CT^2: Colorization Transformer via Color Tokens

Shuchen Weng[#1], Jimeng Sun[#2], Yu Li[3], Si Li[2], and Boxin Shi[*1]

[1] NERCVT, School of Computer Science, Peking University
[2] School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[3] International Digital Economy Academy
{shuchenweng,shiboxin}@pku.edu.cn
{sjm,lisi}@bupt.edu.cn
liyu@idea.edu.cn

Abstract. Automatic image colorization is an ill-posed problem with multi-modal uncertainty, and there remains two main challenges with previous methods: incorrect semantic colors and under-saturation. In this paper, we propose an end-to-end transformer-based model to overcome these challenges. Benefited from the long-range context extraction of transformer and our holistic architecture, our method could colorize images with more diverse colors. Besides, we introduce color tokens into our approach and treat the colorization task as a classification problem, which increases the saturation of results. We also propose a series of modules to make image features interact with color tokens, and restrict the range of possible color candidates, which makes our results visually pleasing and reasonable. In addition, our method does not require any additional external priors, which ensures its well generalization capability. Extensive experiments and user studies demonstrate that our method achieves superior performance than previous works.

1 Introduction

Image colorization, a classic computer vision task, aims to convert the grayscale image into a plausible colorful one, which has broad applications in legacy image/video restoration, artistic creation, and image compression. To meet the requirement of colorization, fully automatic methods seek and cue appropriate color hints from complex image semantics (*e.g.*, shape, texture, and context).

In earlier methods, researchers focus on feature engineering, which takes handcraft approaches [6] or pyramid-shaped encoder [18] to acquire high-level image features, following a stack of convolutions to colorize images. One of them is the Colorful Image Colorization (CIC) [32], which poses colorization as a classification task to make results more colorful. But limited by the content-independent interaction modeling and local inductive bias of convolutional neural network (CNN), their results have **incorrect semantic colors** (Fig. 1 CIC). To capture long-range dependency, ColTran [17] builds a probabilistic model with

[#] Equal contributions. [*] Corresponding author.

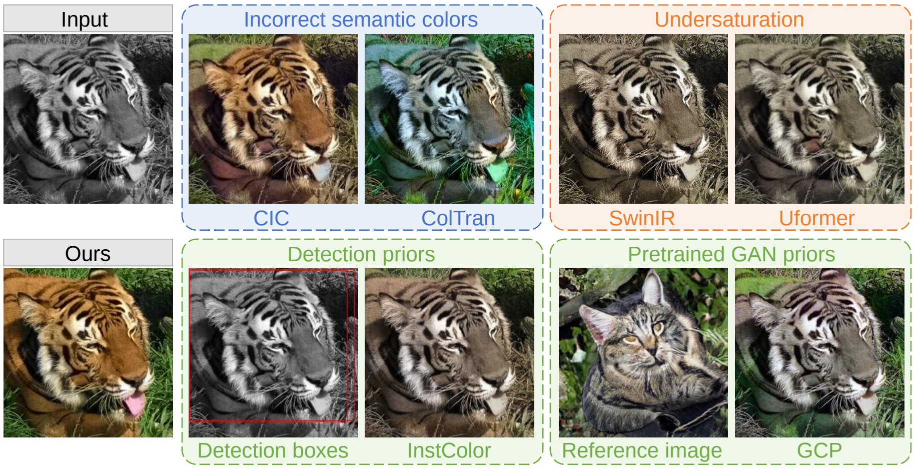


Fig. 1. *Top left*: Existing automatic colorization methods are either limited by insufficient representation ability of the network to infer colors from semantic cues (CIC [32]) or adopting staged training and aggressive sampling strategy (ColTran [17]), leading to counterintuitive colorized results. *Top right*: Advanced transformer-based image restoration methods (SwinIR [21] and Uformer [29]) produce overly conservative undersaturated results because of constructing standard regression model. *Bottom*: Methods taking external priors rely heavily on the performance of upstream models. Detection boxes as priors may be ineffective when the object covers the whole image (red boxes, 2nd column in bottom row), which limits the model performance (InstColor [26]); and pretrained GAN priors may generate inappropriate reference instances (the grey tabby cat, 4th column in bottom row), which leads to incorrect colorization (GCP [31]). We propose CT^2 to generate colorization results with reasonable semantic colors and proper saturation level without any additional external priors.

multiple transformer subnets which takes staged training and aggressive sampling strategies. However, as subnets of the ColTran are trained independently, the prediction error of each subnet will accumulate to a large one, which also leads to noticeable **incorrect semantic colors** in the final colorization results (Fig. 1 ColTran). Advanced transformer-based vision models have shown great success in image restoration, *e.g.*, SwinIR [21] and Uformer [29], benefited from their flexible receptive fields, coarse-to-fine feature expression, and end-to-end training. However, they bear **undersaturation** because the models they adopt are standard regression models, which encourage conservative predictions in the colorization task (Fig. 1 SwinIR and Uformer).

To overcome the aforementioned challenges, some researchers introduce external priors into colorization task, *e.g.*, object detection boxes [26], segmentation masks [35,36], and pretrained GANs [19,31]. However, these priors need additional data annotation or interaction with users, which may be ineffective or inaccurate in "out-of-distribution" scenarios (Fig. 1 InstColor and GCP).

In this paper, we propose **C**olorization **T**ransformer via **C**olor **T**okens (**CT^2**) to deal with incorrect semantic colors and undersaturation without any additional external priors (Fig. 1 Ours). For *(i)* **incorrect semantic colors**, we build our model based on an end-to-end transformer backbone with a newly proposed luminance-selecting module. Thanks to the long-range dependency capture ability of transformer architecture, our method copes with local nuisances in data better. In addition, the end-to-end design with the luminance-selecting module can alleviate error accumulation in staged training and avoid empirically unreasonable colors. For *(ii)* **undersaturation**, we introduce color tokens into colorization pipeline to model this task as the classification problem. We design color attention and color query modules to strengthen the interaction between grayscale image patches and color tokens, and assign vivid and plausible colors under the guidance of the luminance-selecting module.

CT^2 makes the following contributions:

- We develop an end-to-end colorization transformer model with the luminance-selecting module to generate semantically reasonable colorized images by narrowing the range of optional color candidates. Since no additional external priors are required, our model is applicable to more general scenarios.
- We propose color tokens into colorization task by color embedding module, with which colorization task could be treated as the classification problem for increasing saturation.
- We design color attention and color query modules to guide the interaction of grayscale image patches and optional color candidates, and generate more visually pleasing and plausible results than previous methods.

The experiments demonstrate that CT^2 provides higher quality colorization results both quantitatively and qualitatively, and its extensive applicability in colorizing legacy photos.

2 Related Works

Automatic colorization. Early automatic colorization methods struggle at integrating handcraft features into deep neural network [6]. With the emergence of CNN, which significantly increases the representation ability of neural network, some works [10,16,18,32] begin to pay more attention to the network architecture and fully automatic feature extraction engineering to improve colorization performance. Later, researchers experiment with multiple advanced generative models to meet the challenges in colorization. MDN [8] takes variational autoencoder (VAE) to obtain diverse colorized results. colorGAN and ChromaGAN [3,28] take generative adversarial model (GAN) to make results vivid. CINN [2] introduces an invertible neural network to avoid mode collapse benefited from bidirectional architecture. Other works focus on using external prior knowledge to optimize the colorization algorithm. InstColor [26] utilizes the detection model to localize objects, which demonstrates that the clear figure-ground separation helps performance improve. Some works [35,36] take segmentation masks as the pixel-level object semantics to guide colorization. In addition, well-pretrained GANs [19,31] are also regarded as priors by generating reference instances as the guidance of colorization.

Vision transformer for low-level problems. Transformer [27] is firstly proposed to model sequence in natural language processing. Due to its long-range receptive field, it has made a tremendous progress in solving a diversity of computer vision problems, *e.g.*, image classification [9,22], object detection [4,38], and segmentation [25,37]. The significant performance improvement appeals researchers to introduce transformer models into low-level problems, *e.g.* image restoration, and colorization task. IPT [5] jointly trains transformer blocks with multi-heads and multi-tails on multiple low-level vision tasks, by relying on a large-scale synthesized dataset. EDT [20] proposes a novel encoder-decoder architecture to make data and computation efficient. SwinIR [21] incorporates shifted window mechanism into transformer which decreases the calculated amount. Inspired by the famous CNN architecture U-Net [23], Uformer[29] proposes a multi-scale restoration modulator to adjust on multi-scale features. For colorization problem, ColTran[17] builds a probabilistic model with transformer and samples colors from the learned distribution to make results diverse.

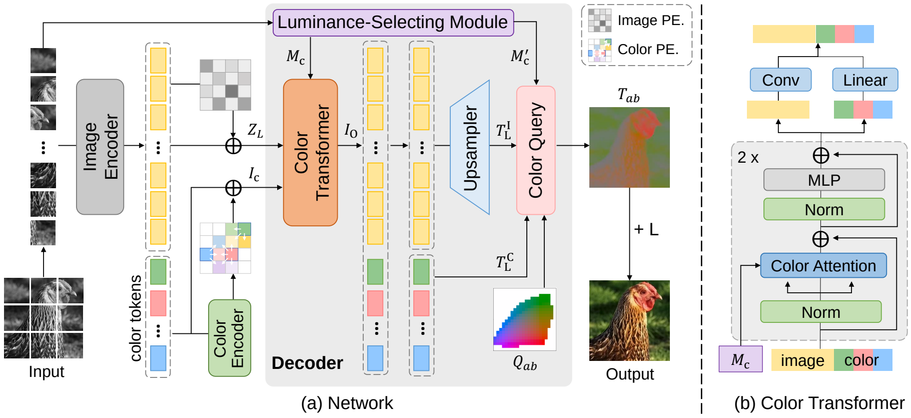
3 Method

The framework of CT^2 is composed of three core components: *(i)* an image encoder to extract grayscale image features and encode the sequence of patches into patch embeddings, *(ii)* a color encoder to acquire the relative position relationship of the defined color tokens in quantized *ab* space, *(iii)* a lightweight decoder consisting of the color transformer to interact color encodings with grayscale image features, the luminance-selecting module to narrow the range of color candidates, the upsampler to expand resolution, and the color query module to assign appropriate colors. See Fig. 2 for an overview. Next, we elaborate on the detailed designs of these modules and the losses we used for colorization.

3.1 Image Encoder

We use the standard vision transformer (ViT) [9] as our image encoder to extract long-range features of the input image. Given a single-channel grayscale image $I_{\mathrm{L}} \in \mathbb{R}^{H \times W}$, we split it into a sequence of patches $I_{\mathrm{L}} = [I_{\mathrm{L}_1}, ..., I_{\mathrm{L}_N}] \in \mathbb{R}^{N \times P^2}$, where H and W are image height and width, (P, P) is the patch size, and $N = HW/P^2$ is the number of patches. Then, with a linear projection layer, we map the input patches into a sequence of patch embeddings $I_{\mathrm{e}} \in \mathbb{R}^{N \times C}$, where C is the number of channels. To capture positional information, learnable positional embeddings $I_{\mathrm{pos}} \in \mathbb{R}^{N \times C}$ are added to the patch embeddings to get the input image tokens, written as $Z_0 = I_{\mathrm{e}} + I_{\mathrm{pos}}$.

The L-layer transformer is applied to the input tokens Z_0 to generate a sequence of contextualized embeddings Z_L. Each transformer layer consists of a multi-headed self-attention (MSA) block, an MLP block with two linear layers,



(a) Network

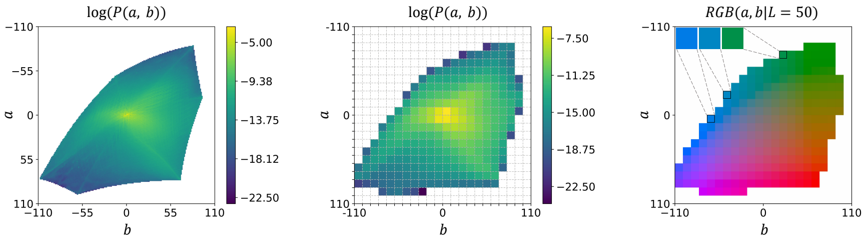
(b) Color Transformer

Fig. 2. (a) Overview of the proposed CT^2 network: The input grayscale image is split into image patches, and encoded into a sequence of tokens. The *ab* color space is quantized and extracted into multiple valid color tokens. Then all tokens are separately added with Positional Encodings (PE), and fed into the color transformer, where the color information is injected into grayscale image tokens under the guidance of the luminance-selecting module. After processed by the upsampler, the tokens are upsampled into pixel level. Color query module predicts *ab* pairs for every pixel conditioned on the luminance-selecting module. We concatenate predicted *ab* pairs with the input luminance channel to obtain colorization results. (b) The structure of the color transformer.

two LayerNorm (LN) modules, and residual connections after blocks. Finally we obtain the output $Z_L \in \mathbb{R}^{N \times C}$, a sequence containing rich long-range image semantics, which is added with the conditional positional encodings [7] and fed into the decoder (Sec. 3.3) as image features.

3.2 Color Encoder

The colorization task aims to learn the mapping from the input luminance channel L to the two associated color channels ab, which is performed in CIE Lab color space. Following CIC [32], we take samples in the training set to calculate the empirical probability distribution of ab values in ab color space (Fig. 3 (a)). The distribution reveals the preference of ab pairs in natural images, *e.g.*, low saturation ab pairs (a, b values close to 0) are used more frequently while colorful ab pairs only appear in a few samples. Thus, if the model penalizes all ab pairs equally during training, the model is not capable to produce colorful results due to the dominance of low-saturation samples, resulting in undersaturated results. In addition, constructing a regression model to solve the colorization problem will produce average results, which tends to colorize images with insufficient saturation depending on empirical distribution (frequent low-saturation samples).



(a) (b) (c)

Fig. 3. Illustration of quantized *ab* space. (a) Statistics of empirical distribution in the training set. We show probability distribution in log scale, where darker colors represent higher probabilities. (b) 484 quantized color patches. The probability of each patch is the mean value of the 10×10 sliding window. Note that only 313 color patches are valid. (c) An example shows that the closer patches are in *ab* color space, the more similar their colors represent, and vice versa.

Hence, we introduce *color tokens* into CT^2 to formulate the colorization task as a classification problem to mitigate undersaturation.

We use the sliding window of size 10 and stride 10 to divide the *ab* color space into 484 color patches, and calculate the mean probability distribution in each color patch (Fig. 3 (b)). Considering there are some color patches that never appear in empirical distribution (the white patches shown in Fig. 3 (b)), we filter them out and encode the remaining valid 313 color patches by assigning a randomly initialized learnable vector to each patch in color embedding module. We define the embedded color patches as color tokens $I_c \in \mathbb{R}^{313\times C}$. Considering that the closer the color tokens are in the quantized *ab* color space, the more similar the color properties are (*e.g.*, blue is more similar with cyan than green, as shown in Fig. 3 (c)), we add positional information into color tokens by applying conditional positional encodings [7]. Specifically, color tokens are reshaped and zero-padded into spatial 2D quantized *ab* space following a convolution with 3×3 kernel, which provides relative position information and constrains the similarity between adjacent vectors in the embedded feature space. Then, we flatten the output of convolution back into the sequence as color positional encodings, which are added into the original color tokens. Finally, we obtain the embedded color tokens $I_c \in \mathbb{R}^{313\times C}$, as color features to feed into the decoder.

3.3 Decoder

The decoder is designed to interact grayscale image features Z_L with color features I_c, and finally generate colorful images, which is the key component of CT^2. As shown in Fig. 2 (a), the decoder first calculates the color mask M_c with the luminance-selecting module to construct the mapping from luminance L to the optional range of color tokens, which is used as the guidance in the following color transformer and color query module. Then the image features Z_L and color

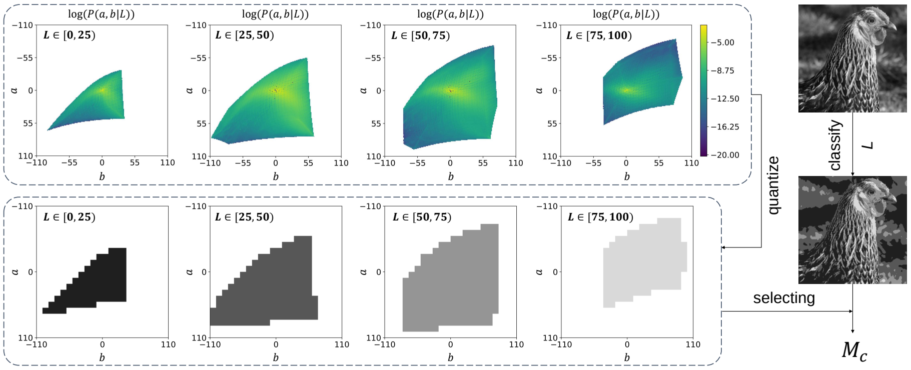


Fig. 4. The process of calculating color mask M_c with the luminance-selecting module. The top row illustrates the empirical probability distribution of ab values conditioned on L, and the bottom row illustrates the valid quantized ab patches corresponding to the different probability distributions.

features I_c are fed into the color transformer module, where color information is injected into grayscale image features. After that, the colorized patch-level features are upsampled to pixel-level in the upsampler module. Finally, the color query module calculates pixel-level color scores and predicts the reasonable color for each pixel. Next, we describe the proposed modules in the decoder in detail.

Luminance-selecting module. By observing that ab distribution varies with luminance L, we split L value into 4 non-overlapping intervals, and show the ab empirical probability distribution conditioned on different L ranges in the top row of Fig. 4. Thus we can reduce the optional quantized ab patches according to the corresponding empirical distribution, and further improve the accuracy of model prediction to avoid generating incorrect semantic colors. Specifically, we first classify the L value of the input image into 4 groups, the same as the 4 aforementioned non-overlapping intervals, which are expressed with 4 varying degrees of gray levels. Then we quantize the ab probability distribution and obtain the quantized ab patches conditioned on L, as shown in the bottom row of Fig. 4. Finally, conditioned on the classified L of the input image, we select the corresponding quantized ab patches and construct a one-hot mask $M_\mathrm{c} \in \mathbb{R}^{H \times W \times 313}$, denoted as *color mask*, where we set the indices of optional ab patches among 313 classes as 1 and otherwise 0 for every pixel. Based on empirical distribution statistics in the training set, color mask rules out the rare strange colorization predictions and further reduces the ambiguity of colorization.

Color transformer. The color transformer is proposed to inject color information into grayscale image features, which is composed of two transformer layers and a following projection module. The grayscale image tokens after the encoder and the embedded color tokens are concatenated firstly, and then injected into the color transformer as a whole input sequence. The transformer layer is modified from the standard version [9] by replacing the multi-headed self-attention with the color attention which we will explain later. The projection module is designed for image features and color tokens respectively, where the conventional 3×3 convolution is used to the reshaped image features after the last transformer layer, and a fully connected layer is applied to color tokens I_{c}, as shown in Fig. 2 (b). Finally, we concatenate the refined image features and color tokens into a sequence as the output of the color transformer, denoted as $I_{\mathrm{O}} \in \mathbb{R}^{(N+313)\times C}$.

Color attention. We propose the color attention module to bridge the gap between color tokens and image features. Specifically, color attention is essentially a masked multi-headed self-attention, which realizes the color-image interaction and injects color information into gray-scale image features under the guidance of the patch mask. To clearly illustrate it, we first describe the design of the patch mask which limits the scope of color-image interaction, and then illustrate the process of performing color attention.

Similar to the input image, we split the color mask M_{c} into a sequence of patches $M_{\mathrm{c}} = [M_{\mathrm{c}_1}, ..., M_{\mathrm{c}_N}] \in \mathbb{R}^{N\times P^2 \times 313}$. For each color mask patch M_{c_i}, the model calculates the corresponding union set of the P^2 pixel values, and then concatenates all union sets as follows:

$$I_{\mathrm{M}} = \mathrm{Concat}_{i\in\{1,...,N\}} \cup_{j\in\{1,...,P^2\}} M_{\mathrm{c}_{i,j}}, \tag{1}$$

where $M_{\mathrm{c}_{i,j}} \in \mathbb{R}^{313}$ denotes the binary mask corresponding to the j-th luminance value in the i-th image feature patch, and $I_{\mathrm{M}} \in \mathbb{R}^{N\times 313}$ represents the *patch mask* which indicates inappropriate color tokens for patch-level image features. Next, considering the input sequence is the concatenation of image patch tokens and color tokens, we compose the patch mask I_{M}, the transpose of patch mask ${I_{\mathrm{M}}}^{\top}$, and two all-1 matrices, denoted as $\mathbb{1}^{N\times N}$ and $\mathbb{1}^{313\times 313}$, into the *attention mask* $I'_{\mathrm{M}} \in \mathbb{R}^{(N+313)\times(N+313)}$, as follows:

$$I'_{\mathrm{M}} = \begin{bmatrix} \mathbb{1}^{N\times N} & I_M \\ {I_M}^{\top} & \mathbb{1}^{313\times 313} \end{bmatrix}. \tag{2}$$

To rule out the unreasonable color tokens in color attention, we convert I'_{M} into another binary mask $M \in \mathbb{R}^{(N+313)\times(N+313)}$, where we set the value to $-\infty$ corresponding to indicate undesirable color tokens and otherwise 0:

$$M = \begin{cases} 0 & \text{where } I'_{\mathrm{M}} = 1 \\ -\infty & \text{where } I'_{\mathrm{M}} = 0 \end{cases}, \tag{3}$$

After that, the binary mask M is utilized in the masked multi-headed self-attention to obtain the refined features:

$$\mathrm{ColorAttention}(Q, K, V, M) = \mathrm{Softmax}\left(M + \frac{QK^{\top}}{\sqrt{C}}\right)V, \tag{4}$$

where $Q, K, V \in \mathbb{R}^{(N+313)\times C}$ denote query, key, and value respectively, which are obtained from LayerNorm and MLP blocks processing the concatenation of image features $Z_L \in \mathbb{R}^{N\times C}$ and color tokens $I_{\mathrm{c}} \in \mathbb{R}^{313\times C}$, note that both of them are added with positional encodings. C is the embedding dim of Q, K, V.

Upsampler. The upsampler is only applied on image patch tokens, which are separated from the output sequence I_{O} of the color transformer. The progressive upsampler is made up of 4 upsampling blocks, realizing 16 times of upsampling to achieve user-desired resolution. Each block is a stack of a BatchNorm, two ReLU functions, a conventional 3×3 convolution, and a 4×4 transposed convolution with a stride of 2 to extract features and extend spatial resolution.
Color query. We design the color query module to estimate the correct semantic color for each image pixel and generate colorful results. Given the upsampled image features $T_{\mathrm{L}}^{\mathrm{I}} \in \mathbb{R}^{HW \times C}$ and refined color tokens $T_{\mathrm{L}}^{\mathrm{C}} \in \mathbb{R}^{313 \times C}$ separated from I_{O}, the color query module calculates the cross product between the ℓ_2-normalized $T_{\mathrm{L}}^{\mathrm{I}}$ and $T_{\mathrm{L}}^{\mathrm{C}}$ under the guidance of the color mask M_{c}, where we also set the value to $-\infty$ for indices of inappropriate color tokens, and otherwise 0, denoted as $M_{\mathrm{c}}' \in \mathbb{R}^{HW \times 313}$. We formulate the process as follows:

$$\hat{I}_{\mathrm{q}} = \mathrm{softmax}(\|T_{\mathrm{L}}^{\mathrm{I}}\|_2 \|T_{\mathrm{L}}^{\mathrm{C}}\|_2^{\top} + M_{\mathrm{c}}'), \tag{5}$$

where $\hat{I}_{\mathrm{q}} \in \mathbb{R}^{HW \times 313}$ is the probability distribution of the 313 color candidates. We utilized the predicted probability as the weight to summarize the quantized *ab* pairs $Q_{\mathrm{ab}} \in \mathbb{R}^{313 \times 2}$ to finally obtain suitable colorized *ab* values, written as:

$$T_{\mathrm{ab}} = \hat{I}_{\mathrm{q}} \cdot Q_{\mathrm{ab}}. \tag{6}$$

The final *Lab* image I_{Lab} is obtained by the concatenation of input grayscale image and estimated *ab* values, written as $I_{\mathrm{Lab}} = \mathrm{Concat}(I_{\mathrm{L}}, T_{\mathrm{ab}})$.

3.4 Optimization

Losses. We treat the colorization problem as the pixel-wise classification task to alleviate undersaturation, thus we optimize our model by minimizing the cross entropy loss L_{cl}. We quantize the *ab* space into 313 color candidates, and obtain the probability distribution $\hat{I}_{\mathrm{q}} \in \mathbb{R}^{H \times W \times 313}$ over optional colors as the model prediction. To compare the prediction with the ground truth, we convert the ground truth I_{ab} into the quantized *ab* space, denoted as $I_{\mathrm{q}} \in \mathbb{R}^{H \times W \times 313}$. Specifically, for every pixel, we find 5-nearest neighbors to I_{ab} among quantized *ab* pairs, and calculate their distance from I_{ab} as the weight to proportionally construct the normalized soft label I_{q}. The classification loss is formulated as:

$$L_{\mathrm{cl}} = -\sum_{x,y,q} (\log(\hat{I}_{\mathrm{q}}(x,y,q)) - \log(I_{\mathrm{q}}(x,y,q))) I_{\mathrm{q}}(x,y,q), \tag{7}$$

where (x, y) is the location in images, q is the index of quantized color candidates.

In addition, following Zhang *et al.* [34], a smooth-ℓ_1 loss with $\delta = 1$ is adopted to make the training process stable and reduce overly saturated color candidates:

$$L_{\delta}\left(T_{\mathrm{ab}}, I_{\mathrm{ab}}\right) = \frac{1}{2}(T_{\mathrm{ab}} - I_{\mathrm{ab}})^2 \mathbb{1}_{\{|T_{\mathrm{ab}} - I_{\mathrm{ab}}| < \delta\}} + \delta(|T_{\mathrm{ab}} - I_{\mathrm{ab}}| - \frac{1}{2}\delta) \mathbb{1}_{\{|T_{\mathrm{ab}} - I_{\mathrm{ab}}| \geq \delta\}}, \tag{8}$$

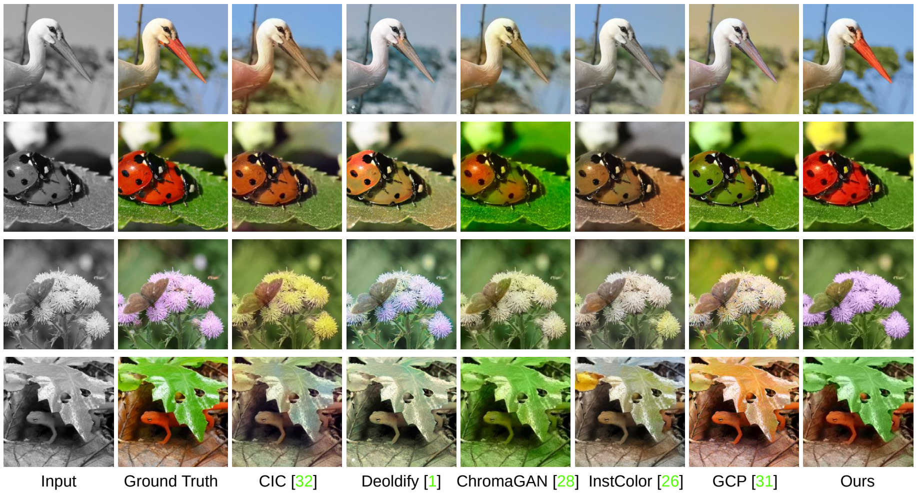


Fig. 5. Comparisons with CNN-based methods. Our method is superior to other comparison methods on semantic color inference, *e.g.*, the beak of the crane (first row) and the ladybug shell (second row). Our method also outperforms other comparison methods on generating colorful results, *e.g.*, flowers (third row) and geckos (last row).

where I_{ab} is the *ab* channels of ground truth images.

Finally, our loss function L_{total} is a combination of L_{cl} and L_{δ}, which can be jointly optimized as follows:

$$L_{\text{total}} = \alpha L_{\text{cl}} + \beta L_{\delta}, \tag{9}$$

where we set α and β as 1 and 10, respectively.

4 Experiments

Dataset. We conduct our experiments on ImageNet [24], which contains 1.3M images covering 1000 categories. We test on the first 5k images of the public validation set, which is consistent with the previous methods [2,17]. All the test images are center cropped and resized into 256 × 256 resolution.

Metrics. We report 6 quantitative metrics in Tab. 1, including Peak Signal-to-Noise Ratio (PSNR) [15], Structural Similarity Index (SSIM) [30], Learned Perceptual Image Patch Similarity (LPIPS) [33], Fréchet inception distance [14], and 2 colorfulness score [12] to reflect the vividness following GCP [31].

Implementation details. For transformer encoder and decoder, we keep the embedding dim of the MLP block 4 times as the hidden size of the attention block. The input patch size is fixed to 16 × 16. The image encoder is initialized with the pretrained ViT [9] weights, and the color transformer in the decoder is initialized with random weights from a truncated normal distribution [11].

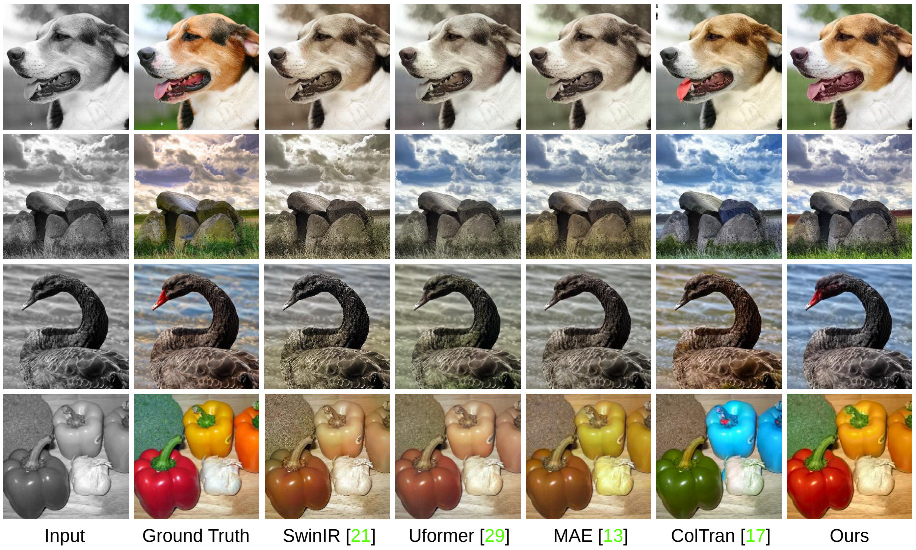


Fig. 6. Comparisons with transformer-based methods. Different from SwinIR [21], Uformer [29], and MAE [13], our method generates saturated colorized images, *e.g.*, the dog (first row) and the landscape scenes (second row). Our method could also generate correct semantic colors by avoiding error accumulation observed on ColTran [17], *e.g.*, the swan (third row) and vegetables (last row).

The details of configurations about layers, hidden size, the number of heads in attention blocks of our model are shown in the supplemental materials.

Training details. We set the batch size to 16 and minimize our objective losses using SGD optimizer and polynomial learning rate schedule. We set the learning rate to 10^{-3} and momentum parameter to 0.9. All experiments are conducted on 8 NVIDIA GeForce RTX 3090 graphic cards and trained for 10 epochs.

4.1 Comparisons with Previous Methods

We make comparisons with 5 CNN-based methods, including CIC [32], DeOldify [1], ChromaGAN[28], InstColor [26], and GCP [31] to show our transformer-based architecture has powerful feature representation ability by capturing long-range dependencies. Note that ChromaGAN [28], InstColor [26], and GCP [31] use additional external priors, while ours without any prior.

We also compare our method with 4 advanced transformer-based methods, including: *(i)* two state-of-the-art image restoration approaches, SwinIR [21] and Uformer [29], by retraining models on the colorization task; *(ii)* the state-of-the-art self-supervised learner MAE [13], by finetuning its pretrained weights to colorization as a downstream task; and *(iii)* the state-of-the-art colorization methods ColTran [17] with same experiment settings.

Table 1. Quantitative comparison results. ↑ (↓) means higher (lower) is better.

Category	Method	FID↓	PSNR↑	SSIM↑	LPIPS↓	colorful↑	△colorful↓
CNN	CIC [32]	8.72	22.64	0.91	0.22	31.60	4.72
	DeOldify [1]	9.45	21.12	0.83	0.24	22.70	13.62
	ChromaGAN [28]	7.66	23.35	0.90	0.21	27.88	8.43
	InstColor [26]	8.06	23.28	0.91	0.21	24.87	11.44
	GCP [31]	5.95	21.68	0.88	0.23	32.98	3.34
Transformer	SwinIR [21]	12.26	21.54	0.78	0.31	16.57	19.75
	Uformer [29]	10.09	22.82	0.86	0.22	17.98	18.33
	MAE [13]	9.45	23.35	0.87	0.21	20.60	15.72
	ColTran [17]	6.44	20.95	0.80	0.29	34.50	2.24
Ours	CT^2	**5.51**	**23.50**	**0.92**	**0.19**	**38.48**	**2.17**

Table 2. User study results. Ours achieves obviously higher score than other methods.

CIC [32]	DeOldify [1]	ChromaGAN [28]	InstColor [26]	GCP [31]
3.02%	3.64%	7.72%	9.16%	15.14%
SwinIR [21]	Uformer [29]	ColTran [17]	MAE [13]	Ours
6.84%	5.48%	6.92%	2.80%	**39.28%**

Quantitative comparisons. We show the quantitative comparisons in Tab. 1, where our method achieves state-of-the-art performance on all metrics. The best scores on FID, PSNR, SSIM, and LPIPS demonstrate our method colorizes images with correct semantic colors. The significant leadings in colorfulness metrics show that our method overcomes the undersaturation challenge.

Qualitative comparisons. The qualitative comparisons demonstrate the effectiveness of our method. We show comparisons with CNN-based methods in Fig. 5. Thanks to the global interaction between features and the strong feature representation ability of transformer, our method could colorize images visually pleasing. We show comparisons with transformer-based methods in Fig. 6. Benefited from our proposed modules for colorization task, we could treat colorization as a classification task, which alleviates undersaturation appeared in other methods. In addition, the end-to-end transformer design avoids error accumulation, resulting in more plausible colors compared with ColTran [17] results.

4.2 User Study

In addition to quantitative and qualitative comparisons, we further conduct user study experiments to evaluate whether our results are favored by human observers. We provide a grayscale image and colorized images from 10 different methods: CIC [32], DeOldify[1], ChromaGAN [28], InstColor [26], GCP [31], SwinIR [21], Uformer [29], MAE [13], ColTran [17] and ours. Participants are asked to choose the most visually pleasing result with respect to the ground

Table 3. Quantitative ablation results. ↑ (↓) means higher (lower) is better.

Category	Method	FID↓	PSNR↑	SSIM↑	LPIPS↓	colorful↑	△colorful↓
Ablation	W/o LSM	7.51	20.99	0.82	0.26	**41.56**	5.24
	W/o color attention	7.76	20.93	0.82	0.27	39.53	3.22
	W/o color query	8.87	21.70	0.90	0.23	39.51	3.19
Ours	CT^2	**5.51**	**23.50**	**0.92**	**0.19**	38.48	**2.17**

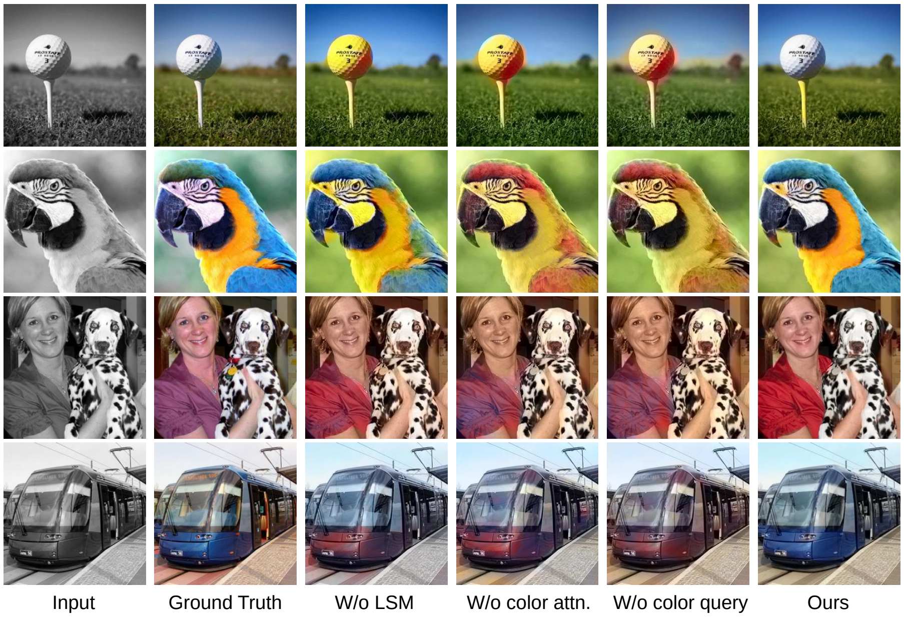
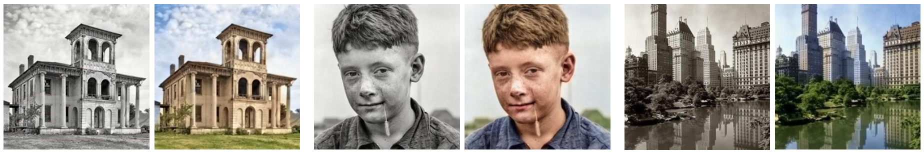


Fig. 7. Ablation study. The results become counterintuitive when our proposed modules are disabled.

truth. The experiment set is composed of 100 synthetic images that are randomly selected from the testing set. We publish the experiments on Amazon Mechanical Turk (AMT), and each experiment is completed by 25 participants. We present the results of user study in Tab. 2, where our method outperforms other comparison methods, confirming its subjective advantages.

4.3 Ablation Study and Discussion

We disable various modules and create three baselines to study the impact of our proposed modules. We show the evaluation scores and colorized images of the ablation study experiments in Tab. 3 and Fig. 7, respectively. The colorfulness metrics (fifth column) of these ablation baselines are higher than ours, which is



1939. "Drish House" by Frances Benjamin Johnston.

1939. "Cow boy" by Arthur Rothstein.

1933. "Manhattan Central Park in New York" by Samuel Gottscho.

Fig. 8. Applying our method to legacy black and white photos.

probably because the counterintuitive and mixed colors are misjudged as vivid colors by this metric, which we will explain next.

W/o LSM. We disable the luminance-selecting module in both color attention and color query modules to study the effectiveness of narrowing optional color tokens. After the range of color candidates is expanded to include colors not in the empirical distribution, the colorized results become counterintuitive, *e.g.*, the golf ball and the parrot (first and second row in Fig. 7).

W/o color attention. We replace color attention with standard self-attention between image feature patches. In this way, the model cannot correctly infer semantic colors, therefore the results present mixed colors, *e.g.*, the woman and the tram (third and last row in Fig. 7).

W/o color query. We replace the color query module with an MLP block as the classifier. As a result, the ability to infer colors from image semantics reduces, which makes the model prediction blurred, and causes mixed colors, *e.g.*, the woman and the tram (third and last row in Fig. 7).

4.4 Application

We apply our method to colorize the legacy black and white photos shown in Fig. 8, which demonstrates the generalization capability of our proposed method.

5 Conclusion

We propose **C**olorization **T**ransformer via **C**olor **T**okens (CT^2), to deal with existing incorrect semantic colors and undersaturation challenges without additional priors. To demonstrate its effectiveness, we make comparisons with the 9 state-of-the-art methods, and the experiment results show that our method achieves highest scores on 4 image quality metrics and 2 colorfulness metrics.

Limitation. We need to calculate the empirical distribution on the training set to narrow the color candidates. Therefore, our method may degenerate if the training data are insufficient or have a clear bias. Fortunately, ImageNet [24] includes 1.3M training data and covers 1000 categories, which to some extent prevents this problem from happening in our experiments.

Acknowledgements. This project is supported by National Natural Science Foundation of China under Grant No. 62136001.

References

1. Antic, J.: A deep learning based project for colorizing and restoring old images (and video!), https://github.com/jantic/DeOldify 11, 12
2. Ardizzone, L., Lüth, C., Kruse, J., Rother, C., Köthe, U.: Guided image generation with conditional invertible neural networks. arXiv (2019) 3, 10
3. Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: ECML-PKDD (2017) 3
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020) 4
5. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: CVPR (2021) 4
6. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: ICCV (2015) 1, 3
7. Chu, X., Zhang, B., Tian, Z., Wei, X., Xia, H.: Do we really need explicit position encodings for vision transformers? arXiv (2021) 5, 6
8. Deshpande, A., Lu, J., Yeh, M.C., Jin Chong, M., Forsyth, D.: Learning diverse image colorization. In: CVPR (2017) 3
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 4, 7, 10
10. Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., Murphy, K.: Pixcolor: Pixel recursive colorization. arXiv (2017) 3
11. Hanin, B., Rolnick, D.: How to start training: The effect of initialization and architecture. In: NIPS (2018) 10
12. Hasler, D., Suesstrunk, S.E.: Measuring colorfulness in natural images. In: Human vision and electronic imaging VIII (2003) 10
13. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv (2021) 11, 12
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a nash equilibrium. In: NIPS (2017) 10
15. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electron. Lett. (2008) 10
16. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ToG (2016) 3
17. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. In: ICLR (2021) 1, 2, 4, 10, 11, 12
18. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: ECCV (2016) 1, 3
19. Lei, C., Wu, Y., Chen, Q.: Towards photorealistic colorization by imagination. arXiv (2021) 2, 4
20. Li, W., Lu, X., Lu, J., Zhang, X., Jia, J.: On efficient transformer and image pre-training for low-level vision. arXiv (2021) 4
21. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: ICCV (2021) 2, 4, 11, 12
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV (2021) 4

23. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 4
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015) 10, 14
25. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: ICCV (2021) 4
26. Su, J.W., Chu, H.K., Huang, J.B.: Instance-aware image colorization. In: CVPR (2020) 2, 3, 11, 12
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017) 4
28. Vitoria, P., Raad, L., Ballester, C.: Chromagan: Adversarial picture colorization with semantic class distribution. In: WACV (2020) 3, 11, 12
29. Wang, Z., Cun, X., Bao, J., Liu, J.: Uformer: A general u-shaped transformer for image restoration. arXiv (2021) 2, 4, 11, 12
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004) 10
31. Wu, Y., Wang, X., Li, Y., Zhang, H., Zhao, X., Shan, Y.: Towards vivid and diverse image colorization with generative color prior. In: ICCV (2021) 2, 4, 10, 11, 12
32. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016) 1, 2, 3, 5, 11, 12
33. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 10
34. Zhang, R., Zhu, J.Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. ACM TOG (2017) 9
35. Zhao, J., Han, J., Shao, L., Snoek, C.G.: Pixelated semantic colorization. IJCV (2020) 2, 3
36. Zhao, J., Liu, L., Snoek, C.G., Han, J., Shao, L.: Pixel-level semantics guided image colorization. In: BMVC (2018) 2, 3
37. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021) 4
38. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2020) 4