introduced next. The results show that our model outperforms the baselines by a large margin on both metrics, demonstrating that our model can generate more faithful recoloring results.

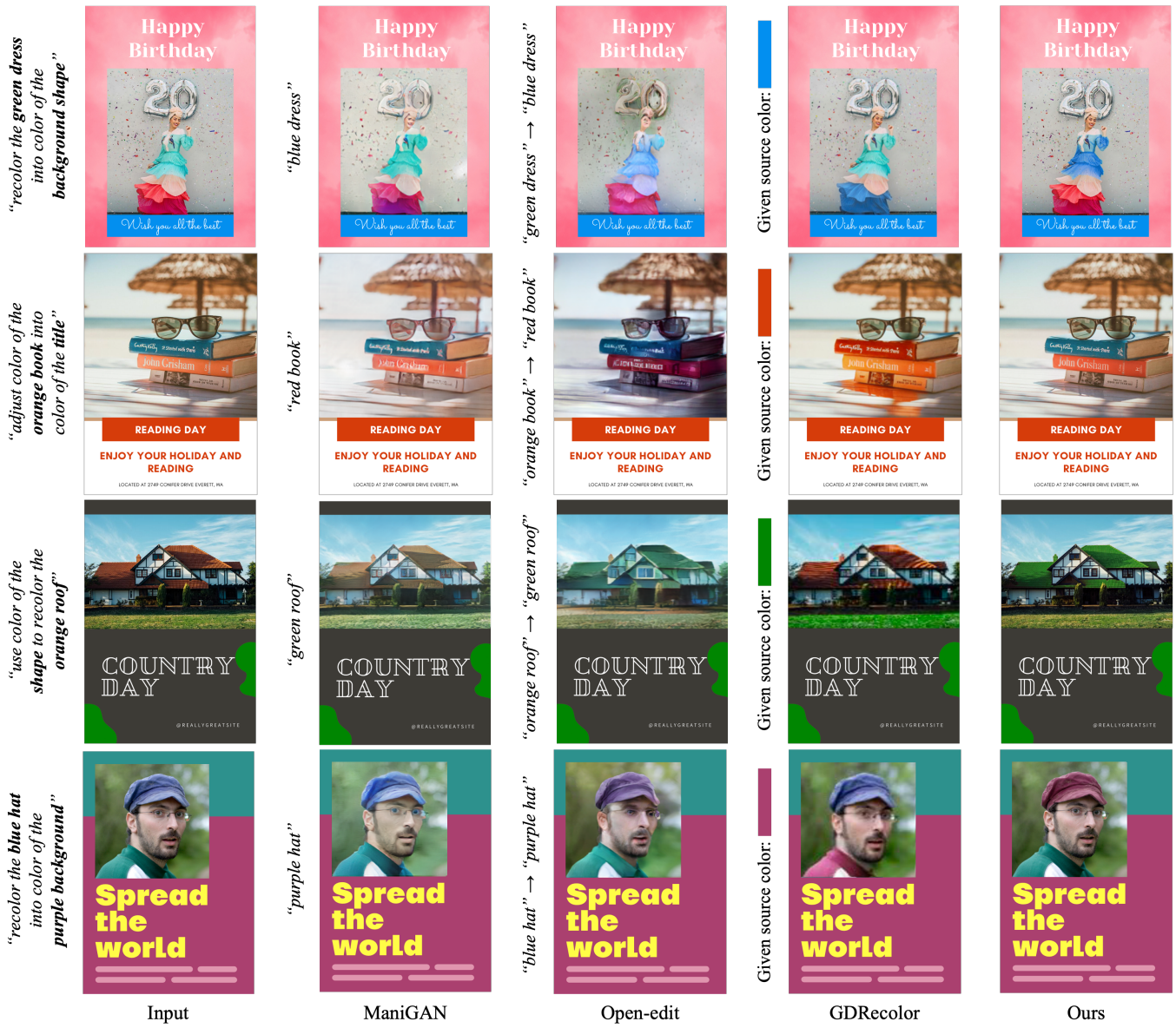


Fig. 9. Comparison results with state-of-the-art methods. We compare with two language-based image recoloring methods: Open-edit [Liu et al. 2020] and ManiGAN [Li et al. 2020b]; and a graphic design photo recoloring method GDRecolor [Zhao et al. 2021].

6.1.4 User Study. Since the metrics used above mainly measure the similarity between the results and ground truth, we further conduct a user study to validate our approach according to the naturalness of the recoloring results, and the faithfulness of the results to the instructions. We randomly select a set of design-instruction pairs (*i.e.,* 68) from our test set. Participants were invited to complete a questionnaire consisting of 30 pairwise comparisons in person, using the same color-calibrated screen. There are a total of 35 participants in our study, recruited from a local Computer Science department. The results of this user study are presented in Tab. 2. We can see that our model significantly outperforms existing language-based image recoloring methods in both naturalness and faithfulness. Given the challenging task of parsing language-based instructions in the graphic design context, there is still a gap between our results and those generated by designers. More details of the user study can be found in the supplementary material.

6.2 Effect of the Language-based Source Color Prediction Module

To evaluate the effectiveness of the two main subnets in the language-based source color prediction module, we design two variants as:

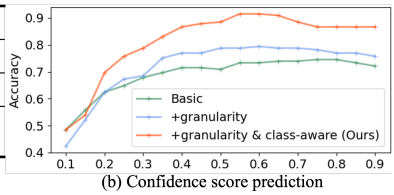
- Ours without granularity recognition subnet (**Basic**). We use uniform mean voting for both text elements and filled-color-based

Table 2. Results of the user study. We compare our method (Ours) with either designers or previous methods using 2AFC pairwise comparisons. All preferences are statistically significant ($p < 0.05$, chi-squared test).

vs. Methods	% Preferred Ours	
	Naturalness	Faithfulness
Open-Edit [Liu et al. 2020]	82.2%	83.0%
ManiGAN [Li et al. 2020b]	69.6%	82.6%
Designers	42.2%	36.5%

Models	MSE
Basic	3.719×10^{-3}
+granularity	1.691×10^{-3}
+granularity & class-aware (Ours)	0.540×10^{-3}

(a) Source color prediction



(b) Confidence score prediction

Fig. 10. The effect of the granularity recognition subnet (granularity) and the class-aware color prediction subnet (class-aware).

elements. Besides, we only train a single class-agnostic source color refinement subnet.

- Ours without class-aware color prediction subnet (**+granularity**). We replace the class-aware voting strategy with the mean voting for all element types. The other components, including the granularity recognition subnet, remain the same as our final model.

6.2.1 Quantitative Results. We evaluate the predicted source colors and confidence scores using different metrics. For source colors, we calculate the mean square error (MSE) between the predictions and ground truth. The results are shown in Fig. 10(a). We can see that our complete language-based source color prediction module achieves the best performance with the smallest MSE value, showing that the source colors predicted by our method are very close to the ground truth (*i.e.,* an average error of ±5.95 for each of the RGB values $\in [0, 255]$). The better performance of *+granularity* over the *Basic* variant demonstrates that jointly training with the granularity recognition task can enhance the model's ability on parsing design context based on the instructions. Besides, further incorporating a class-aware color prediction subnet can help increase the performance, leading to the best result, which is our final model.

The confidence scores determine the number of source colors extracted based on the instructions. We regard the confidence prediction as a correct one if its corresponding predicted source color matches with a ground truth source color while the confidence score is larger than a threshold. As this threshold for recognizing correctness influences the performance, we compute the accuracy under different thresholds, and show the result in Fig. 10(b). We can see that the absence of either the class-aware color prediction subnet or both subnets can significantly affect the accuracy w.r.t the confidence scores of the predicted source colors. A threshold of 0.55 for the confidence score achieves the best performance on our test set. Refer to the supplementary material for more details on the evaluation of the confidence scores.

6.2.2 Qualitative Results. To further understand what the language-based source color prediction module actually learns, we qualitatively compare our method with the two variants and show the results in Fig. 11.

As can be seen, without the help of the granularity recognition subnet, the model fails to distinguish between coarse-grained and fine-grained instructions, and locates wrong or inaccurate design elements. For example, in the 1^{st} row, the *Basic* model fails to locate the accurate partial background region (*e.g.,* "blue background" and "green background") but predicts the whole background, and predicts the subtitle for "yellow text" in the $3rd$ example by mistake. Regarding the source color prediction, without the class-aware color prediction subnet, the model can generate noisy background color (*e.g.,* "blue plain text" in the 2^{nd} row) for text elements, and noisy color in the transition region (*e.g.,* "orange shape" in the 2^{nd} row) for shape elements. Instead, our full model can accurately predict both design element masks and source colors for various instructions.

6.3 Effect of the Semantic-palette-based Photo Recoloring Module

6.3.1 Baselines. We compare our semantic-palette-based photo recoloring module with several baselines on image recoloring based on the given source colors: scribble-based recoloring [Levin et al. 2004]; deep user-guided recoloring [Zhang et al. 2017b] (with Sparse and Dense modes); semantic-based photo recoloring [Zhao et al. 2021] (*i.e.,* an ablation variant of our model without the semantic-palette-based region refinement); Palette-based photo recoloring [Tan et al. 2018]. More details can be found in the supplementary material.

6.3.2 Qualitative Results. We show the qualitative comparison in Fig. 12. Though the scribble-based, deep user-guided, and semantic-based methods [Levin et al. 2004; Zhang et al. 2017b; Zhao et al. 2021] can propagate the source color in the local region, they generate explicit artifacts because of the imperfect target region, such as color bleeding near the red T-shirt in the 2^{nd} row. Note that the scribbles and hint points of the compared methods [Levin et al. 2004; Zhang et al. 2017b] may need to be carefully drawn in order to obtain high-quality results. Besides, the color of the resulting images may not follow the source color for scribble-based and semantic-based methods (*e.g.,* dress incorrectly colored in dark rose in the 3^{rd} row). This may be because these methods rely on Lab color space by maintaining the L-channel to be the same as the input, leading to misaligned results if the values of the L-channel are very different between the source color and the original color. As for the palette-based method, without the constraint on the local region, it can generate unnatural results, such as the skin in the 1^{st} row. Our approach can recolor the image properly and naturally even the initial target region mask is not compact nor complete.

6.4 Effect of Using the Language-based Input Modality

We conduct a usability user study [O'Donovan et al. 2015; Xiao et al. 2021] to further evaluate the effectiveness of our language-based input modality in helping novices recolor photos in the context of graphic designs.

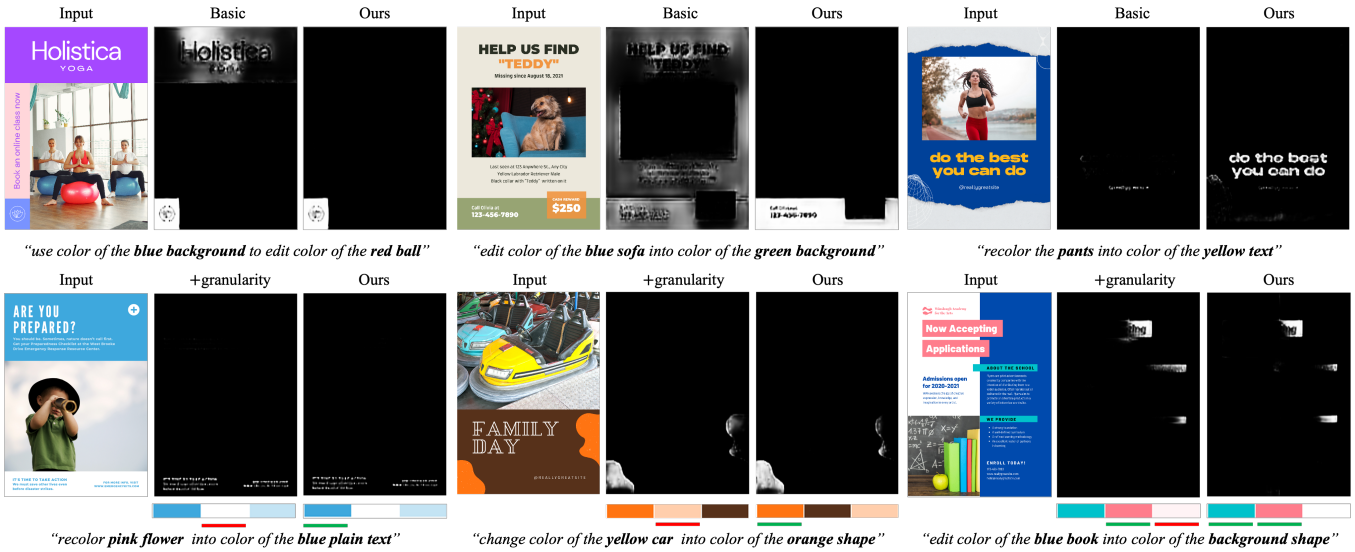


Fig. 11. What does our language-based source color prediction module learn? We compare our model with two variants: *Basic* (without granularity recognition nor class-aware subnets in the 1^{st} row) and + *granularity* (without class-aware color prediction subnet in the 2^{nd} row). We show the predicted element mask in each case and predicted colors only for the examples in the 2^{nd} row. The bar below the color indicates its confidence score is larger than 0.5 with green for a correct prediction and red for a wrong one. Please zoom in for details.

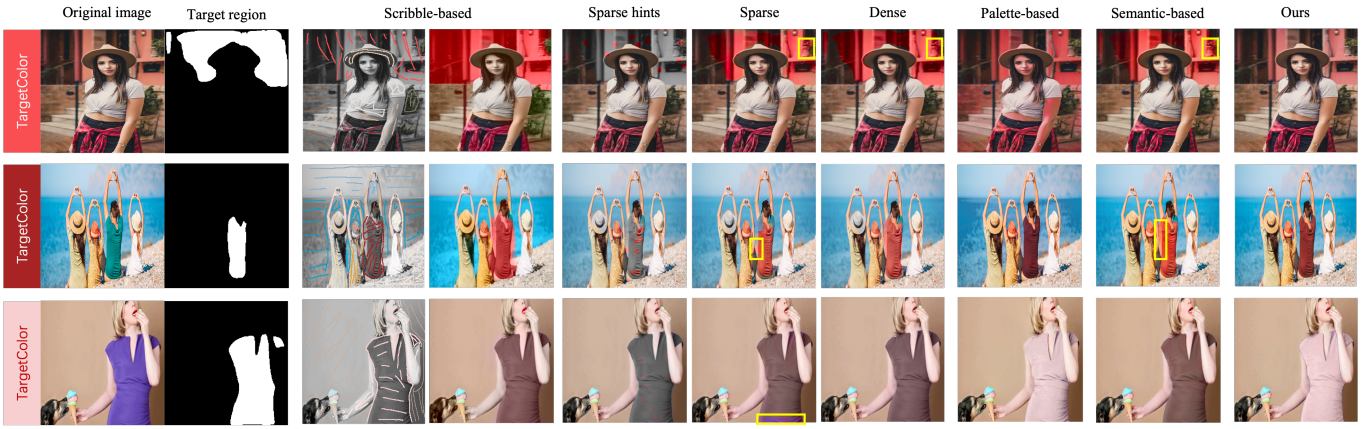


Fig. 12. Comparison with photo recoloring methods based on target regions predicted by our model. We compare with scribble-based method [Levin et al. 2004], deep user-guided method [Zhang et al. 2017b], semantic-based method [Zhao et al. 2021], and palette-based method [Tan et al. 2018].

6.4.1 Setup. We invited ten novice users to test two interfaces: 1) Language-based interface of our *LangRecol* framework (**Ours**), as shown in Fig. 13, and 2) a traditional scribble-based interface without language as input (**Baseline**). **Baseline** uses the same interface layout as **Ours** but language inputs (b) is disabled. Users need to manually draw the initial rough region mask and select source colors via color pickers. For fair comparisons, both interfaces adopt our semantic-palette-based photo recoloring module for producing results. Users took a 15-min tutorial before the real tasks. For each user, we randomly selected 5 design-instruction pairs from the test set as the recoloring task and asked the user to recolor following the given instructions. The order that users test the two interfaces is set to random, and the time usage for each design case is recorded. After conducting the design tasks, users were asked to provide a five-point System Usability Scale (SUS, 1 = strongly disagree to 5 = strongly agree) on four different perspectives each interface. More details can be found in the supplementary material.

6.4.2 Results. The average rating scores for different perspectives and the statistics of interaction time are shown in Fig. 14. As can be seen, our interface is rated significantly higher than the baseline interface for most perspectives except for faithfulness, as the fully human-guided results should be most consistent with human intentions. For the interaction time, our language-based input can help

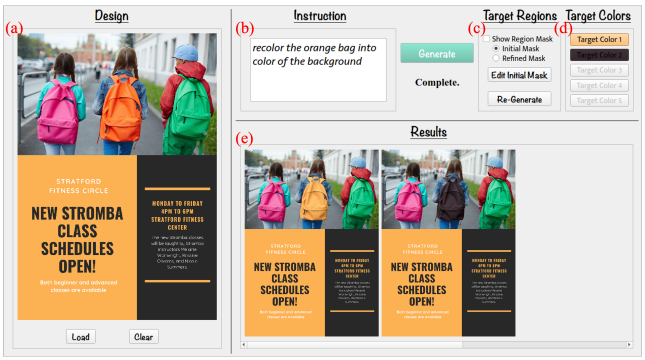


Fig. 13. Our interface. (a) is the input design with a photo. (b) is the input language-based instruction. (c) and (d) display target region and source color predictions. (e) shows the recoloring results based on the instruction in (b).

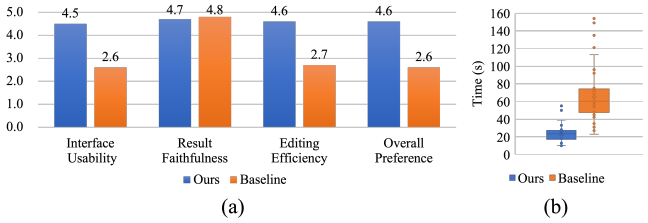


Fig. 14. Results of the usability study. (a) is the users' average rating score for two interfaces from different perspectives, including usability of the interface, faithfulness of the results to the task instruction, efficiency of finishing an editing task, and overall preference of the interface. (b) is a box plot showing the statistics of the time usage for different design cases.

save 43s (63.70% of the time) per example on average, which indicates the effectiveness of our language-based interface. Users also provided many positive feedback on our interface, such as "This interface is really cool as I can create diverse (recoloring) designs with only a single click" and "I enjoy this creative interface, where I can directly tell the system what I want via typing or even voice in the future". Additional user feedback can be found in the supplementary material.

7 APPLICATIONS

We depict several practical applications enabled by our framework.

Design Template Pairing. As the graphic design community grows, there are many well-curated online design template repositories. It is quite often that users need to find a good matching template for their given photo, such as an event poster. However, directly adjusting the photo color based on each individual template is tedious and it may take a long time to find the proper one. With our model, users can examine a set of templates by automatically recoloring the image to match with each of the templates with a single instruction, as shown in Fig. 15. This allows designers to brainstorm ideas, and novices to finalize their template in an efficient way.

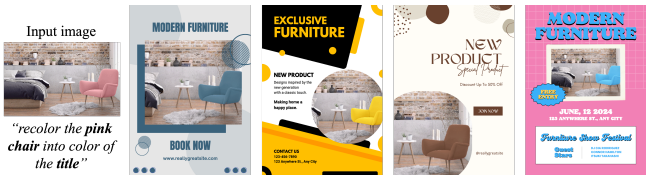


Fig. 15. Design template pairing. Our model can adjust photo color based on a set of design templates for quick brainstorm and authoring.

Fig. 16. Brochure Photo Recoloring. Our model can recolor multiple photos with a single design.

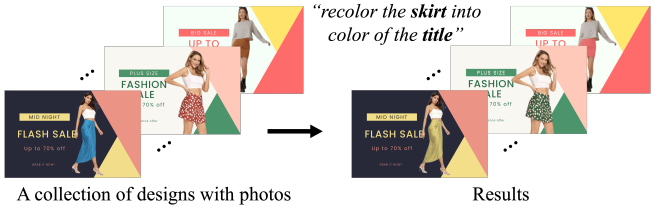


Fig. 17. Recoloring a design collection. We show a design case for banners used in online promotion.

Brochure Photo Recoloring. Some types of graphic design may contain more than one photo, *e.g.*, brochure. Our model can also deal with this case by recoloring multiple inserted photos together with a single instruction. As the photos may not share the same semantic content, we adapt instructions containing multiple target region descriptions by separating the instruction into parts including only a single target region description. Fig. 16 shows some example results. We leave more complex instructions to future work.

Recoloring a Design Collection. Managing an online business (*i.e.*, e-commerce), such as a clothing shop on Amazon, becomes more and more popular, especially after the Coronavirus pandemic. People (*e.g.*, businessmen) need to produce banners and advertisements efficiently to incorporate the fast-changing trend. As shown in Fig. 17, with a single instruction, our model enables recoloring for a design collection containing different images.

Iterative Graphic Design Photo Recoloring. Iterative design is a common strategy in general editing tasks. Our model also provides this function, where users can continually adjust the photo color until reaching satisfactory results, as shown in Fig. 18.

Fig. 18. Iterative graphic design photo recoloring.

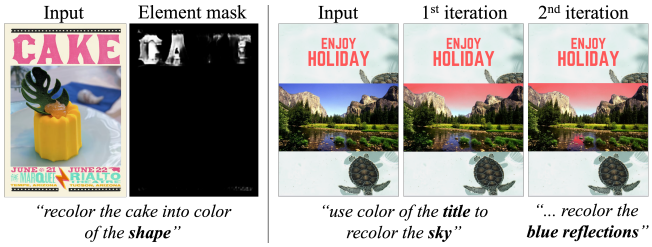


Fig. 19. Failure cases.

8 CONCLUSIONS

In this work, we have introduced a new language-based method for photo color adjustment in graphic designs. With multi-granularity instructions, users can express their design intention freely. Our method includes two essential components: the language-based source color prediction module and the semantic-palette-based photo recoloring module. A language-based design synthesis method is proposed to enable the training of our framework. We have validated the effectiveness of our model through comprehensive experiments. Our model outperforms existing methods both quantitatively and qualitatively. We have also shown a number of practical applications of our model. We will release our code and dataset to encourage more research on language-based graphic design editing.

Discussions. Our work opens up a new venue for adjusting photo color in graphic designs with language-based instructions, but it still has limitations and room for further improvement. First, following a state-of-the-art referring image segmentation work [Feng et al. 2021], the language model we use here is a basic Bi-GRU trained on our dataset from scratch, and it may not work if the target object or design element does not appear in our training set. A possible solution is to utilize a more powerful pretrained cross-modality model, *e.g.,* CLIP [Radford et al. 2021]. However, simply replacing the text encoder may not significantly improve this limitation since our language-based instructions are unique (*i.e.,* with two parts of information referring to source color and target region), and are different from the prompts used for training existing cross-modality model. Nevertheless, it would be interesting to try them as a future work. Second, even though we have tried to synthesize graphic designs with appearances as diverse as possible, we still cannot cover all the cases that appear in real life, especially creative design. For example, in Fig. 19 left, since the title "CAKE" occupies a large portion of the design with a creative font face, our model misinterprets it as a combination of shapes. We may consider alleviating this problem in the future by finetuning our language-based source color prediction module with a dataset containing real graphic designs. Third, our model may have difficulties in preserving long-range consistency, such as reflection of the sky in Fig. 19 right. To address it, further instructions can be added to iteratively refine the result, such as using "... recolor the blue reflections ..." in the example. More discussions on the scope and potential ethics issues of our work can be found in the supplementary material.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for the insightful comments and constructive suggestions on our paper. This work is in part supported by a GRF grant from the Research Grants Council of Hong Kong (Ref. No.: 11205620).

REFERENCES

Mahmoud Afifi, Abdullah Abuolaim, Mostafa Hussien, Marcus A Brubaker, and Michael S Brown. 2021a. CAMS: Color-Aware Multi-Style Transfer. *arXiv preprint arXiv:2106.13920* (2021).

Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. 2021b. Histogan: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7941–7950.

Elad Aharoni-Mack, Yakov Shambik, and Dani Lischinski. 2017. Pigment-based recoloring of watercolor paintings. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. 1–11.

Yağiz Aksoy, Tunç Ozan Aydin, Aljoša Smolić, and Marc Pollefeys. 2017. Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 1–19.

Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. 2018. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Maggie Aland. 2017. *10 tips and ideas to make a flyer that stands out.* Retrieved Jan 03, 2022 from https://www.lucidpress.com/blog/10-creative-ways-to-make-flyer-stand-out

ANL [n. d.]. *Guide to Effective Poster Design.* Retrieved Jan 03, 2022 from https://www.anl.gov/education/guide-to-effective-poster-design

Benoit Arbelot, Romain Vergne, Thomas Hurtut, and Joëlle Thollot. 2017. Local texture-based color transfer and colorization. *Computers & Graphics* 62 (2017), 15–27.

Soonmin Bae, Sylvain Paris, and Frédo Durand. 2006. Two-scale tone management for photographic look. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 637–645.

Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Keetae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo. 2018. Coloring with words: Guiding image colorization through text-based palette generation. In *Proceedings of the european conference on computer vision (eccv)*. 431–447.

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. 2020. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727* (2020).

Brent Berlin and Paul Kay. 1991. *Basic color terms: Their universality and evolution.* Univ of California Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Canva [n. d.]. *The best Google Font combinations to try.* Retrieved May 06, 2022 from https://www.canva.com/learn/best-google-font-combinations/

Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based photo recoloring. *ACM Trans. Graph.* 34, 4 (2015), 139–1.

Youngha Chang, Suguru Saito, Keiji Uchikawa, and Masayuki Nakajima. 2005. Example-based color stylization of images. *ACM Trans. Appl. Percept.* 2, 3 (2005), 322–345.

Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. 2018. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8721–8729.

Ming-Ming Cheng, Shuai Zheng, Wen-Yan Lin, Vibhav Vineet, Paul Sturgess, Nigel Crook, Niloy J Mitra, and Philip Torr. 2014. ImageSpirit: Verbal guided image parsing. *ACM Transactions on Graphics (TOG)* 34, 1 (2014), 1–11.

Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. 2020. Sequential attention GAN for interactive image editing. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4383–4391.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. 2006. Color harmonization. In *ACM SIGGRAPH 2006 Papers*. 624–630.

Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. 2017. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 5706–5714.

Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. 2016. Deepprop: Extracting deep features from a single image for edit propagation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 189–201.

Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. 2021. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15506–15515.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

GraphicsZoo 2020. *15 Tips to Create Unique Poster Design for Your Brand*. Retrieved Feb 16, 2022 from https://www.graphicszoo.com/article/15-tips-to-create-unique-poster-design-for-your-brand

Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2013. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. 2018. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–16.

Jing Huang, Shizhe Zhou, Xianyi Zhu, Yiwen Li, and Chengfeng Zhou. 2018. Automatic image style transfer using emotion-palette. In *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, Vol. 10806. International Society for Optics and Photonics, 108064A.

Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 351–354.

Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. 2021. Talk-to-Edit: Fine-Grained Facial Editing via Dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13799–13808.

Begoña Jordá-Albiñana, Olga Ampuero-Canellas, Natalia Vila, and José Ignacio Rojas-Sola. 2009. Brand identity documentation: a cross-national examination of identity standards manuals. *International Marketing Review* (2009).

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276* (2022).

Siavash Khodadadeh, Saeid Motiian, Zhe Lin, Ladislau Boloni, and Shabnam Ghadar. 2021. Automatic Object Recoloring Using Adversarial Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1488–1496.

Eungyeup Kim, Sanghyeon Lee, Jeonghoon Park, Somi Choi, Choonghyun Seo, and Jaegul Choo. 2021. Deep Edge-Aware Interactive Colorization against Color-Bleeding Effects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14667–14676.

EunJin Kim and Hyeon-Jeong Suk. 2018. Image color adjustment for harmony with a target color. *Color Research & Application* 43, 1 (2018), 75–88.

Gierad P Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. Pixeltone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2194.

Anat Levin, Dani Lischinski, and Yair Weiss. 2004. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*. 689–694.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. 2020a. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7880–7889.

Bowen Li, Xiaojuan Qi, Philip Torr, and Thomas Lukasiewicz. 2020b. Lightweight generative adversarial networks for text-guided image manipulation. *Advances in Neural Information Processing Systems* 33 (2020), 22020–22031.

Yuanzhen Li, Edward Adelson, and Aseem Agarwala. 2008. ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1255–1264.

Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. 2020. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 89–106.

Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4990–4998.

Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. 2007. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 309–320.

Timo Lüddecke and Alexander Ecker. 2022. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7086–7096.

Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. 2018. Language-driven synthesis of 3D scenes from scene databases. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–16.

Sara Mcguire. 2019. *Poster Design Guide: How to Make an Eye-Catching Poster in 2020*. Retrieved Jan 03, 2022 from https://venngage.com/blog/poster-design/

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. 2018. Text-adaptive generative adversarial networks: manipulating images with natural language. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 42–51.

Rang MH Nguyen, Brian Price, Scott Cohen, and Michael S Brown. 2017. Group-Theme Recoloring for Multi-Image Color Consistency. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 83–92.

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

Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 1221–1224.

Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics* 20, 8 (2014), 1200–1213.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2085–2094.

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*. PMLR, 8748–8763.

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

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

Jianchao Tan, Jose Echevarria, and Yotam Gingold. 2018. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Jianchao Tan, Jyh-Ming Lien, and Yotam Gingold. 2016. Decomposing images into layers via RGB-space geometry. *ACM Transactions on Graphics (TOG)* 36, 1 (2016), 1–14.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Yili Wang, Yifan Liu, and Kun Xu. 2019. An Improved Geometric Approach for Palette-based Image Decomposition and Recoloring. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 11–22.

Yi Wang, Menghan Xia, Lu Qi, Jing Shao, and Yu Qiao. 2022. PalGAN: Image Colorization with Palette Generative Adversarial Networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*. Springer, 271–288.

Shuchen Weng, Hao Wu, Zheng Chang, Jiajun Tang, Si Li, and Boxin Shi. 2022. L-CoDe:Language-Based Colorization Using Color-Object Decoupled Conditions. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 3 (2022), 2677–2684. https://doi.org/10.1609/aaai.v36i3.20170

Chloe West. 2020. *The Ultimate Guide to Flyer Design*. Retrieved May 02, 2022 from https://visme.co/blog/flyer-design/

Geoff Woolfe et al. 2007. Natural language color editing. In *ISCC Annual Meeting*. Citeseer.

Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. 2020. Phrasecut: Language-based image segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10216–10225.

Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2256–2265.

Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. 2021. SketchHairSalon: Deep Sketch-based Hair Image Synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2021)* 40, 6 (2021), 1–16.

Kun Xu, Yong Li, Tao Ju, Shi-Min Hu, and Tian-Qiang Liu. 2009. Efficient affinity-based edit propagation using kd tree. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 1–6.

Liron Yatziv and Guillermo Sapiro. 2006. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing* 15, 5 (2006), 1120–1129.

Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. 2017a. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing* 26, 4 (2017), 1952–1964.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017b. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999* (2017).

Nanxuan Zhao, Quanlong Zheng, Jing Liao, Ying Cao, Hanspeter Pfister, and Rynson WH Lau. 2021. Selective Region-based Photo Color Adjustment for Graphic Designs. *ACM Transactions on Graphics (TOG)* 40, 2 (2021), 1–16.

Shizhan Zhu, Raquel Urtasun, Sanja Fidler, Dahua Lin, and Chen Change Loy. 2017. Be your own prada: Fashion synthesis with structural coherence. In *Proceedings of the IEEE international conference on computer vision*. 1680–1688.

Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. 2019. Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.