Supplementary Materials for "Guiding Users to Where to Give Color Hints for Efficient Interactive Sketch Colorization via Unsupervised Region Prioritization"

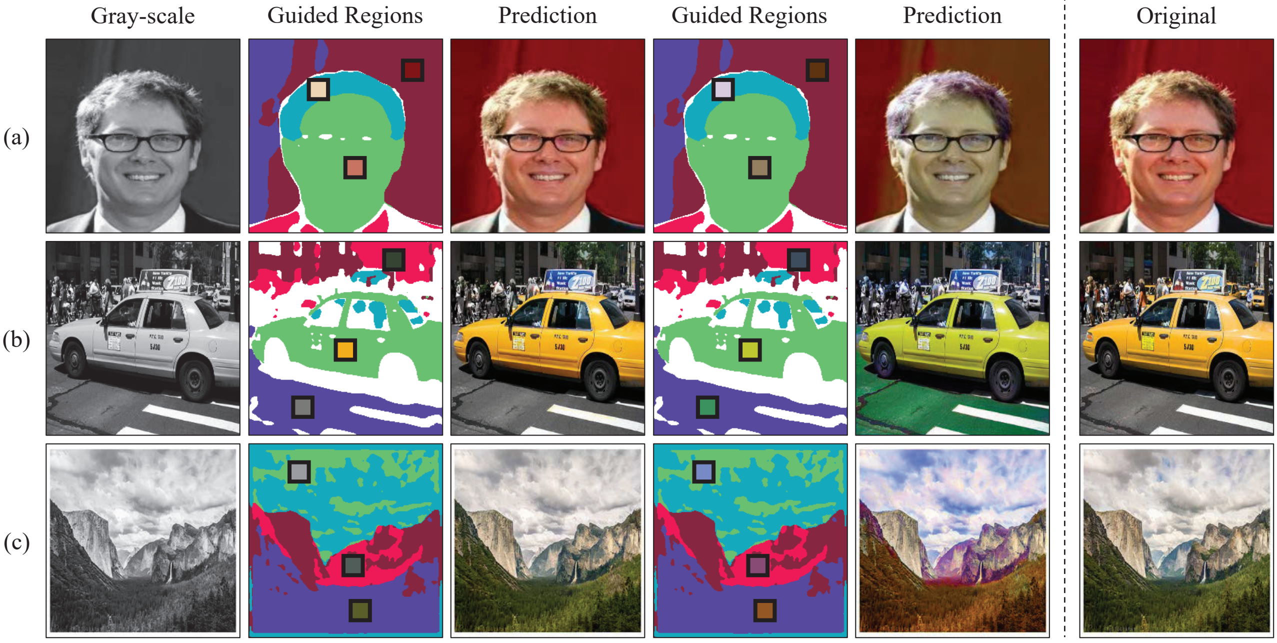

Figure 1: **Qualitative results of gray-scale colorization on (a) CelebA, (b) Imagenet Car, and (c) Summer2Winter Yosemite datasets.** First column represents gray-scale inputs. Second and fourth columns show first five guided regions generated by the segmentation network and mark regions to change the color as small squares. Third column shows the colorization results when reflecting seven original color hints and sixth column shows the colorization results when changing the color of the marked regions.

6. Overview

This document addresses with additional information that we does not cover in our main paper due to the page limit. Section 7 show the applicability of our approach to a gray-scale image on diverse datasets. Section 8 describes the implementation details for the reproducibility, including the network architectures, the hyperparameters of optimizers we used, and the training details. Section 9 provides discussions on our approach, including our hinting mechanism, a method to change colorization order, and limitation of our method. Section 10 gives detailed descriptions of the user studies we conducted in Section 4.4 of our main paper. At the rest of this document (Section 11), we demonstrate qualitative results of our proposed method. Note that an attached video material could help to understand the behaviors of GuidingPainter in a visual manner, and source codes are provided for needs of code-level details.

7. Application to Gray-scale Colorization

We test our approach using the gray-scale input on CelebA [6], Imagenet Car [1], and Summer2Winter Yosemite [11] dataset. To do this, we modify the colorization model to take the L channel of an image and output the AB channel of the image. Table 1 represents the quantitative results of RTUG and our model on each dataset. Our model surpasses the baseline in terms of both PSNR and FID. This demonstrates that our model can produce realistic images while reflecting color hints in grayscale colorization. As shown in Fig. 1, our model can guide the color hints to reasonably shaped regions and reflect the color hints.

8. Implementation Details

U-Net Architecture. We adopt the U-Net architecture in the segmentation network and colorization network, except for the size of channels in the input layer and output layer. The layer specification is shown in Table. 2, where (C_{in}, C_{out}) is equal to $(1, N_c)$ for seg-

	$\text{PSNR}_{\uparrow}$ / $\text{FID}_{\downarrow}$		
	CelebA	Imagenet Car	Yosemite
RTUG	29.92 / 3.61	26.40 / 31.90	27.58 / 57.97
Ours	**30.60 / 2.40**	**26.97 / 27.54**	**27.94 / 54.05**

Table 1: **Quality comparison of grayscale colorization results** in terms of PSNR, FID. We provide RTUG and our model the same amount of color hints which follows $\mathcal{G}$.

Label	Layer
E_1	DoubleConv($I : C_{in}$,$O : 64$)
E_2	DoubleConv($I : 64$, $O : 128$)
E_3	DoubleConv($I : 128$, $O : 256$)
E_4	DoubleConv($I : 256$, $O : 512$)
D_0	DoubleConv($I : 512$, $O : 512$)
D_1	DoubleConv($I : 1024$, $O : 256$)
D_2	DoubleConv($I : 512$, $O : 128$)
D_3	DoubleConv($I : 256$, $O : 64$)
D_4	DoubleConv($I : 128$, $O : 64$)
Out	Conv($I : 64$, $O : C_{out}$)

Table 2: **The layer specification of the U-Net.** DoubleConv denotes two consecutive Conv-BatchNorm-ReLU blocks, and Conv denotes a convolution layer. I, O denote the size of input channels, the size of output channels, respectively.

mentation network and equal to $(N_c + 5, 3)$ for colorization network. Maxpooling is applied to the front of each layer E_2-D_0 to downsample the input tensor by a factor of 2. For each $i = 1...4$, the bilinear upsampled output of D_{i-1} and the output of E_{5-i} are concatenated in the channel dimension, and then pass through D_i. Every convolution in E_1-D_4 is applied with 3×3 kernel, whereas the convolution in output layer is applied with 1×1 kernel.

Discriminator. The discriminator D is implemented with PatchGAN[3], which outputs a 30×30 tensor. We use the LSGAN[7] objective to train the GAN architecture.

Training Details. We initialize the weight of networks from the normal distribution with a mean of 0 and a standard deviation of 0.02. The Adam optimizer [5] with $\beta_1 = 0.5, \beta_2 = 0.999$ is used to train our networks on all datasets. The learning rate is fixed at 0.0002 for the first half of epochs and linearly decays to zero for the remaining half of epochs. We schedule the temperature τ of ST gumbel estimator with exponential policy $\tau = 0.1^{\text{current epoch/total epochs}}$, adopted from RelGAN [9]. The total numbers of epochs for Yumi's Cells, Tag2Pix, and CelebA datasets are 500, 30, and 20, respectively. The optimization typically takes about 1-2 days on 4 TITAN RTX GPUs.

9. Discussions

9.1. RoI-based Hinting Mechanism

In this study, we mainly compare our RoI-based hinting mechanism with the point and scribble-based hinting mechanisms. Note that there are trade-offs between each method. Point or scribble-based approaches have their own advantages in that they can utilize the location of the color hint. However, our goal is to colorize images with a few hints, not to colorize perfectly with plenty of time. We therefore focus on validating the effectiveness of our region-based guidance system in terms of interaction efficiency. As shown in the user study (Section 4.4 of our main paper), our region-based guidance system can reduce the average time per interaction, resulting in the improved convenience score.

Since artists spend a lot of time in adding a base color on a sketch image(s) in real-world applications, it is valid work to find efficient method to mitigate such labour-intensive process. In this context, our work will be able to make the labour-intensive process significantly efficient. Although accurately and efficiently colorizing more complex images is still difficult, we expect that combining our RoI hinting mechanism with the point or scribble hinting mechanism would be one of promising works to solve this problem.

9.2. How to Change Colorization Order

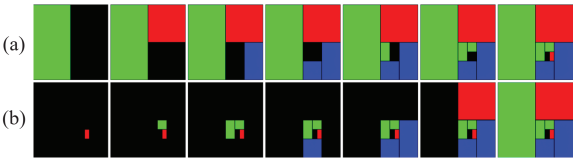

Figure 2: **Colorization results of a Dark snail example in two different colorization orders.** (a) shows the colorization results when hinting regions in decreasing order of its size. Conversely, (b) shows the colorization results when hinting the small region first.

While GuidingPainter automatically guides color hints to regions in an efficient order, the users may want to paint the regions regardless of this fixed order. We found that it is possible to change the colorization order of GuidingPainter through two-stage learning. After training GuidingPainter with the ordinary learning process, the segmentation networks gain the ability to estimate regions from a given sketch image.

To make the colorization order changeable, we train only the colorization networks by fixing $N_h = N_c$ and randomly dropping out some hints produced by the hint generation module, i.e., letting each m_i Bernoulli random variable with success probability $p = 0.125$, in a second learning phase. As a result of this learning, the colorization networks can colorize the sketch image even if only some random regions are given color hints in random order. As shown in Fig. 2, our modified GuidingPainter can colorize a sketch image in different colorization orders.

9.3. Effectiveness of the Number of Hints

We investigate how the performance of our model and baseline models changes as the number of hints increases. Fig. 6 shows the change of PSNR and FID score when each of 2,4,6,8,10,12 hints are provided to each model. We found that GuidingPainter mostly surpasses performances of the baselines if the same size of hints are given. If more than or equal to two color hints are given, our model surpasses other baseline models in both PSNR and FID scores. The results show that our hint guidance method enables the colorization module to effectively reflect hints.

9.4. Limitation of Our Study

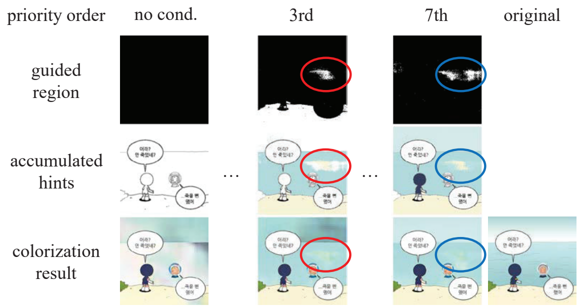

Figure 3: **A failure case** of the segmentation network.

In some cases, the quality of the result image is degraded due to wrong prediction of the segmentation network. In Fig. 3, the region marked as red shows a misaligned segment of the 'sand' region inside the 'sea' one. We found that these minor errors can be slightly refined by the following colorization network. Despite its self-correction, the stain still remains on the result, which is marked as blue. Mitigating this problem through segmentation correction techniques would be one of promising future works.

10. User Study

This section describes details of the user study in Section 4.4 of our main paper. In addition, we conduct user-perception study to evaluate whether our model can reflect unusual color hints.

10.1. Efficiency of Interactive Process

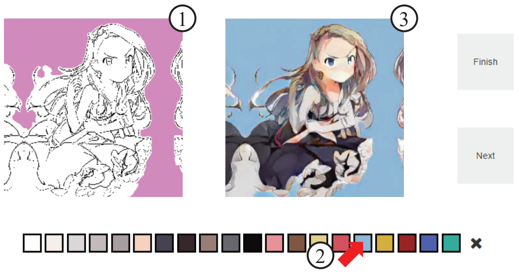

Figure 4: **User interface of user study.** (1) Our model highlights recommended region to colorize. (2) Considering the region, a user selects a color from color palettes. (3) The result of reflecting the hint is displayed.

Since our framework aims to increase the interaction-efficiency on colorization process, we conduct a user study to estimate how our model can enhance the overall process when the users intervene in it. As a competitive approach to our framework for estimating interaction-efficiency, we choose RTUG since the interaction process presented in this original paper [10] is most similar to ours and we can easily quantify the amount of interactions in the process.

As shown in Fig. 4, we develop a straightforward user interface(UI) to rule out peripheral variables except for the main algorithms as much as possible. The UI consists of a color palette, screens for checking hints and colorization results, and a few buttons to reflect hints or to select next hints. The users test our method and RTUG on three datasets, Yumi's Cells, Tag2pix, and CelebA. If the user tests our method, the user can see a region guided by our mechanism in the left image (Fig. 4 (1)), and choose a color from the palette at the bottom of the screen (Fig. 4 (2)). After selecting the color, the user can see an inference image on the right of the screen (Fig. 4 (3)). This overall process is same as when the user tests RTUG, except the facts that the location of the hint is displayed in the form of a point on the left image and the user can move the point to the location by clicking the left image. The user can click the next button to add another color hint or click the finish button to end up the colorization process.

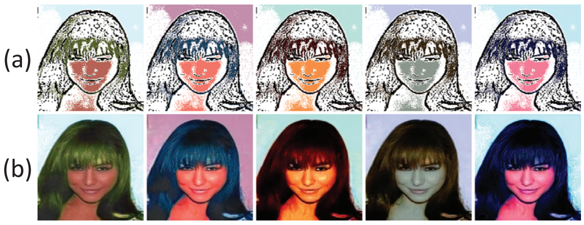

Figure 5: **Results of reflecting diverse color hints.** (a) represents accumulated hint images when giving diverse color hints to face, hair, and background. (b) shows the output images of our model for each hint image.

	Top-1$_\uparrow$ / Top-2$_\uparrow$		
	Yumi's Cells	Tag2pix	CelebA
AlacGAN	2.7 / 12.7	0.0 / 10.0	1.4 / 7.3
RTUG	20.0 / 90.0	13.6 / 91.8	29.1 / **97.7**
Ours	**77.3** / **97.3**	**86.4** / **98.2**	**69.5** / 95.0

Table 3: **User-perception study results** to evaluate how faithfully models reflect user-provided conditions. 'Top-k' indicates how many the generated images are ranked within the top-k among models over three datasets. All numbers are in percentages.

Before the evaluation of participants, we let the users freely use the UI for about 5 minutes so that the users can be familiar with it. A total of 13 participants comprised of researchers or engineers related to computer science and AI attend our user study. Each user is asked colorizing the given sketch image as naturally as possible without a reference image. For each dataset and method, the user completes three images using the UI. We guide the users to finish each colorization task in roughly one minute, preventing the users from spending too much time on a single task. The evaluation results of the user study are shown in table 2 of our main paper.

10.2. Reflecting Unusual Color Hints

We also conduct a user study to evaluate how faithfully our model and the baselines reflect the user interaction, even though the hints can contain unusual colors. To be specific, we randomly select 500 images for each test dataset and prepare images generated by each model with strongly perturbed color hints as shown in Fig. 5 (a). The perturbed color is created by adding random values between -64 and 64 to each RGB value of the groundtruth color.

For a fair comparison, we unify the locations of given hints to each model by randomly sampling them within the guided regions produced by segmentation network. Simply, the number of provided hints and their positions are similar across the baselines and ours. The hint images have up to seven hints, and each model generates images based on the same number of hints. With the generated image and the hint map, the user is asked to rank the generated images in the order of how much the hints are properly reflected. Table 3 shows the percentage of generated images within the top-k of the rank over all datasets. Our model does not only get the highest top-1 ratio over all datasets, but also successfully reflect the diverse color conditions, as shown in Fig. 5. This implies that our model can work robustly in terms of color variations.

11. Qualitative Results

This section provides additional qualitative results with the size of 256×256 over three different datasets. Fig. 7 compares qualitative results for both automatic and conditional colorization models. Fig. 8, 9 and 10 show how the output images approach groundtruth when there are interactions between the model and an user on the CelebA[6], Tag2pix[4], and Yumi's Cells[8], respectively. Fig. 11 represents diverse output images according to the colors of given hints on each dataset.

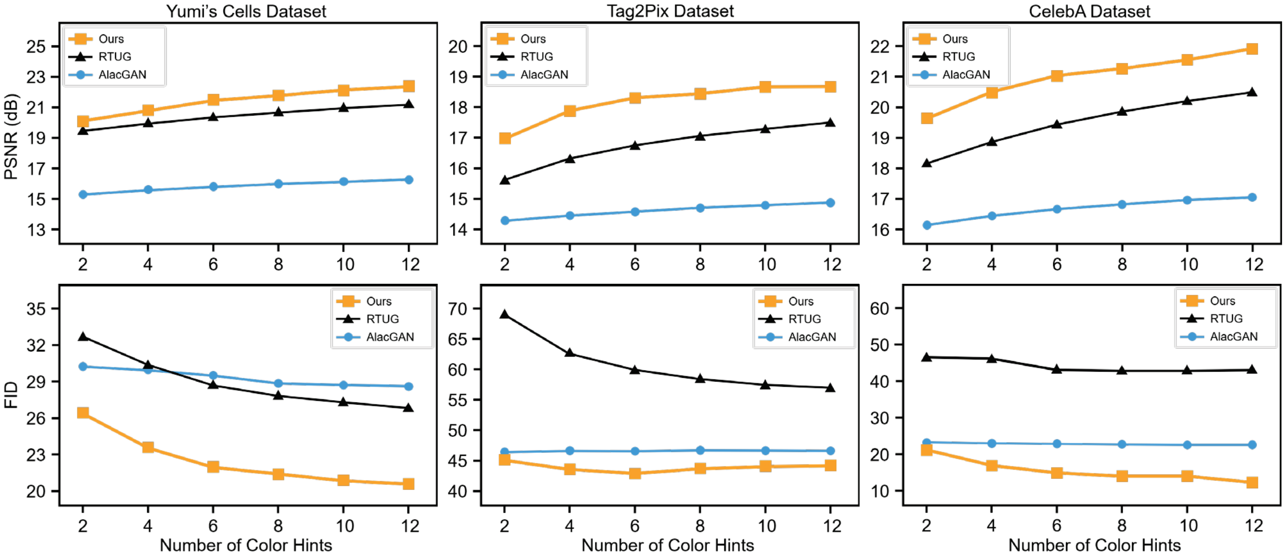

Figure 6: **Performance changes according to the number of user hints.** Columns mean the results of Yumi's Cells [8], Tag2pix [4] and CelebA [6] over baselines and GuidingPainter. The first row shows the PSNR scores and second row presents its corresponding FID [2] scores.

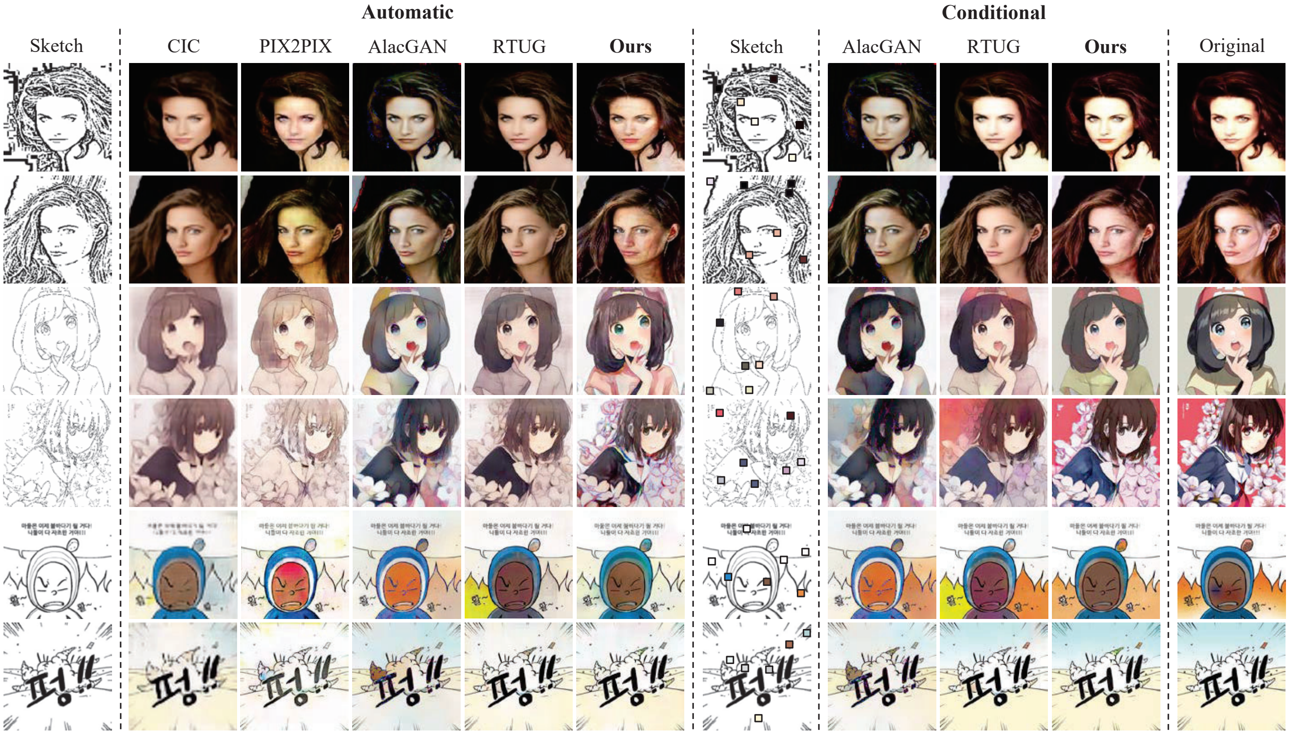

Figure 7: **Comparison to baselines on diverse datasets.** We compare our model in two approaches: automatic and conditional colorization ones. To assess the performance of our model in an automatic setting, we choose CIC, Pix2Pix, AlacGAN, and RTUG as baselines. For the condtional case, we equalize the number of hints given to all baselines and our model. Our model successfully colorizes each segment without color bleeding artifact, e.g., the third and fourth rows, and generates the continuous colors for each segment, e.g., hair in the first two rows, the sky and the ground in the fifth row.

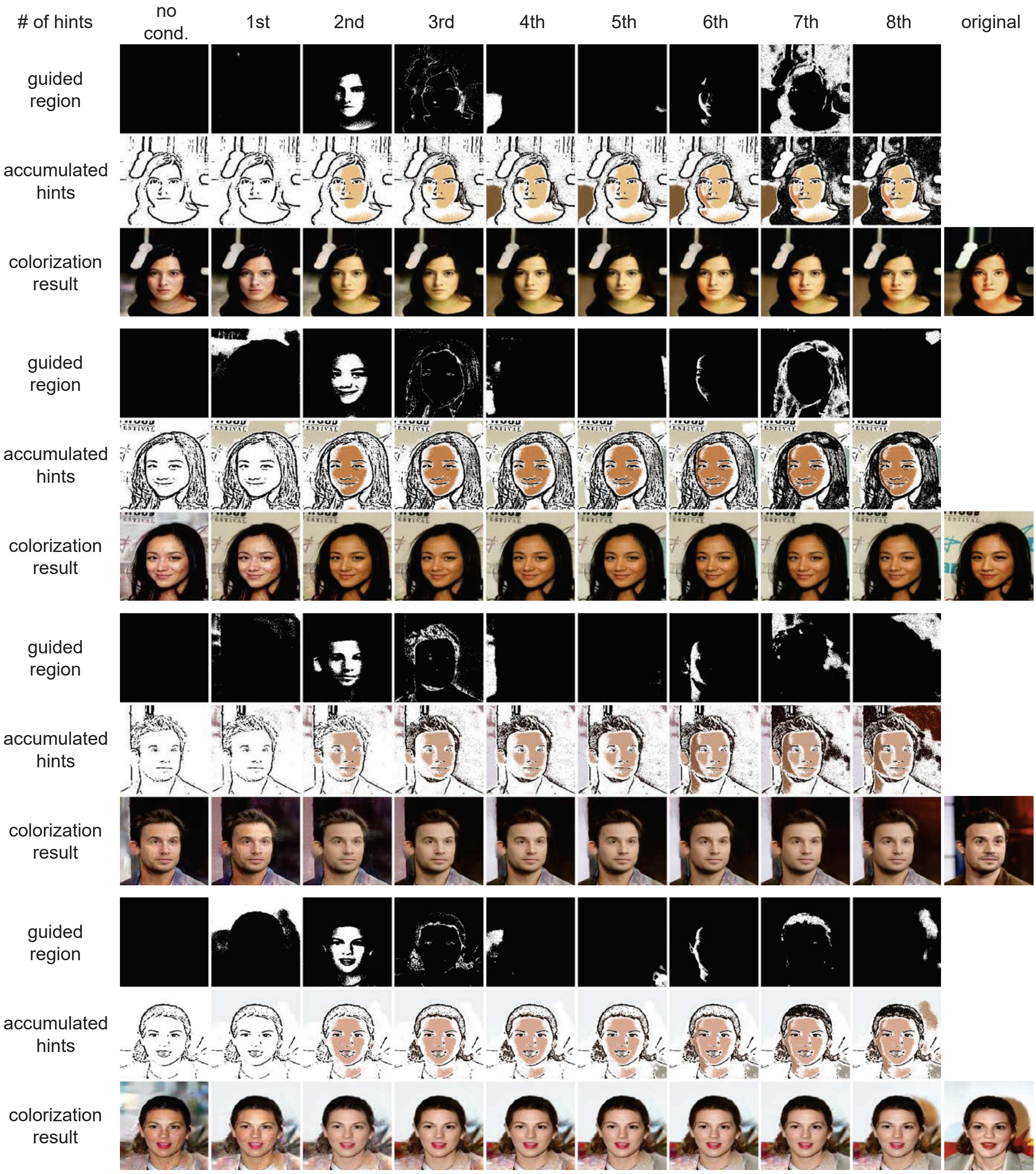

Figure 8: **Qualitative results on CelebA dataset.** For each interaction, which is noted as number of hints, our model estimates a hint region which the model wants to know first. Then, we select a representative color for the region and the color is spread to the region as visualized in accumulated hints. Finally, a colorization result is generated by our colorization network taking the guided regions accumulated hints and the sketch image. In these examples, we do not remove the noise used in the Gumbel Softmax operation to directly represent the guided regions provided to the colorization network. We summarize the interaction process in three rows for each four image. The results show how the input images change along with the color hints at each interaction step. In particular, the *6-th* column shows that the model captures the shadow of human face and colorizes it appropriate to the image.

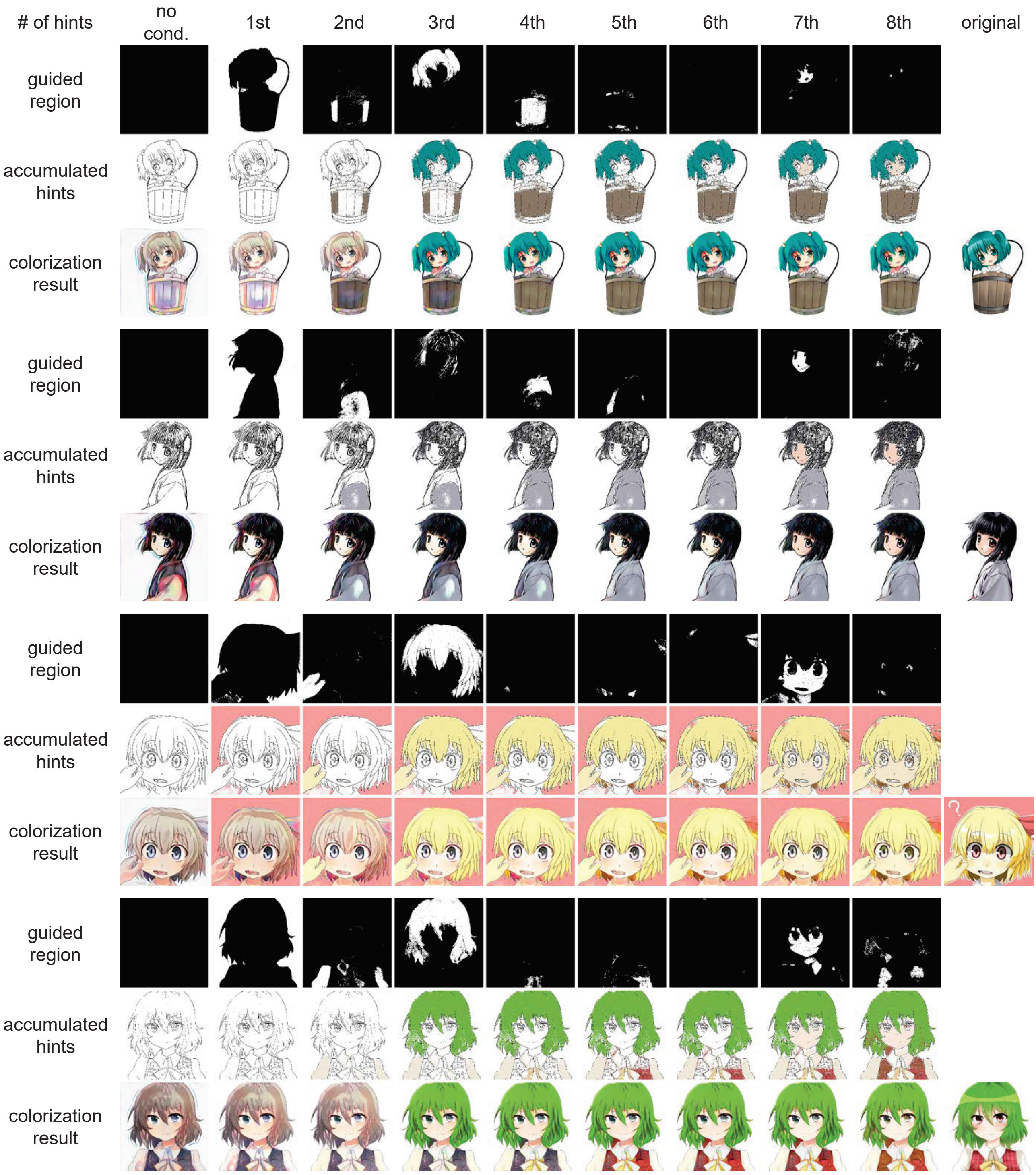

Figure 9: **Qualitative results on Tag2pix dataset.** Each interaction reveals that our model recognizes semantically related segments for each image, e.g., background, clothes, hair, and face of a character. In the *1-st* iteration, the model concentrates on the background and adapts a color if the background color is inputted. Especially, for the hair segment in the *3-rd* iteration, our model successfully reflects the color changes, not bleeding the color outside of the hair region.

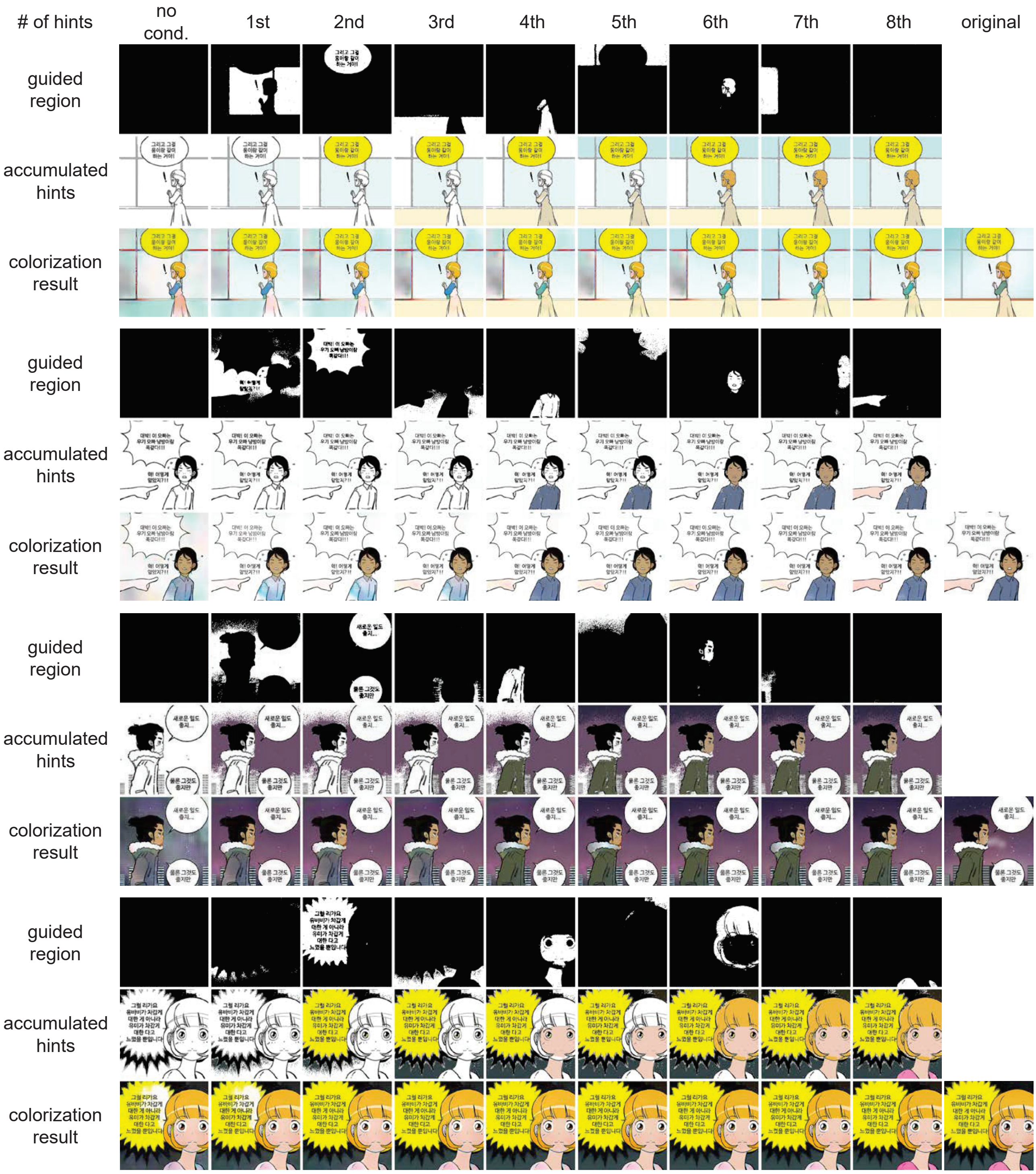

Figure 10: **Qualitative results on Yumi's Cells dataset.** Each intermediate iteration shows that semantically meaningful segments are recommended and colorized, such as, parts of background, speech balloon, clothes, face, and hair for an image. As shown in the rows of *colorization result*, the automatically colorized images become similar to the groundtruth image by adding each color hint in only eight iteration. This demonstrates that our model not only reflects the color condition in adequate location, but also improve the quality of result images. Especially on the last two images, our model fixes the key color errors, such as the purple night sky, the green clothes, and the yellow speech bubble.

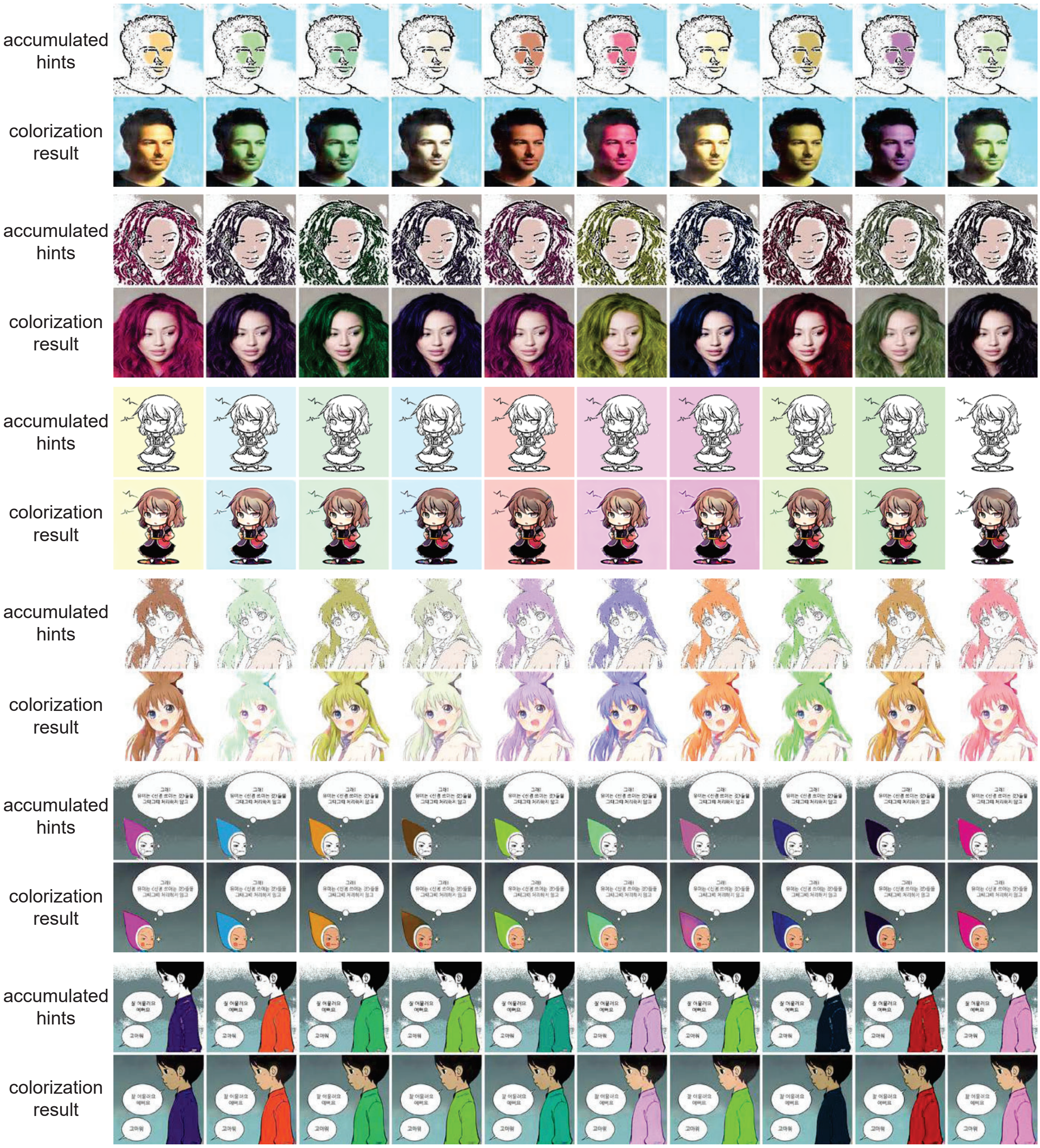

Figure 11: **Qualitative results of varying color hints on diverse datasets.** We sample two representative items from each three different dataset of CelebA, Tag2pix, and Yumi's cells. To confirm how well our model can reflect user-interaction, we only vary the color of the hint after fixing the input sketch image and its guided regions. The results show our model covers a wide range of color palette, including the green and purple faces.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 1

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In NIPS, 2017. 5

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017. 2

[4] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In ICCV, 2019. 4, 5

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, ICLR, 2015. 2

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In ICCV, 2015. 1, 4, 5

[7] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In ICCV, 2017. 2

[8] NaverWebtoon. Yumi's cells. https://comic.naver.com/webtoon/list.nhn?titleId=651673, 2019. [Online; accessed 22-11-2019]. 4, 5

[9] Weili Nie, Nina Narodytska, and Ankit Patel. Relgan: Relational generative adversarial networks for text generation. In ICLR, 2019. 2

[10] 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 (TOG), 36(4):1–11, 2017. 3

[11] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, 2017. 1