LGA-Net: Learning Local and Global Affinities for Sparse Scribble based Image Colorization

Hongjin Lyu[1], Bo Li[2]*, Paul L. Rosin[1], Yu-Kun Lai[1]
[1] School of Computer Science and Informatics, Cardiff University
[2] School of Mathematics and Information Science, Nanchang Hangkong University
{lyuh2, RosinPL, LaiY4}@cardiff.ac.uk, libo@nchu.edu.cn

Abstract

Image colorization is a typical ill-posed problem. Among various colorization methods, scribble-based methods have a unique advantage that allows users to accurately resolve ambiguities and modify the colors of any objects to suit their specific tastes. However, due to the time-consuming scribble drawing process, users tend to draw sparse scribbles instead of dense and detailed scribbles, which makes it challenging for existing methods, especially for regions with no immediate scribbles. Facing the above problems, this paper proposes a novel colorization algorithm named Local and Global Affinity Net (LGA-Net) that formulates the scribble-based colorization task as an affinity propagation process at both local and global levels. Instead of predicting color values directly, our neural network learns to predict local and global affinity relationships between pixels for a given grayscale input, describing how colors should be propagated, which are independent of the scribbles. Given reliable affinity relationships, the color propagation process is formulated as a maximum a posteriori problem. Both local and global affinities are represented using a weighted graph and enabled by a graph Laplacian regularizer to ensure accurate color propagation. Extensive experiments demonstrate that LGA-Net produces state-of-the-art colorization results when using sparse scribbles.

1. Introduction

Image colorization predicts color channels based on grayscale input, which as an ill-posed problem. Since image colorization tasks often do not have specific 'correct' answers, a controllable colorization algorithm that can accurately depict users' aesthetic preference and/or prior knowledge of the target scene is more in line with practical requirements.

Existing image colorization works can be broadly divided into two categories: automatic and user-guided colorization, which are described in detail in Section 2. Although automatic colorization algorithms [14, 16, 31, 36, 46] can directly generate colorized results with remarkable visual quality, users cannot make changes to the colorization results based on their knowledge or individual preference. According to the different forms of user guidance, user-guided methods can be further divided into the following three sub-categories: scribbles/pixel hints based colorization [10, 20, 30, 43, 45] (referred to as scribble-based methods for simplicity), example-based colorization [8, 19, 21, 22, 38], and text-driven colorization [2, 15, 35, 47]. Although example-based and text-driven methods can also provide some control, they are not sufficiently detailed for fine-grained control, and can often introduce some ambiguities w.r.t. user preference. In contrast, scribble-based method provides intuitive, fine-grained colorization control.

Although scribble-based methods can satisfy users' aesthetic preference and prior knowledge to the greatest extent, they require user-provided detailed scribbles/pixel hints. In real-world scenarios however, users tend to draw sparse scribbles to reduce effort. They tend to only draw one or a few scribbles for multiple regions with similar texture/intended color, corresponding to e.g. multiple instances of the same type of objects, or one object split into multiple image regions due to occlusion. Moreover, existing methods tend to predict a color image as output directly, given a grayscale image and scribbles/pixel hints as input. This approach however means the network needs to not only understand the grayscale image content and structure but also how the pixel hints should be propagated in an entangled manner, leading to limited generality for realistic sparse scribbles scenarios, even with large-scale training data. Furthermore, as real hand-drawn scribbles are difficult to acquire, randomly choosing pixels as hints during training further restricts existing methods in sparse scribble scenarios, especially for distant regions with similar texture that require accurate understanding of global relationships.

*Corresponding author

Facing the above problems, we propose a fundamentally different approach for scribble-based colorization: instead of predicting color images, LGA-Net learns to predict Local and Global Affinities which are solely dependent on the grayscale image, and irrelevant to the scribbles. This makes learning much more effective and generalizable. Once the affinities are predicted, we formulate scribble-based colorization as a maximum a posteriori problem that propagates scribble colors to the rest of the image, regulated by the affinity relationships. In contrast, existing scribble-based methods learn to directly predict color, which may cause conflicts between known color from training data and user-scribbled color. Our key contributions are:

- This paper innovatively conceptualizes the image colorization task as a color propagation process based on affinity relationships. Once the affinity relationships are effectively learned, the scribbles' colors can be appropriately propagated to suitable regions.
- To efficiently produce high-quality colorized images that are in line with users' intentions, the color propagation process is formulated as a maximum a posteriori problem with a Laplacian prior on a weighted graph that represents pixel affinities, which explicitly enables both local and global affinity relationships by introducing adjacent and global points in the graph Laplacian matrix.
- Extensive experiments demonstrate that LGA-Net trained on a very small dataset containing only 4K images outperforms state-of-the-art methods even when they are trained on more than 1M images. Our method also shows good scalability and cross dataset generalization[1].

2. Related Work

2.1. Automatic Colorization

Early effort for automatic colorization [7] applied graph theory to maximize global image color probability, but the method relies on handcrafted features, and thus has limited robustness. Deep learning based methods [9, 13, 18, 44] showed improved robustness, by exploiting large amounts of paired grayscale and color images. Subsequently, Wu et al. [36] integrated the generative priors of pre-trained GANs (Generative Adversarial Networks) into the colorization process instead of using natural examples as most example-based methods do. Palette [29] applies a unified conditional diffusion model to deal with different image-to-image tasks. UniColor [12] can handle both unconditional and conditional tasks, with the help of hint points serving as an intermediate unified representation. Automatic methods can produce impressive results, but they lack the flexibility for users to customize colors according to their preferences.

[1]The source code and trained models are available at https://github.com/HONGJINLYUCS/LGA-Net-ICCV-2025.

2.2. Example-based Colorization

Example-based methods use user-provided reference images for guidance to offer some user control. He et al. [11] provided recommended reference images from the ImageNet dataset [28] based on semantic and luminance statistics. Wang et al. [33] deployed a dual pyramid architecture into exemplar-based colorization. Li et al. [23] bridged the gap between input and target images by a more reasonable gradient updating method. Although the above-mentioned methods allow user involvement during the colorization process, users still cannot provided detailed customization for the colorization results.

2.3. Text-driven Colorization

The work [25] firstly converted input text into a vector representation using a bi-directional LSTM (Long Short-Term Memory) network, which was then integrated into all intermediate feature maps of a basic fully-convolutional network. Kim et al. [15] fused textual and image structure features at the deepest layer of the generator. Weng et al. [35] innovatively used the object-adjective correspondence by utilizing a bi-affine mechanism and an attention transfer module. While text-based methods offer some level of customization, the control can still be quite limited.

2.4. Scribbles/points-based Colorization

Some scribble-based colorization methods are designed for colorizing line arts [5, 6, 43], and not suitable for general grayscale images. For grayscale images, pioneering work by Levin et al. [20] achieved scribble-based colorization by propagating scribble colors to neighboring pixels based on intensity similarities. Luan et al. [24] presented an interactive system that first groups regions with similar colors and then fine-tunes colors in such regions. These methods rely on handcrafted formulas and have limited robustness. Zhang et al. [45] proposed to use ground-truth colors of randomly sampled pixels as simulated user input to train a neural network for interactive colorization. Such hint pixels can differ significantly from user scribbles, and the method relies on large amounts of training data. Yun et al. [41] enhanced the global affinity learning based on the global receptive field and self-attention mechanism. However, existing methods poorly handle sparse scribble inputs. To address this, our LGA-Net first predicts local and global affinities from grayscale inputs, then reliably transfers user colors to the entire image via these affinities, regardless of the distance. The idea is conceptually related to edit propagation research [1, 3, 37] for image and/or video editing, although these methods use handcrafted similarity measures rather than learned affinities and address a rather different problem. Observing how color is propagated in automatic video colorization also shows usefulness for label-free visual tracking [32].

3. Methodology

3.1. Colorization by Learning Affinities

Colorization in this paper is the process of inferring the color channels based on the grayscale image (gsi) and user-specified scribbles. Inspired by concepts from graph theory, an image with N pixels can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges in $\mathcal{G}$, respectively. Each pixel i, $(i = 1, 2, \ldots, N)$ in the image is represented by a vertex in $\mathcal{V}$, and the connection between two pixels is represented by an edge in $\mathcal{E}$. After assigning weights to $\mathcal{E}$, we can get the relationship weight matrix $\boldsymbol{\mathcal{W}}$, where $w_{ij} \in \boldsymbol{\mathcal{W}}, (i, j) \in \mathcal{E}$ for $i, j \in \mathcal{V}$.

Based on graph theory, the scribbles' colors can be reliably propagated throughout an entire image if the precise $\mathcal{G}$ corresponding to gsi can be obtained. The key point is to obtain $\boldsymbol{\mathcal{W}}$ in a way that is both accurate and reliable. Based on the powerful feature extraction and representation capability of neural networks, we compute $w_{ij} \in \boldsymbol{\mathcal{W}}$ using a neural network CNN_F, which provides a suitable representation of the original input image for color propagation.

Instead of directly predicting pairwise affinity relationships between pixels, which can be highly expensive, CNN_F is formulated as a matrix-feature function $\mathcal{F}_{gsi}$: $\mathbb{R}^N \mapsto \mathbb{R}^{N \times H}$, which maps each grayscale pixel to an H-dimensional feature space. Then the edge weight w_{ij} is calculated according to the following formula:

$$w_{ij} = \exp\left(-\frac{\sum_{h=1}^{H}\left(f_i^h - f_j^h\right)^2}{2\sigma^2}\right) \tag{1}$$

where f_i^h represents the i^{th} element in the h^{th} feature map, and σ is a pre-defined parameter.

3.2. Laplacian Coloring Layer

Building upon the user-provided scribbles and the obtained weighted graph $\mathcal{G}$, we treat the color propagation process as a maximum a posteriori problem with a Laplacian prior, realized through the Laplacian Coloring layer (LCL) which does not require training:

$$x^\star = \arg\min_x \|y - x\|_{\boldsymbol{M}}^2 + \alpha \cdot x^\text{T} \boldsymbol{L} x \tag{2}$$

where y of size $N \times 1$ denotes the user-provided scribbles' colors (taking one chrominance channel for example), and $x^\star$ of size $N \times 1$ is the color channel after propagation. The first term in Eq.(2) is the fidelity term which minimizes the discrepancy between y and $x^\star$. $\boldsymbol{M}$ is an $N \times N$ diagonal matrix, in which the diagonal elements with value 1 indicate the positions of scribbled points (and 0 otherwise). The second term is the Laplacian regularizer of the graph corresponding to gsi; α serves as a weighting term to balance the above two terms. Given a specific $\boldsymbol{\mathcal{W}}$, the corresponding degree matrix $\boldsymbol{D}$ is a diagonal matrix, in which the diagonal elements $D_{ii} = \sum_{j=1}^{N} w_{ij}$. Then, the Laplacian matrix of $\mathcal{G}$ is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{\mathcal{W}}$.

3.2.1. Enhanced Local-Global Affinity Regularization

In this paper, we build a graph Laplacian matrix containing both local and global affinities for a more accurate regularizer $x^\text{T} \boldsymbol{L} x$. In detail, each pixel is firstly related to the surrounding eight neighboring pixels, which enables local level affinity. Secondly, we further choose a subset of pixels as global pixels, which are evenly spaced in a regular grid, with the gap between adjacent global pixels referred to as global steps (GS). Each global pixel is connected to every other global pixel when forming $\mathcal{G}$; these are used to describe global (long-range) affinity relationships. Intuitively, these global pixels are used as anchor points at different positions, which enables the priors for global affinities to be explicitly added into regularizer $x^\text{T} \boldsymbol{L} x$ to enhance the representation for image structure.

3.2.2. Singular Matrix and Solution Ambiguity

Eq.(2) gives a formulation for scribble color propagation, and can be rewritten in the matrix form:

$$(\boldsymbol{M} + \alpha \boldsymbol{L}) x^\star = \boldsymbol{M} y \tag{3}$$

In practice, depending on the scribbles drawn and the affinity weights, the matrix $\boldsymbol{M} + \alpha \boldsymbol{L}$ may be degenerate, resulting in the linear system not having a unique solution. To address such ambiguities, a weak regularization constraint on adjacent pixels is introduced with a small balancing weight ϕ, which enforces adjacent pixels to have similar colors. This extends Eq. (2) to the following:

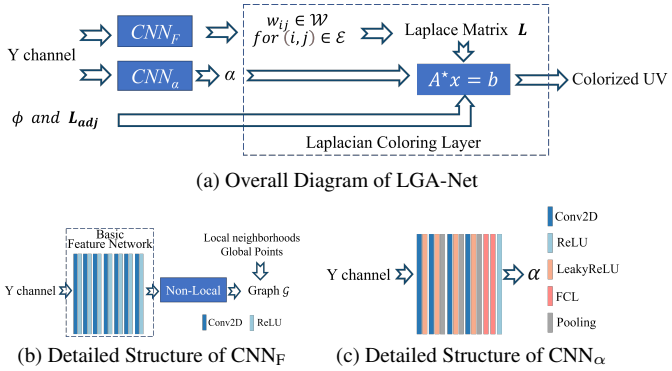
$$x^\star = \arg\min_x \|y - x\|_{\boldsymbol{M}}^2 + \alpha \cdot x^\text{T} \boldsymbol{L} x + \phi \cdot x^\text{T} \boldsymbol{L}_{adj} x \tag{4}$$

$$x^\star = (\boldsymbol{M} + \alpha \boldsymbol{L} + \phi \boldsymbol{L}_{adj})^{-1} \boldsymbol{M} y \tag{5}$$

where $\boldsymbol{L}_{adj}$ is an $N \times N$ matrix that serves as the newly added adjacency constraint term, which is 1 for adjacent pixels and 0 otherwise. ϕ is set to an appropriate value, which is sufficient to stabilize the linear system while minimizing its impact on the final solution as much as possible. Since the new coefficient matrix $\boldsymbol{A}^\star = \boldsymbol{M} + \alpha \boldsymbol{L} + \phi \boldsymbol{L}_{adj}$ is non-singular, there is a closed-form solution, as shown in Eq. (4) and Eq. (5).

3.2.3. Sparse Tensor Optimization

The coefficient matrix $\boldsymbol{A}^\star$ is of size $N \times N$, which would have significant memory/computation demands during training if treated as a dense tensor. Thus, all operations of LCL are conducted using sparse tensors. Especially, the calculation of $(\boldsymbol{M} + \alpha \boldsymbol{L} + \phi \boldsymbol{L}_{adj})^{-1} \boldsymbol{M} y$ using dense tensors is prohibitively expensive due to the immense GPU resources required. To overcome this, LGA-Net utilizes a CPU-based method [17] to solve this sparse linear system, which significantly reduces the memory and time required.



(a) Overall Diagram of LGA-Net

(b) Detailed Structure of CNN_F

(c) Detailed Structure of CNN_α

Figure 1. Overview and detailed structure of LGA-Net

3.3. Network architecture

As shown in Fig. 1a, the luminance channel Y is firstly fed into CNN_F to obtain the rich high-dimensional feature representation, which is beneficial for predicting more accurate affinity weights w_{ij}. α, as a key weight to balance the fidelity term and the regularization term, is also learnable, and predicted by a separate light-weight neural network CNN_α as shown in Fig. 1c; $\boldsymbol{L}_{adj}$ is a fixed matrix that enforces a weak connection between each pixel and its neighbors; ϕ is a fixed weight for $\boldsymbol{L}_{adj}$. LGA-Net first builds a Laplacian matrix $\boldsymbol{L}$ reflecting the structural information of the input image based on the output of CNN_F, and then solves a large sparse linear equation system $\boldsymbol{A}^\star x = b$ according to the given Laplacian prior, α, ϕ and $\boldsymbol{L}_{adj}$.

CNN_F as shown in Fig. 1b, consists of the basic feature extraction sub-network BFEnet and the Non-local block (NLB) [34]. BFEnet is composed of only seven convolutional layers, which reduces computational complexity, enhances efficient gradient propagation and training convergence. Further, richer high-dimensional features are introduced by embedding NLB after BFEnet, which improves the capability of LGA-Net to perceive image structure.

3.3.1. Loss function

Let $\mathcal{T}$ denote the ground truth domain, $\mathcal{C}$ denote the colorized domain, and $\mathcal{D}_t$ denote the training dataset. For a training example $d \in \mathcal{D}_t$, $\mathcal{T}^d$ and $\mathcal{C}^d$ refer to the ground truth color image and the colorized result by LGA-Net.

Firstly the simple but efficient Least Absolute Deviations $\mathcal{L}_1$ is applied to enable a pixel-wise difference judgment:

$$\mathcal{L}_1\left(\mathcal{T}^d, \mathcal{C}^d\right) = \sum_{i=1}^{N} \|\mathcal{T}_i^d - \mathcal{C}_i^d\|. \tag{6}$$

where i is used as the pixel index.

Secondly, we apply the Laplacian pyramid loss $\mathcal{L}_{lap}$ [4] to perform multi-resolution analysis between $\mathcal{T}$ and $\mathcal{C}$:

$$\mathcal{L}_{lap}\left(\mathcal{T}^d, \mathcal{C}^d\right) = \sum_{p} 2^{2p} \left\|L^p(\mathcal{T}^d) - L^p(\mathcal{C}^d)\right\| \tag{7}$$

where $L^p(\cdot)$ means the p-th level of the Laplacian pyramid feature of the input.

Finally, for smoother results, we introduce total variation loss $\mathcal{L}_{TV}$ to explicitly constrain $\mathcal{C}$ in terms of spatial variation:

$$\mathcal{L}_{TV}(\mathcal{C}^d) = \sum_{i=1,2,\ldots,N; j \in Nbr(i)} \|\mathcal{C}_i^d - \mathcal{C}_j^d\|_2^2 \tag{8}$$

where $Nbr(i)$ defines the neighbor pixels of i. The total loss is a weighted sum of these terms:

$$\mathcal{L} = \sum_{d \in \mathcal{D}_t} \left(\mathcal{L}_1(\mathcal{T}^d, \mathcal{C}^d) + \lambda_{lap}\mathcal{L}_1(\mathcal{T}^d, \mathcal{C}^d) + \lambda_{TV}\mathcal{L}_{TV}(\mathcal{C}^d)\right), \tag{9}$$

where λ_{lap} and λ_{TV} are balancing weights.

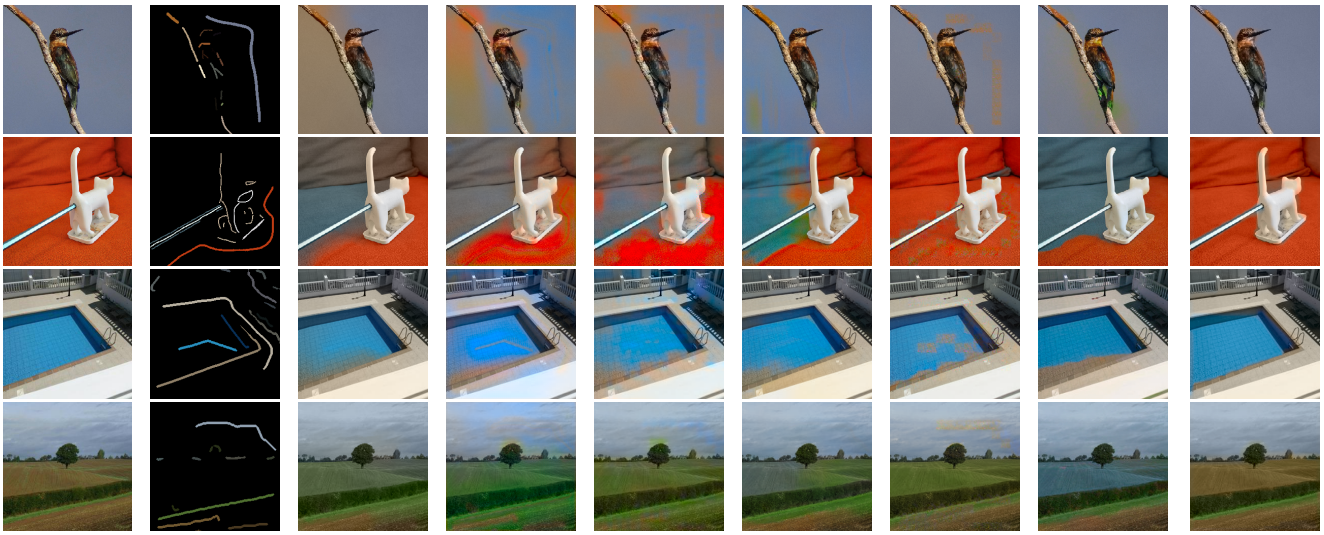
4. Evaluation

In this section, the implementation details are described in detail in Section 4.1. Then, we compare LGA-Net with the state-of-the-art scribble-based colorization methods in Section 4.2. The visualization of affinities relationships is shown in Section 4.3, enabling a more intuitive understanding of LGA-Net. The ablation study for each key component is presented in Section 4.4.

4.1. Implementation Details

LGA-Net is implemented based on PyTorch, where the training and testing phases of all experiments presented in this paper are conducted on an NVIDIA Tesla P100 GPU. ϕ, serving as the pre-defined weight of L_{adj}, is configured as 10^{-8}. λ_{lap} and λ_{TV} are set to 1.5 and 25, respectively. The Adam optimizer is used, where the learning rate is set to 10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.999$. LGA-Net training process is performed at 128×128 scale, with the Y-channel resized to this size before input and the colorized results bilinearly up-sampled for the final output. This is sufficient in practice, as lower chrominance resolution is often unnoticeable. In addition, LGA-Net can be trained to handle larger images. For 128×128, 256×256, and 512×512, peak GPU memory usage is 3.05GB, 8.36GB, and 22.70GB respectively, so the space complexity is sublinear w.r.t. the number of pixels. The training time per epoch is 2.3h, 12.5h, and 75h respectively, which is slightly higher than linear w.r.t. the number of pixels. The reason behind this is that the CPU-based method [17] mentioned in Section 3.2.3 is applied to solve the large sparse system. Although the current CPU-based sparse solver is quite slow, it ensures LGA-Net is memory efficient and has good scalability compared with GPU-based dense solvers.

Leveraging ImageNet's semantic diversity, we create the training dataset $\mathcal{D}_t$ containing 4K images by selecting 4 images from each category. For testing purposes, $\mathcal{D}_{manual}$ contains 200 color images randomly chosen from the ImageNet test dataset, with the authors creating sparse scribbles



(a) GT (b) Hint (c) Levin (d) Z4K (e) iC4K (f) Z1M (g) iC1M (h) UC1M (i) LGA-Net

Figure 2. Qualitative comparison with state-of-the-art methods in Section 4.2.1. This figure shows four different examples (top two from $\mathcal{D}_{manual}$ and bottom two from $\mathcal{D}_{auto}$. The first two columns show the ground truth and the applied scribbles, while the following columns separately display: Levin [20]; Z4K and iC4K (Zhang [45] and iColoriT [41] trained on $\mathcal{D}_t$); Z1M, iC1M and UC1M (official pre-trained models on ImageNet for Zhang et al. [45], iColoriT [41] and UniColor [12]); our LGA-Net.

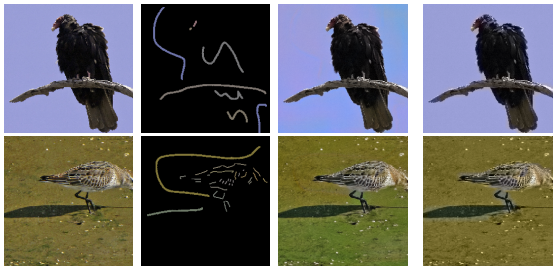
for each to simulate user-provided inputs. While $\mathcal{D}_{manual}$ only comprises 200 examples due to the labor-intensive process of manually creating scribbles, leveraging AutoSS (a new automatic scribble generation algorithm described in the supplementary material), we further create $\mathcal{D}_{auto}$ by randomly selecting 3K images from the Place365 test dataset and generating sparse scribbles accordingly.

4.2. Compared with Representative Methods

Diffusion-based colorization methods are becoming popular but existing diffusion-based methods often use text prompts rather than scribbles to guide colorization. Moreover,: [39] lacks a mechanism to map scribbles or points to latent space, making direct comparison in scribble-based tasks unfeasible. The methods [40, 42] have not released any official code. Thus, LGA-Net is compared with the following four scribble-based methods: Levin et al. [20], Zhang et al. [45], UniColor[12] and iColoriT[41]. Zhang [45] and iColoriT [41] trained under $\mathcal{D}_t$ are abbreviated as Z4K and iC4K. The official pre-trained models on the ImageNet of Zhang[45], UniColor[12] and iColoriT[41] are abbreviated as Z1M, UC1M and iC1M separately. Due to the lack of official training instruction, we do not compare UniColor under $\mathcal{D}_t$.

Regarding [26], no official code release makes direct comparison infeasible. Despite a seemingly similar colorization equation, LGA-Net differs fundamentally. The method can auto-generate diverse colorizations and accept user constraints. As a tougher task, it needs two-stage training with color images as input and relies on strong priors (e.g., "pixels with similar intensities should have similar colorizations") to regularize the problem. In contrast, LGA-Net applies pre-learned affinities to achieve scribble-based colorization. Taking only grayscale images, our method shows strong generalizability: our 4K-trained model outperforms other methods trained on 1.3M images and works well on datasets beyond the training set (Section 4.2). [26] is trained on larger datasets, excelling on specific-content ones (e.g., faces). Its binary-term approach is costly, and shown images have low resolution. Conversely, LGA-Net with sparse matrices has better scalability (Section 4.1).

In order to obtain more reliable comparison, the following content includes qualitative analysis (Figs. 2, 3, and 4), quantitative analysis (Tables 1 – 2), and user study (Fig. 5). More examples including different scenarios are shown in the supplementary material.



(a) GT (b) Hint (c) Z1M (d) LGA-Net

Figure 3. Poor Z1M result under sparse scribbles.

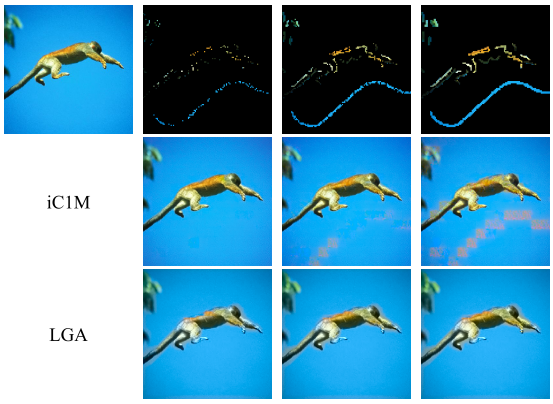


Figure 4. iC1M comparison under different point sparsity levels.

Table 1. Quantitative analysis under $\mathcal{D}_{manual}$ in Section 4.2.2

$\mathcal{D}_{manual}$ 200 cases	MSE↓ $\times 10^{-2}$	PSNR↑ dB	MS-SSIM↑ $\times 10^{-2}$	LPIPS↓ $\times 10^{-2}$
Levin	4.41	21.48	91.69	16.50
Z4K	5.81	19.72	83.43	25.16
iC4K	4.49	21.32	85.22	21.41
Z1M	3.65	23.19	90.77	12.14
iC1M	1.53	25.92	90.34	12.43
UC1M	2.13	25.36	93.62	7.90
LGA-Net	1.42	26.07	92.78	10.49

Table 2. Quantitative analysis under $\mathcal{D}_{auto}$ in Section 4.2.2

$\mathcal{D}_{auto}$ 3000 cases	MSE↓ $\times 10^{-2}$	PSNR↑ dB	MS-SSIM↑ $\times 10^{-2}$	LPIPS↓ $\times 10^{-2}$
Levin	1.11	29.52	95.73	10.65
Z4K	2.42	25.12	91.53	16.26
iC4K	2.96	25.08	91.83	15.70
Z1M	1.16	28.47	95.30	9.74
iC1M	1.08	28.75	94.84	10.42
UC1M	0.95	29.15	95.79	8.11
LGA-Net	0.90	29.89	95.73	9.62

4.2.1. Qualitative Analysis

Levin's results shown in the 3rd column of Fig. 2, are acceptable for local regions based on provided color scribbles. But Levin struggles to propagate color to distant regions with similar texture since it lacks a mechanism for learning global affinities, hindering remote color propagation.

In Fig. 2, the 4th and 5th columns show Z4K and iC4K results. Despite strictly following official training guidances on $\mathcal{D}_t$, both models fail to generate stable outputs across local and remote regions due to the dataset's limited size. This limited size of $\mathcal{D}_t$ hinders Zhang and iColoriT, which require extensive training data, and even iC4K utilizing Transformers fails to compensate adequately.

In Fig. 2, the 6th, 7th, and 8th columns show Z1M, iC1M, and UC1M results. Unlike Z4K and iC4K trained under limited cases, these models leverage the power of

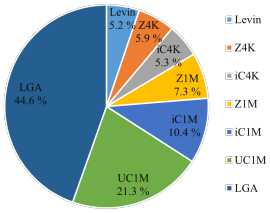


Figure 5. The user study's statistical results

Figure 6. Visualization of local and global affinities in Section 4.3. For local affinity, the heatmaps show the affinity for each pixel with its neighbor in the specific direction. For global affinity, each local map indicates the affinity of the highlighted position (in red) and other global points.

ImageNet to handle color propagation. However, Z1M, as an image-to-image colorization approach, lacks robustness for long-range propagation due to implicit affinity storage in network parameters. In addition, Fig. 3 highlights challenges with sparse scribbles only in the sky (upper left and lower right) and grass (upper half), where LGA-Net properly extends colors to regions without scribbles, while Z1M relies on prior knowledge, leading to inaccuracies in distant regions without scribbles.

The self-attention mechanism allows iC1M to learn the global affinities. However, using a light-weight pixel shuffling operation to speed up processing, especially with high upsampling ratios, leads to significant artifacts when dense scribble hints are provided. Fig. 4 demonstrates iC1M's generation results under varying levels of hint sparsity, revealing increased artifact occurrence and prominence with denser hints. This highlights the limitations of iColoriT [41] in handling scribbles and dense pixel hint inputs.

UniColor leverages Transformer and VQGAN to improve final performance. However, unlike LGA-Net, which directly enables local and global affinities, UC1M struggles with remote color propagation due to lack of explicit global affinity learning, even with the large ImageNet dataset.

Fig. 2's 9th column displays LGA-Net's results, offering proper colors for both short and long-range propagation, even with the much smaller 4K training set. This is due to explicitly enabling local and global affinities and faithful color propagation process.

4.2.2. Quantitative Analysis

Four evaluation metrics (mean squared error (MSE), peak signal-to-noise ratio (PSNR), Multi-scale Structural Simi-

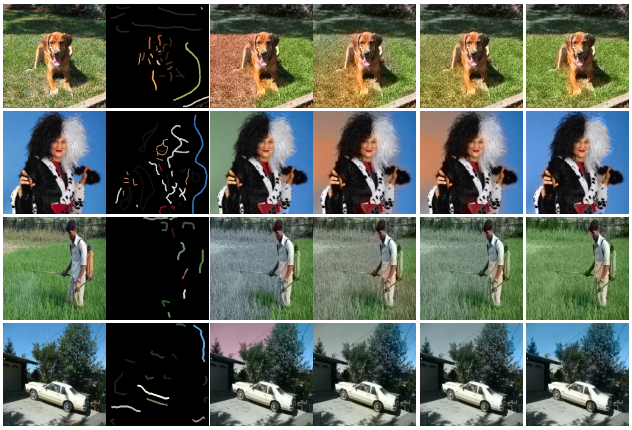
larity index measure (MS-SSIM), and the learned perceptual image patch similarity (LPIPS)) are applied to reflect the colorization quality. MS-SSIM is applied instead of SSIM since [27] shows that MS-SSIM is the most consistent metric with human judgment. Quantitative analyses on $\mathcal{D}_{manual}$ and $\mathcal{D}_{auto}$ are separately presented in Tables 1 and 2. Key conclusions drawn from the experiments are:

- Lacking global affinity capture, Levin [20] performs worse than LGA-Net on both test datasets, especially the 57.29% increase in LPIPS on $\mathcal{D}_{manual}$.
- Z4K and iC4K exhibit the worst performance, highlighting Zhang [45] and iColoriT's [41] reliance on large datasets. iC4K slightly outperforms Z4K due to the Transformer's enhanced affinity learning.
- Z1M and iC1M consistently underperform compared to LGA-Net across all metrics. Z1M's implicit affinity representation hampers remote color propagation, with an average LPIPS increase of 8.49% and PSNR drop of 2.15 dB. Although iC1M benefits from the Transformer, it still struggles with artifacts from scribbles and dense hints.
- UC1M trained on 1.3M cases roughly matches LGA-Net trained on 4K cases, highlighting that LGA-Net reduces the need for extensive data and simplifies training by redefining the coloring task and explicitly incorporating local and global affinities.
- LGA-Net outperforms others in MSE and PSNR, and ranks second in LPIPS and MS-SSIM with minimal differences. The explicit affinity learning enables stable colorization, as confirmed by qualitative analysis.

4.2.3. User Study

A user study shown in Fig. 5 is conducted for more convincing evaluation. The total area covered by the colors represents 100%. Larger areas indicate stronger competitiveness of the corresponding methods. We randomly selected 15 test examples from both $\mathcal{D}_{manual}$ and $\mathcal{D}_{auto}$, forming a total of 30 questions. Each question presents participants with results from seven methods, displayed in random order. Participants were asked to select the option they believed achieved the best coloring quality. 1500 votes from 50 participants were finally collected.

LGA-Net obtains the most votes (44.6%), which further demonstrates the superiority of LGA-Net in the sparse-scribbles-based colorization scenario. Levin which lacks sufficient ability to face remote color propagation, receives the least 5.2% votes. Z4K and iC4K cannot securely generate acceptable colorization results, which only receives 5.9% and 5.3% of the votes, respectively. iC1M and Z1M receives 10.4% and 7.3% of the votes, respectively, mainly attributing to the powerful training dataset. This big disparity with LGA-Net arises primarily from the lack of an explicit affinity learning mechanism. With the help of Transformer and VQGAN, UC1M achieves the performance closest to LGA-Net among all the comparison methods, albeit with a significant 23.3% gap still present.



(a) GT (b) Hint (c) R_{GP} (d) R_{AP} (e) R_{NLB} (f) LGA

Figure 7. Ablation study results in Section 4.4.1.

4.3. Affinity Visualization

Fig. 6 clearly illustrates the roles of local and global affinities in LGA-Net through heatmaps, providing intuitive insights into the model's operational principles. The 4th and 5th columns depict the heatmaps of local affinities corresponding to the left and bottom directions, showing that LGA-Net can accurately reflect image structures from these local directions. LGA-Net employs a total of eight distinct directions of local affinities, jointly ensuring the stability of local color propagation. The 6th column uses heatmaps to show affinities between sampled points (red boxes) and all global points (blue boxes), where high-value global affinities digitize accurate numerical representations of distant similar-textured regions, assisting LGA-Net in a more comprehensive understanding of image structures.

Table 3. Quantitative analysis under $\mathcal{D}_{manual}$ in Section 4.4.2

$\mathcal{D}_{manual}$ 200 cases	MSE↓ $\times 10^{-2}$	PSNR↑ dB	MS-SSIM↑ $\times 10^{-2}$	LPIPS↓ $\times 10^{-2}$
R_{GP}	4.81	21.89	91.10	15.38
R_{AP}	2.24	24.40	91.02	13.89
R_{NLB}	1.55	25.82	92.56	11.19
LGA-Net	1.42	26.07	92.78	10.49

Table 4. Quantitative analysis under $\mathcal{D}_{auto}$ in Section 4.4.2

$\mathcal{D}_{auto}$ 3000 cases	MSE↓ $\times 10^{-2}$	PSNR↑ dB	MS-SSIM↑ $\times 10^{-2}$	LPIPS↓ $\times 10^{-2}$
R_{GP}	2.01	28.16	95.01	11.30
R_{AP}	1.32	28.82	95.06	11.48
R_{NLB}	1.04	29.55	95.64	10.10
LGA-Net	0.90	29.89	95.73	9.62

4.4. Ablation Study

The role of each key element in LGA-Net is analyzed in this section from two perspectives: qualitative analysis as shown in Fig. 7 and quantitative analysis in Table 3 and Table 4. Further ablation study on three different loss terms

Table 5. Quantitative analysis in different NLBs in Section 4.4.4

$\mathcal{D}_{manual}$ 200 cases	MSE↓ $\times 10^{-2}$	PSNR↑ dB	MS-SSIM↑ $\times 10^{-2}$	LPIPS↓ $\times 10^{-2}$
Concatenation	1.55	25.77	92.56	10.93
EbGaussian	1.47	25.88	92.54	10.94
Gaussian	1.52	25.81	92.62	10.98
Dot product	1.42	26.07	92.78	10.49

and more examples including different scenarios are shown in the supplementary material.

4.4.1. Qualitative Analysis

Fig. 7 shows results from three ablation experiments (removing global points/affinities (R_{GP}), removing local points/affinities (R_{AP}) and removing the Non-local Block (R_{NLB})) and full LGA. The top two examples are from $\mathcal{D}_{manual}$, and the bottom two are from $\mathcal{D}_{auto}$. R_{GP} (3rd column) only keep adjacency affinities constraints, limiting its ability to achieve proper remote color propagation. R_{AP} (4th column), which retains only global affinities, exhibits some degree of short-range and long-range color propagation but falls short of stably generating high-quality colorization results. R_{NLB} (5th column) relies solely on the raw feature representation from BFEnet, compromising color propagation accuracy. With the help of NLB, Full LGA-Net (6th column) incorporating both local and global affinities achieves the best performance, demonstrating the importance of all components for optimal performance.

4.4.2. Quantitative Analysis

Similar to Section 4.2.2, the same four metrics are applied to evaluate the effects of different components quantitatively. Results on $\mathcal{D}_{manual}$ and $\mathcal{D}_{auto}$ are shown in Table 3 and Table 4, respectively.

In Tables 3 and 4, R_{GP} exhibits the worst performance across all metrics, particularly evident with a 4.18dB decline in PSNR and 46.62% deterioration in LPIPS on $\mathcal{D}_{manual}$ compared to LGA-Net. This performance gap highlights the crucial role of global points/affinities. Compared to R_{GP}, R_{AP} performs better, achieving only marginal improvements in the LPIPS evaluation under $\mathcal{D}_{manual}$ and the MS-SSIM evaluation under $\mathcal{D}_{auto}$. The reason behind is that, in the colorization task with sparse scribbles, the global points retained in R_{AP} can still learn both local and global affinities to some extent. R_{NLB} consistently under-performs compared to LGA-Net in all evaluation scenarios due to the lack of richer feature representation from NLB. LGA-Net uniformly outperforms R_{GP}, R_{AP} and R_{NLB} across all evaluation scenarios, which further demonstrates the essential roles of local/global affinities and NLB in achieving high-quality colorization results.

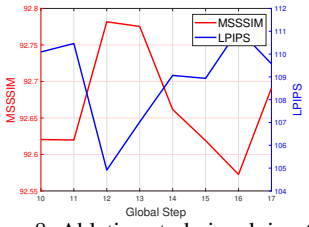


Figure 8. Ablation study involving GS.

4.4.3. Global Step

This section examines the impact of global points sparsity on colorization quality by adjusting $GS \in [10, 17]$, as shown in Fig. 8. Increasing GS enables sparser global pixels, leading to insufficient global affinity constraints, thereby increasing LPIPS and decreasing MS-SSIM. Conversely, decreasing GS enhances global affinities but also increases sensitivity to training data, limiting generalizability. At GS=12, LGA-Net achieves optimal performance.

4.4.4. Different Non-Local Blocks

Four NLB types based on different affinity functions (Concatenation; Embedded Gaussian (EbGaussian); Gaussian; Dot Product) are utilized and analyzed quantitatively on $\mathcal{D}_{manual}$ (Table 5). Overall, all NLB types enhance LGA-Net's ability to learn better affinities. The dot product version achieves the best performance, due to its suitability for weight computation relying on Euclidean geometry.

5. Conclusion

This paper proposes LGA-Net which regards the scribble-based colorization task as an affinity propagation process. For a given grayscale input, LGA-Net can accurately predict the pixel affinities that indicate the image structure information, regardless of the input hints or the color propagation process, thus achieving better generality. User-provided color information is propagated into the whole image in the form of solving a maximum a posteriori problem with Laplacian prior under the guidance of the pre-calculated local and global affinities. Global affinities boost accuracy of image structure understanding but raise computational burden. Future study will further seek better affinity formulation for efficiency.

Acknowledgments

This work was supported by the China Scholarship Council [grant number 201806420014]. The study also benefited from the ARCCA computing facilities. This work was partially funded by Natural Science Foundation of China (NSFC) under Grant 62472205, 62172198, Key Project of Jiangxi Natural Science Foundation 20224ACB202008, Key R&D Plan of Jiangxi Province (20232BBE50022) Ganpo Talent Support Program 20232BCJ22001, and the Engineering and Physical Sciences Research Council [No. EP/Y028805/1].

References

[1] Xiaobo An and Fabio Pellacini. Appprop: all-pairs appearance-space edit propagation. In *ACM SIGGRAPH 2008 papers*, pages 1–9. 2008. 2

[2] Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Keetae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo. Coloring with words: Guiding image colorization through text-based palette generation. In *European Conference on Computer Vision*, pages 431–447, 2018. 1

[3] Pravin Bhat, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics (TOG)*, 29(2):1–14, 2010. 2

[4] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv:1707.05776*, 2017. 4

[5] Yu Cao, Xiangqiao Meng, PY Mok, Xueting Liu, Tong-Yee Lee, and Ping Li. AnimeDiffusion: Anime face line drawing colorization via diffusion models. *arXiv preprint arXiv:2303.11137*, 2023. 2

[6] Hernan Carrillo, Michaël Clément, Aurélie Bugeau, and Edgar Simo-Serra. Diffusart: Enhancing line art colorization with conditional diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3485–3489, 2023. 2

[7] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European Conference on Computer Vision*, pages 126–139, 2008. 2

[8] Shu-Yu Chen, Jia-Qi Zhang, Lin Gao, Yue He, Shihong Xia, Min Shi, and Fang-Lue Zhang. Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1198–1208, 2020. 1

[9] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *IEEE International Conference on Computer Vision*, pages 567–575, 2015. 2

[10] Zhi Dou, Ning Wang, Baopu Li, Zhihui Wang, Haojie Li, and Bin Liu. Dual color space guided sketch colorization. *IEEE Transactions on Image Processing*, 30:7292–7304, 2021. 1

[11] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018. 2

[12] Zhitong Huang, Nanxuan Zhao, and Jing Liao. UniColor: A unified framework for multi-modal colorization with transformer. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 2, 5

[13] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016. 2

[14] Xin Jin, Zhonglan Li, Ke Liu, Dongqing Zou, Xiaodong Li, Xingfan Zhu, Ziyin Zhou, Qilong Sun, and Qingyu Liu. Focusing on persons: Colorizing old images learning from modern historical movies. In *ACM International Conference on Multimedia*, pages 1176–1184, 2021. 1

[15] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2Pix: Line art colorization using text tag with secat and changing loss. In *IEEE/CVF International Conference on Computer Vision*, pages 9056–9065, 2019. 1, 2

[16] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer. *arXiv preprint arXiv:2102.04432*, 2021. 1

[17] Floris Laporte. torch_sparse_solve: A sparse KLU solver for PyTorch. https://pypi.org/project/torch-sparse-solve/, 2021. Accessed: 2024-07-18. 3, 4

[18] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016. 2

[19] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5801–5810, 2020. 1

[20] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH*, pages 689–694. 2004. 1, 2, 5, 7

[21] Bo Li, Yu-Kun Lai, Matthew John, and Paul L Rosin. Automatic example-based image colorization using location-aware cross-scale matching. *IEEE Transactions on Image Processing*, 28(9):4606–4619, 2019. 1

[22] Haoxuan Li, Bin Sheng, Ping Li, Riaz Ali, and CL Philip Chen. Globally and locally semantic colorization via exemplar-based broad-GAN. *IEEE Transactions on Image Processing*, 30:8526–8539, 2021. 1

[23] Zekun Li, Zhengyang Geng, Zhao Kang, Wenyu Chen, and Yibo Yang. Eliminating gradient conflict in reference-based line-art colorization. In *European Conference on Computer Vision*, pages 579–596. Springer, 2022. 2

[24] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007. 2

[25] Varun Manjunatha, Mohit Iyyer, Jordan Boyd-Graber, and Larry Davis. Learning to color from language. *arXiv preprint arXiv:1804.06026*, 2018. 2

[26] Safa Messaoud, David Forsyth, and Alexander G Schwing. Structural consistency and controllability for diverse colorization. In *European Conference on Computer Vision (ECCV)*, pages 596–612, 2018. 5

[27] Seán Mullery and Paul F Whelan. Human vs objective evaluation of colourisation performance. *arXiv:2204.05200*, 2022. 7

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2

[29] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, pages 1–10, 2022. 2

[30] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017. 1

[31] Patricia Vitoria, Lara Raad, and Coloma Ballester. ChromaGAN: Adversarial picture colorization with semantic class distribution. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2445–2454, 2020. 1

[32] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018. 2

[33] Hanzhang Wang, Deming Zhai, Xianming Liu, Junjun Jiang, and Wen Gao. Unsupervised deep exemplar colorization via pyramid dual non-local attention. *IEEE Transactions on Image Processing*, 2023. 2

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 4

[35] Shuchen Weng, Hao Wu, Zheng Chang, Jiajun Tang, Si Li, and Boxin Shi. L-CoDe: Language-based colorization using color-object decoupled conditions. 2022. 1, 2

[36] Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. Towards vivid and diverse image colorization with generative color prior. In *IEEE/CVF International Conference on Computer Vision*, pages 14377–14386, 2021. 1, 2

[37] Li Xu, Qiong Yan, and Jiaya Jia. A sparse control model for image and video editing. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013. 2

[38] Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang. Stylization-based architecture for fast deep exemplar colorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9363–9372, 2020. 1

[39] Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. In *ECCV*, pages 74–91. Springer, 2025. 5

[40] Wanyu Yang, Feifan Cai, Yang Shu, Zihao Zhang, Qi Liu, and Youdong Ding. Colorize at will: Harnessing diffusion prior for image colorization. *IEEE Access*, 2024. 5

[41] Jooyeol Yun, Sanghyeon Lee, Minho Park, and Jaegul Choo. iColoriT: Towards propagating local hints to the right region in interactive colorization by leveraging vision transformer. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1787–1796, 2023. 2, 5, 6, 7

[42] Nir Zabari, Aharon Azulay, Alexey Gorkor, Tavi Halperin, and Ohad Fried. Diffusing colors: Image colorization with text guided diffusion. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 5

[43] Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, and Chunping Liu. User-guided line art flat filling with split filling mechanism. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9889–9898, 2021. 1, 2

[44] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*, pages 649–666. Springer, 2016. 2

[45] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 36(4):119:1–11, 2017. 1, 2, 5, 7

[46] Jiaojiao Zhao, Jungong Han, Ling Shao, and Cees GM Snoek. Pixelated semantic colorization. *International Journal of Computer Vision*, 128(4):818–834, 2020. 1

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