COLORIZATION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

We present the Colorization Transformer, a novel approach for diverse high fidelity image colorization based on self-attention. Given a grayscale image, the colorization proceeds in three steps. We first use an autoregressive transformer to produce a low resolution coarse coloring of the grayscale image. Our architecture adopts conditional self-attention blocks to effectively capture grayscale input. Two subsequent fully parallel networks upsample the coarse colored low resolution image into a finely colored high resolution image. Sampling from the Colorization Transformer produces diverse colorings whose fidelity outperforms the previous state-of-the-art on colorising ImageNet based on FID results and based on a human evaluation in a Mechanical Turk test. Remarkably, in more than 60% of cases human evaluators prefer the highest rated among three generated colorings over the ground truth.

1 INTRODUCTION

Figure 1: Samples of our model showing diverse, high-fidelity colorizations.

Image colorization is a challenging, inherently stochastic task that requires a semantic understanding of the scene as well as knowledge of the world. Core immediate applications of the technique include producing organic new colorizations of existing image and video content as well as giving life to originally grayscale media, such as old archival images (Tsaftaris et al., 2014), videos (Geshwind, 1986) and black-and-white cartoons (Sỳkora et al., 2004; Qu et al., 2006; Cinarel & Zhang, 2017). Colorization also has important technical uses as a way to learn meaningful representations without explicit supervision (Zhang et al., 2016; Larsson et al., 2016; Vondrick et al., 2018) or as an unsupervised data augmentation technique, whereby diverse semantics-preserving colorizations of labelled images are produced with a colorization model trained on a potentially much larger set of unlabelled images.

The current state-of-the-art in automated colorization are neural generative approaches based on log-likelihood estimation (Guadarrama et al., 2017; Royer et al., 2017; Ardizzone et al., 2019). Probabilistic models are a natural fit for the one-to-many task of image colorization and obtain better results than earlier determinisitic approaches avoiding some of the persistent pitfalls (Zhang et al., 2016). Probabilistic models also have the central advantage of producing multiple diverse colorings that are sampled from the learnt distribution.

The Colorization Transformer (ColTran) is a colorizing probabilistic model that is composed of an autoregressive generative core (ColTran core) and two fully parallel upsampler networks (ColTran upsamplers). All three networks make use exclusively of suitably modified axial self-attention blocks (Ho et al., 2019b), wherein a self-attention layer over rows is followed by a self-attention layer

over columns. To improve the grayscale conditioning, ColTran introduces a novel conditioning scheme, whereby each self-attention block is additionally conditioned by the grayscale input by way of multiple components.

The generation process first uses the core autoregressive transformer to produce in raster-scan order a coarsely colored image of size 64×64 with only 3 bits per color channel, resulting in a total of 512 colors. The second stage of the process enriches the color depth from 3 bits per color channel to the desired 8 bits per channel. This is achieved with a fully parallel network also based on axial self-attention blocks. The final stage, uses a similar network which then super-resolves the image into the desired resolution of 256×256.

We evaluate ColTran on colorizing images from ImageNet and show that ColTran achieves an image fidelity FID score (Heusel et al., 2017) of 19.26 as opposed to the previous state-of-the-art of 24.32 (Guadarrama et al., 2017). We perform ablations with respect to multiple architectural components including the conditioning scheme and show the benefits of the respective sub-components. We also show that ColTran outperforms previous approaches in a 2-Alternative Forced Choice (2AFC) assessment using human evaluators designed with Mechanical Turk and achieves a fooling rate of 42.3% compared to the previous best of 35.4% (Guadarrama et al., 2017). The fooling rate increases to 62% in an oracle setting, when the highest rated among three generated ColTran colorings is compared with the ground truth, suggesting that in this setting human raters find on average ColTran colorings more natural than the ground truth images themselves. This also outperforms the previous best oracle fooling rate of 38.3% (Guadarrama et al., 2017). Finally, we also find qualitatively that our model can uncover meaningful object structure in the images by visualizing the uncertainty of the predictions of ColTran's autoregressive core.

2 Related work

Colorization methods have initially relied on human-in-the-loop approaches to provide hints in the form of scribbles (Levin et al., 2004; Ironi et al., 2005; Huang et al., 2005; Yatziv & Sapiro, 2006; Qu et al., 2006; Luan et al., 2007; Tsaftaris et al., 2014; Zhang et al., 2017; Ci et al., 2018) and exemplar-based techniques that involve identifying a reference source image to copy colors from (Reinhard et al., 2001; Welsh et al., 2002; Tai et al., 2005; Ironi et al., 2005; Pitié et al., 2007; Morimoto et al., 2009; Gupta et al., 2012; Xiao et al., 2020). Exemplar based techniques have been recently extended to video as well (Zhang et al., 2019a). In the past few years, the focus has moved on to more automated, neural colorization methods. The deterministic family of approaches (Cheng et al., 2015; Dahl, 2016; Iizuka et al., 2016; Larsson et al., 2016; Zhang et al., 2016; Isola et al., 2017) involve variations of CNNs to model per-pixel color information conditioned on the intensity.

Probabilistic approaches typically extend unconditional image generation models such as VAE (Kingma & Welling, 2013), GANs (Goodfellow et al., 2014) and normalizing flows (Dinh et al., 2014) to incorporate conditioning information from a grayscale image (Deshpande et al., 2017; 2015; Cao et al., 2017; Ardizzone et al., 2019; Messaoud et al., 2018). Most closely related to ColTran are other autoregressive approaches such as PixColor (Guadarrama et al., 2017) and PIC (Royer et al., 2017) with PixColor obtaining slightly better results than PIC due to its CNN-based upsampling strategy. ColTran differs from PixColor in the following ways. We train ColTran in a completely unsupervised fashion, while the conditioning network in PixColor requires pre-training with an object detection network that provides substantial semantic information. PixColor relies on PixelCNN (Oord et al., 2016) that requires a large depth to model interactions between all pixels. ColTran relies on the AxialTransformer (Ho et al., 2019b) and can model all interactions between pixels with just 2 layers. Finally, PixColor uses different architectures for conditioning, colorization and super-resolution, while ColTran is conceptually simpler as we use self-attention blocks everywhere for both colorization and superresolution. The multi-stage generation process in ColTran that upsamples in depth and in size is related to that used in Subscale Pixel Networks (Menick & Kalchbrenner, 2018) for image generation, with differences in the order and representation of bits as well as in the use of fully parallel networks. The self-attention blocks that are the building blocks of ColTran were initially developed for machine translation (Vaswani et al., 2017), but are now widely used in a number of other applications including density estimation (Parmar et al., 2018; Child et al., 2019; Ho et al., 2019a; Weissenborn et al., 2019) and GANs (Zhang et al., 2019b).

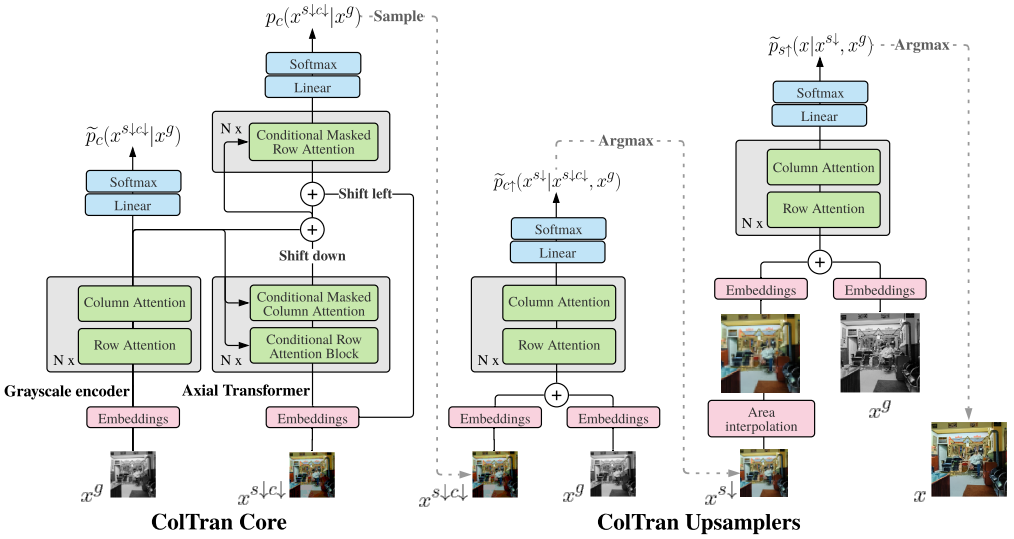


Figure 2: Depiction of ColTran. It consists of 3 individual models: an autoregressive colorizer (left), a color upsampler (middle) and a spatial upsampler (right). The autoregressive colorizer (ColTran core) is an instantiation of the Axial Transformer (Ho et al., 2019b) with conditional self-attention layers. All encoders and upsampler are stacked, row-/column self-attention blocks. Each attention block (in green) is residual and consists of the following operations: layer-norm $\rightarrow$ multihead self-attention $\rightarrow$ MLP.

3 MODEL

Image colorization is the task of transforming a grayscale image $x^g \in \mathbb{R}^{H \times W \times 1}$ into a colored image $x \in \mathbb{R}^{H \times W \times 3}$. The task is inherently stochastic; for a given grayscale image x^g, there exists a conditional distribution over x, $p(x|x^g)$. Instead of predicting x directly from x^g, we instead sequentially predict two intermediate low resolution images $x^{s\downarrow}$ and $x^{s\downarrow c\downarrow}$ with different color depth first. Besides simplifying the individual tasks, the smaller resolution allows for training larger models.

An illustration of the overall architecture is shown in Figure 2. It depicts all individual components described in the following.

Intermediate representations. We obtain $x^{s\downarrow}$, a spatially downsampled representation of x, by standard area interpolation. $x^{s\downarrow c\downarrow}$ is a 3 bit per-channel representation of $x^{s\downarrow}$, that is, each color channel has only 8 intensities. Thus, there are $8^3 = 512$ coarse colors per pixel which are predicted directly as a single "color" channel. We rewrite the conditional likelihood $p(x|x^g)$ to incorporate the intermediate representations as follows:

$$p(x|x^g) = p(x|x^g) \cdot 1 = p(x|x^g) \cdot p(x^{s\downarrow c\downarrow}, x^{s\downarrow}|x, x^g) = p(x^{s\downarrow c\downarrow}, x^{s\downarrow}, x|x^g) \tag{1}$$

$$= p(x|x^{s\downarrow}, x^g) \cdot p(x^{s\downarrow}|x^{s\downarrow c\downarrow}, x^g) \cdot p(x^{s\downarrow c\downarrow}|x^g) \tag{2}$$

ColTran Core. $\mathbf{f}_c$ is a conditional, auto-regressive axial transformer. We use $\mathbf{f}_c$ to model a distribution $p_c(x^{s\downarrow c\downarrow})$ over 512 coarse colors for every pixel, conditioned on colors from previously predicted pixels as per raster order (Eq. 3). In addition, we also train a parallel prediction head $\widetilde{p}_c(x^{s\downarrow c\downarrow})$ which we found beneficial for regularization (Eq. 4):

$$p_c(x^{s\downarrow c\downarrow}|x^g) = \prod_{i=1}^{M} \prod_{j=1}^{N} p_c(x_{ij}^{s\downarrow c\downarrow}|x^g, x_{<i}^{s\downarrow c\downarrow}, x_{i,<j}^{s\downarrow c\downarrow}) \tag{3}$$

$$\widetilde{p}_c(x^{s\downarrow c\downarrow}|x^g) = \prod_{i=1}^{M} \prod_{j=1}^{N} \widetilde{p}_c(x_{ij}^{s\downarrow c\downarrow}|x^g) \tag{4}$$

ColTran Upsamplers. We use alternating blocks of axial self-attention for upsampling color ($\mathbf{f}_{c\uparrow}$) and resolution ($\mathbf{f}_{s\uparrow}$), respectively, for all pixels in parallel. We apply $\mathbf{f}_{c\uparrow}$ to upsample the coarse image

$x^{s\downarrow c\downarrow}$ conditioned on the grayscale image x^g into a per-pixel, per-channel distribution over 256 color intensities (Eq. 5). Finally, $\mathbf{f}_{s\uparrow}$ spatially upsamples $x^{s\downarrow}$ to x (Eq. 6).

$$\widetilde{p}_{c\uparrow}(x^{s\downarrow}|x^g) = \prod_{i=1}^{M}\prod_{j=1}^{N} \widetilde{p}_{c\uparrow}(x_{ij}^{s\downarrow c\downarrow}|x^g, x^{s\downarrow c\downarrow}) \tag{5}$$

$$\widetilde{p}_{s\uparrow}(x|x^g) = \prod_{i=1}^{M}\prod_{j=1}^{N} \widetilde{p}_{s\uparrow}(x_{ij}|x^g, x^{s\downarrow}) \tag{6}$$

In our experiments, similar to (Guadarrama et al., 2017), we found parallel upsampling to be sufficient for high quality colorizations. Instead of sampling, however, we predict the most likely color for every pixel to avoid minor color inconsistencies between pixels that could stem from parallel sampling. Even though this slightly limits the potential diversity of colorizations, in practice we observe that sampling only coarse colors is still enough to produce a great variety of colorizations. Parallel upsampling has the huge advantage of fast generation which would be notoriously slow for full autoregressive models on high resolution.

Objective. We train our architecture to minimize the negative log-likelihood (Eq. 7) of the data. Note that $p_c/\widetilde{p}_c, \widetilde{p}_{s\uparrow}, \widetilde{p}_{c\uparrow}$ are maximized independently and therefore refer to different models that

$$\mathcal{L} = (1-\lambda)\log p_c + \lambda \log \widetilde{p}_c + \log \widetilde{p}_{c\uparrow} + \log \widetilde{p}_{s\uparrow} \tag{7}$$

We would like to remark that the success of colorization crucially depends on the quality of the autoregressive colorizer which models almost all of the stochasticity involved in the colorization process. For simplicity, we will refer to all image resolutions simply by $H \times W$ in the following.

4 Architecture

4.1 Row and column self-attention

Self-attention (SA) has become a standard building block in many neural architectures. Although the complexity of self-attention is quadratic with the number of input elements (here pixels), it has become quite popular for image modeling recently (Parmar et al., 2018; Weissenborn et al., 2019) due to modeling innovations that don't require running global self-attention between all pixels. Following the work of (Ho et al., 2019b) we employ standard $\mathbf{qkv}$ self-attention (Vaswani et al., 2017) within rows and columns of an image. By alternating row- and column self-attention we effectively allow global exchange of information between all pixel positions. For the sake of brevity we omit the exact equations for multihead self-attention and refer the interested reader to the Appendix A for more details. Row/column attention layers are the core components of our model. We use them in the autoregressive colorizer, the spatial upsampler and the color upsampler.

4.2 Axial Transformer

We adapt the Axial Transfomer (Ho et al., 2019b) model for colorization. The Axial Transformer is an autoregressive model that applies (masked) row- and column self-attention operations in a way that efficiently summarizes all past information $\mathbf{x}_{i,<j}$ and $\mathbf{x}_{<i,\cdot}$ to predict the next pixel at position i, j. The inputs to the model are the embedded pixels plus their respective position embeddings. The outer decoder computes a state over all previous rows $\mathbf{x}_{<i,\cdot}$ by applying multiple layers of full row self-attention followed by masked column self-attention. Causal masking is employed by setting all $A_{m,n} = 0$ where $n > m$ during self-attention (see Eq. 9). The resulting output of the outer decoder is shifted by a single row, such that the representation at position i, j only contains information about pixels $\mathbf{x}_{<i,\cdot}$ from prior rows. Finally, the inner decoder combines the output of the outer decoder and the color embedding from the previous pixel $\mathbf{x}_{i,j-1}$, and runs multiple layers of masked row-wise self-attention in order to infuse information from prior pixels of the same row $\mathbf{x}_{i,<j}$. The final state at position (i, j) (potentially) comprises information about all past pixels and is used to make a prediction for the color at position (i, j). Figure 2 illustrates the main components of the architecture. We would like to refer the interested reader to Ho et al. (2019b) for more detailed information.

Component	Unconditional	Conditional
Self-Attention	$\mathbf{y} = \text{Softmax}(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{D}})\mathbf{v}$	$\mathbf{y} = \text{Softmax}(\frac{\mathbf{q}_c\mathbf{k}_c^\top}{\sqrt{D}})\mathbf{v}_c$ where $\forall \mathbf{z} = \mathbf{k}, \mathbf{q}, \mathbf{v}$ $\mathbf{z}_c = (\mathbf{c}U_s^z) \odot \mathbf{z} + (\mathbf{c}U_b^z)$
MLP	$\mathbf{y} = \text{ReLU}(\mathbf{x}U_1 + \mathbf{b}_1)U_2 + \mathbf{b}_2$	$\mathbf{h} = \text{ReLU}(\mathbf{x}U_1 + \mathbf{b}_1)U_2 + \mathbf{b}_2$ $\mathbf{y} = (\mathbf{c}U_s^f) \odot \mathbf{h} + (\mathbf{c}U_b^f)$
Layer Norm	$\mathbf{y} = \beta\,\text{Norm}(\mathbf{x}) + \gamma$	$\mathbf{y} = \beta_c\,\text{Norm}(\mathbf{x}) + \gamma_c$ where $\forall \mu = \beta_c, \gamma_c$ $\mathbf{c} \in \mathbb{R}^{H\times W\times D} \rightarrow \hat{\mathbf{c}} \in \mathbb{R}^{HW\times D}$ $\mu = (\mathbf{u}\cdot\hat{\mathbf{c}})U_d^\mu \quad \mathbf{u} \in \mathbb{R}^{HW}$

Table 1: We contrast the different components of unconditional self-attention with self-attention conditioned on context $\mathbf{c} \in \mathbb{R}^{H\times W\times D}$. Learnable parameters specific to conditioning are denoted by $\mathbf{u}$ and $U_\cdot \in \mathbb{R}^{D\times D}$.

4.3 Conditional Axial Transformer

For coarse colorization of the low resolution images (Eq. 3) we have to condition the Axial Transformer on the low resolution grayscale image $x^{g\downarrow}$. In its original form, the Axial Transformer only supports conditioning by adding context states directly to the input embeddings which we found to be insufficient for good performance. Therefore, given a context representation $\mathbf{c} \in \mathbb{R}^{H\times W\times D}$ we propose the following conditional versions of all layers within the standard attention block (see Appendix A, Eqs. 8-12). An overview of the proposed changes is also provided in Table 1.

Conditional Self-Attention. For every layer in the decoder, we apply six 1×1 convolutions to $\mathbf{c}$ to obtain three scale and shift vectors which we apply element-wise to $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$ of the self-attention operation (Appendix 4.1), respectively.

Conditional MLP. A standard component of the transformer architecture is a two layer pointwise feed-forward network after the self-attention layer. We scale and shift to the output of each MLP conditioned on $\mathbf{c}$ as for self-attention.

Conditional Layer Norm. Layer normalization (Ba et al., 2016) globally scales and shifts a given normalized input using learnable vectors β, γ. Instead, we predict β_c and γ_c as a function of $\mathbf{c}$. We first aggregate $\mathbf{c}$ into a global 1-D representation $\overline{\mathbf{c}} \in \mathbb{R}^L$ via a learnable, spatial pooling layer. Spatial pooling is initialized as a mean pooling layer. Similar to conditional normalization layers (Perez et al., 2017; De Vries et al., 2017; Dumoulin et al., 2016; Huang & Belongie, 2017), we then apply a linear projection on $\overline{\mathbf{c}}$ to predict β_c and γ_c, respectively.

The initial conditioning context $\mathbf{c}^g$ is the output of a grayscale encoder consisting of multiple, alternating row and column self-attention layers. It serves as both context for the conditional layers and additional input to the embeddings of the outer decoder. The sum of the outer decoder's output and $\mathbf{c}^g$ serve as conditioning for the inner decoder. Figure 2 illustrates how conditioning is applied in the autoregressive core of the ColTran architecture.

4.4 Auxiliary parallel model

We additionally found it beneficial to train an auxiliary parallel prediction model directly on top of representations learned by the grayscale encoder. Intuitively, this forces the model to compute richer representations already at the output of the encoder which can help conditioning and therefore has a beneficial, regularizing effect on learning. We apply a linear projection, $U_{\text{parallel}} \in \mathbb{R}^{L\times 512}$ on top of $\mathbf{c}^g$ (the output of the grayscale encoder) into a per-pixel distribution over 512 coarse colors. It was crucial to tune the relative contribution of the autoregressive and parallel predictions, that is λ in the optimization objective to improve performance.

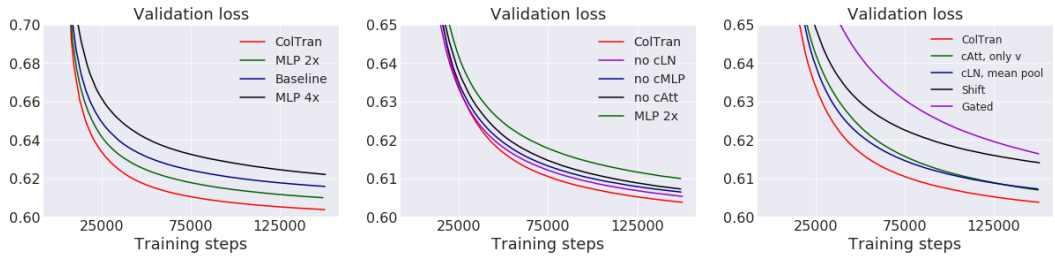


Figure 3: Per pixel log-likelihood of coarse colored 64×64 images over the validation set as a function of training steps. We ablate the various components of the ColTran core in each plot. See Section 5.2 for more details. **Left:** ColTran vs. Axial Transformer baselines. **Center:** Changing conditional to unconditional layers. **Right:** Conditioning details.

4.5 Color & Spatial Upsampling

In order to produce high-fidelity colorized images from low resolution, coarse color images and a given high resolution grayscale image, we train color- and spatial upsampling models. They share the same architecture while differing in their respective inputs and resolution at which they operate. Similar to the grayscale encoder, the upsamplers comprise multiple layers of alternating row-/column self-attention layers. The output of the encoder is projected to compute the logits underlying the per pixel color probabilities of the respective upsampler. Figure 2 illustrates the architectures.

Color Upsampler. We convert the coarse image $x^{s\downarrow c\downarrow} \in \mathbb{R}^{H\downarrow \times W\downarrow \times 1}$ of 512 colors back into a 3 bit RGB image with 8 symbols per channel. The channels are embedded using separate embedding matrices to $\mathbf{x}_k^{s\downarrow c\downarrow} \in \mathbb{R}^{H\downarrow \times W\downarrow \times D}$, where $k \in \{R, G, B\}$ indicates the channel. We upsample each channel individually conditioning only on the respective channel's embedding. The channel embedding is summed with the respective grayscale and position embedding for each pixel and serve as input to the subsequent self-attention layers (encoder). The output of the encoder is further projected to per pixel-channel probability distributions $\widetilde{p}_{c\uparrow}(x_k^{s\downarrow}|x^{s\downarrow c\downarrow}, x^g) \in \mathbb{R}^{H\downarrow \times W\downarrow \times 256}$ over 256 color intensities for all $k \in \{R, G, B\}$.

Spatial Upsampler. We first naively upsample $x^{s\downarrow} \in \mathbb{R}^{H\downarrow \times W\downarrow \times 3}$ into a blurry, high-resolution RGB image using area interpolation. As above, we then embed each channel of the blurry RGB image and run a per-channel encoder exactly the same way as with the color upsampler. The output of the encoder is finally projected to per pixel-channel probability distributions $\widetilde{p}_{s\uparrow}(x_k|x^{s\downarrow}, x^g) \in \mathbb{R}^{H \times W \times 256}$ over 256 color intensities for all $k \in \{R, G, B\}$.

5 Experiments

5.1 Training and Evaluation

We evaluate ColTran on colorizing 256×256 grayscale images from the ImageNet dataset (Russakovsky et al., 2015). We train the ColTran core, color and spatial upsamplers independently on 16 TPUv2 chips with a batch-size of 224, 768 and 32 respectively. We use 4 axial attention blocks in each component of our architecture, with a hidden size of 512 and 4 heads. We use RMSprop (Tieleman & Hinton, 2012) with a fixed learning rate of 3e-4. We set apart 10000 images from the training set as a holdout set to tune hyperparameters and perform ablations. To compute FID, we generate 5000 samples conditioned on the grayscale images from this holdout set. We use the public validation set to display qualitative results and report final numbers.

5.2 Ablations of ColTran Core

The autoregressive core of ColTran models downsampled, coarse-colored images of resolution 64×64 with 3 bits per channel, conditioned on the respective grayscale image. In a series of experiments we ablate the different components of the architecture (Figure 3). We report the per-pixel log-likelihood over 512 coarse colors on the validation set as a function of training steps.

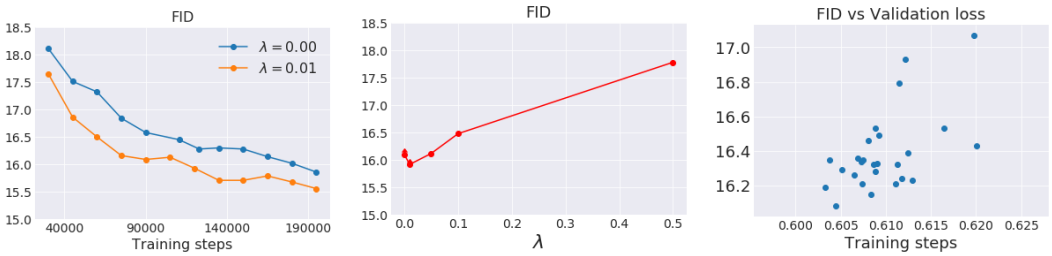


Figure 4: Left: FID of generated 64×64 coarse samples as a function of training steps for $\lambda = 0.01$ and $\lambda = 0.0$. **Center:** Final FID scores as a function of λ. **Right:** FID as a function of log-likelihood.

Axial Transformers with and without conditional layers. Figure 3 left compares the standard Axial Transformer with the conditional Axial Transformer in the ColTran core (Section 4.3). Our network has a slightly larger parameter count, as we apply additional 1×1 projections for conditioning. In order to make a fair comparison, we train baselines with $2\times$ and $4\times$ wider MLP dimensions, labelled as MLP $2\times$ and MLP $4\times$. We tune the learning rate for all our baselines.

Conditional layers. We remove each conditional component of the conditional self attention blocks in ColTran core, one at a time and retrain the new ablated model. Each ablated model on its own improves over the baseline. Further, each conditional component is almost equally important. Together they outperform all the three ablated models.

Conditioning details. Figure 3 right shows an ablation of all design decisions of our conditional layers. Either of the following three ablations leads to worse performance: using conditional shifts only (*Shift*); applying conditioning only on $\mathbf{v}$ for self-attention (*cAtt, only v*); or using fixed mean pooling instead of the learnable spatial pooling (*cLN, mean pool*). ColTran also performs better than an existing method that applies gating functions (Van den Oord et al., 2016) for conditioning (*Gated*).

5.3 Other Ablations

Auxiliary Parallel Model. We study the effect of the hyperparameter λ, which controls the contribution of the auxiliary parallel prediction model described in Section 4.4. For a given λ, we now optimize $\hat{p}_c(\lambda) = (1-\lambda)\log p_c(.) + \lambda \log \widetilde{p}_c(.)$ instead of just $\log p_c(.)$. Note that $\widetilde{p}_c(.)$, models each pixel independently, which is more difficult than modelling each pixel conditioned on previous pixels given by $p_c(.)$. Hence, employing $\hat{p}_c(\lambda)$ as a holdout metric, would just lead to a trivial soluion at $\lambda = 0$. Instead, the FID of the generated coarse 64x64 samples provides a reliable way to find an optimal value of λ. In Figure 4, at $\lambda = 0.01$, our model converges to a better FID faster with a marginal but consistent final improvement. At higher values the performance deteriorates quickly.

Upsamplers. Upsampling coarse colored, low-resolution images to a higher resolution is much simpler. Given ground truth 64×64 coarse images, the ColTran upsamplers map these to fine grained 256×256 images without any visible artifacts and FID of 16.4. For comparison, the FID between two random sets of 5000 samples from our holdout set is 15.5. It is further extremely important to provide the grayscale image as input to each of the individual upsamplers, without which the generated images appear highly smoothed out and the FID drops to 27.0. We also trained a single upsampler for both color and resolution. The FID in this case, drops marginally to 16.6.

5.4 Frechet Inception Distance

We compute FID using colorizations of 5000 grayscale images of resolution 256×256 from the ImageNet validation set. PixColor samples even lower resolution 28×28 colored images autoregressively as compared to ColTran's 64×64. As a control experiment, we train a core model on resolution 28×28 (ColTran-S) to disentangle architectural choices and the inherent stochasticity of modelling higher resolution images. The ColTran-S obtains a FID of 21.9 that significantly improves over PixColor's FID of 24.32. Finally, ColTran achieves an even better score of 19.71. All results are presented in Table 2 left.

Models	FID
ColTran	**19.71 ± 0.12**
ColTran-S	21.9 ± 0.09
PixColor	24.32 ± 0.21
cGAN	24.41 ± 0.27
CNN	24.95 ± 0.27
cINN	25.13 ± 0.3
VAE-MDN	25.98 ± 0.28
Ground truth	14.68 ± 0.15
Grayscale	30.19 ± 0.1

Models	AMT Fooling rate
ColTran (Oracle)	62.0 % ± 0.99
ColTran (Seed 1)	40.5 % ± 0.81
ColTran (Seed 2)	**42.3 % ± 0.76**
ColTran (Seed 3)	41.7 % ± 0.83
PixColor (Oracle)	38.3 % ± 0.98
PixColor (Seed 1)	33.3 % ± 1.04
PixColor (Seed 2)	35.4 % ± 1.01
PixColor (Seed 3)	33.2 % ± 1.03
CIC	29.2 % ± 0.98
LRAC	30.9 % ± 1.02
LTBC	25.8 % ± 0.97

Table 2: We outperform various state-of-the-art colorization models both on FID (left) and human evaluation (right).

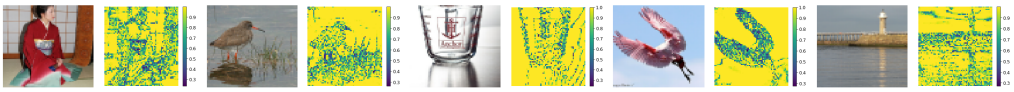


Figure 5: We display the per-pixel, maximum predicted probability over 512 colors as a proxy for uncertainty.

Correlation between FID and Log-likelihood. For each architectural variant, Figure 4 right illustrates the correlation between the log-likelihood and FID after 150K training steps. There is a moderately positive correlation of 0.57 between the log-likelihood and FID. Importantly, even an absolute improvement on the order of 0.01 - 0.02 can improve FID significantly. This suggests that designing architectures that achieve better log-likelihood values is likely to lead to improved FID scores and colorization fidelity.

5.5 Qualitative Evaluation

Human Evaluation. For our qualitative assessment, we follow the protocol used in PixColor. ColTran (L) colorizes 500 grayscale images, with 3 different colorizations per image, denoted as seeds. Human raters assess the quality of these colorizations with a two alternative-forced choice (2AFC) test. We display both the ground-truth and recolorized image sequentially for one second in random order. The raters are then asked to identify the image with fake colors. For each seed, we report the mean fooling rate over 500 colorizations and 5 different raters. For the oracle methods, we use the human rating to pick the best-of-three colorizations. ColTran's best seed achieves a fooling rate of 42.3 % compared to the 35.4 % of PixColor's best seed. ColTran Oracle achieves a fooling rate of 62 %, indicating that human raters prefer ColTran's best-of-three colorizations over the ground truth image itself.

Visualizing uncertainty. The autoregressive core model of ColTran should be highly uncertain at object boundaries when colors change. Figure 9 illustrates the per-pixel, maximum predicted probability over 512 colors as a proxy for uncertainty. We observe that the model is indeed highly uncertain at edges and within more complicated textures.

6 Conclusions

We presented the Colorization Transformer (ColTran), an architecture that entirely relies on self-attention for image colorization. We introduce a conditional self-attention, a novel building block for conditional, generative models based on self-attention. Our ablations show the superiority of employing this mechanism over a number of different baselines. ColTran achieves very strong empirical results, outperforming prior state-of-the-art by a large margin. Finally, we demonstrate that ColTran can generate diverse, high-fidelity colorizations on ImageNet, which are largely indistinguishable from the ground-truth even for human raters.

REFERENCES

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu. Unsupervised diverse colorization via generative adversarial networks, 2017.

Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415–423, 2015.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1536–1544, 2018.

Ceyda Cinarel and Byoung-Tak Zhang. Into the colorful world of webtoons: Through the lens of neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 3, pp. 35–40. IEEE, 2017.

Ryan Dahl. Automatic colorization, 2016.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pp. 6594–6604, 2017.

Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 567–575, 2015.

Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth. Learning diverse image colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6837–6845, 2017.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

David M Geshwind. Method for colorizing black and white footage, August 19 1986. US Patent 4,606,625.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.

Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM international conference on Multimedia*, pp. 369–378, 2012.

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 *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *arXiv preprint arXiv:1902.00275*, 2019a.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019b.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

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

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.

Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Rendering Techniques*, pp. 201–210. Citeseer, 2005.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pp. 577–593. Springer, 2016.

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

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*, pp. 309–320, 2007.

Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018.

Safa Messaoud, David Forsyth, and Alexander G. Schwing. Structural consistency and controllability for diverse colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Yuji Morimoto, Yuichi Taguchi, and Takeshi Naemura. Automatic colorization of grayscale images using multiple images on the web. In *SIGGRAPH 2009: Talks*, pp. 1–1. 2009.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007.

Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006.

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

Amelie Royer, Alexander Kolesnikov, and Christoph H Lampert. Probabilistic image colorization. *arXiv preprint arXiv:1705.04258*, 2017.

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.

Daniel Sỳkora, Jan Buriánek, and Jiří Žára. Unsupervised colorization of black-and-white cartoons. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pp. 121–127, 2004.

Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 747–754. IEEE, 2005.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Sotirios A Tsaftaris, Francesca Casadio, Jean-Louis Andral, and Aggelos K Katsaggelos. A novel visualization tool for art history and conservation: Automated colorization of black and white archival photographs of works of art. *Studies in conservation*, 59(3):125–135, 2014.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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)*, pp. 391–408, 2018.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019.

Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 277–280, 2002.

Chufeng Xiao, Chu Han, Zhuming Zhang, Jing Qin, Tien-Tsin Wong, Guoqiang Han, and Shengfeng He. Example-based colourization via dense encoding pyramids. In *Computer Graphics Forum*, volume 39, pp. 20–33. Wiley Online Library, 2020.

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

Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar-based video colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8052–8061, 2019a.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363. PMLR, 2019b.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

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. *arXiv preprint arXiv:1705.02999*, 2017.

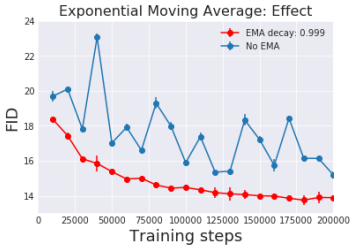


Figure 6: FID vs training steps, with and without polyak averaging.

Figure 7: Top: Colorizations **Bottom**: Ground truth. From left to right, our colorizations have a progressively higher fooling rate.

A Row/Column Self-attention

In the following we describe row self-attention, that is, we omit the height dimension as all operations are performed in parallel for each column. Given the representation of a single row within of an image $\mathbf{x}_{i,\cdot} \in \mathbb{R}^{W \times D}$, row-wise self-attention block is applied as follows:

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathrm{LN}(\mathbf{x}_{i,\cdot}) U_{qkv} \qquad U_{qkv} \in \mathbb{R}^{D \times 3D_h} \tag{8}$$

$$A = \mathrm{softmax}\left(\mathbf{q}\mathbf{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{W \times W} \tag{9}$$

$$\mathrm{SA}(\mathbf{x}_{i,\cdot}) = A\mathbf{v} \tag{10}$$

$$\mathrm{MSA}(\mathbf{x}_{i,\cdot}) = [\mathrm{SA}_1(\mathbf{x}_{i,\cdot}), \mathrm{SA}_2(\mathbf{x}_{i,\cdot}), \cdots, \mathrm{SA}_k(\mathbf{x}_{i,\cdot})]\, U_{out} \qquad U_{out} \in \mathbb{R}^{k \cdot D_h \times D} \tag{11}$$

LN refers to the application of layer normalization (Ba et al., 2016). Finally, we apply residual connections and a feed-forward neural network with a single hidden layer and ReLU activation (MLP) after each self-attention block as it is common practice in transformers.

$$\hat{\mathbf{x}}_{i,\cdot} = \mathrm{MLP}(\mathrm{LN}(\mathbf{x}'_{i,\cdot})) + \mathbf{x}'_{i,\cdot} \qquad \mathbf{x}'_{i,\cdot} = \mathrm{MSA}(\mathbf{x}_{i,\cdot}) + \mathbf{x}_{i,\cdot} \tag{12}$$

Column-wise self-attention over $\mathbf{x}_{\cdot,j} \in \mathbb{R}^{H \times D}$ works analogously.

B Exponential Moving Average

We found using an exponential moving average (EMA) of our checkpoints, extremely crucial to generate high quality samples. In Figure 6, we display the FID as a function of training steps, with and without EMA. On applying EMA, our FID score improves steadily over time.

C Colorizations at different fooling rates

D Analysis of MTurk ratings

We analyzed our samples on the basis of the MTurk ratings in Figure 8. To the left, we show images, where all the samples have a fool rate > 60 %. Our model is able to show diversity in color for both high-level structure and low-level details. In the center, we display samples that have a high variance in MTurk ratings, with a difference of 80 % between the best and the worst sample. All of these are complex objects, that our model is able to colorize reasonably well given multiple attempts. To the

Figure 8: In each column, we display the ground truth followed by 3 samples. **Left:** Diverse and real. **Center:** Realism improves from left to right. **Right:** Failure cases

right of Figure 8, we show failure cases where all samples have a fool rate of 0 %, For these cases, our model is unable to colorize highly complex structure, that would arguably be difficult even for a human.

E MORE PROBABILITY MAPS

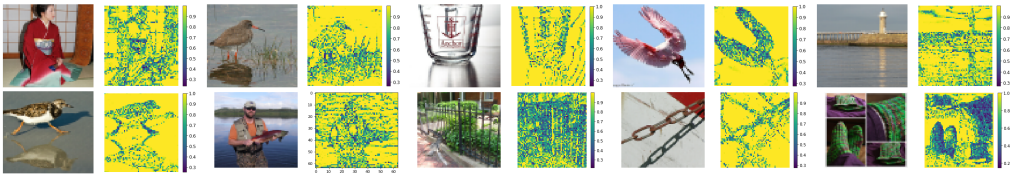
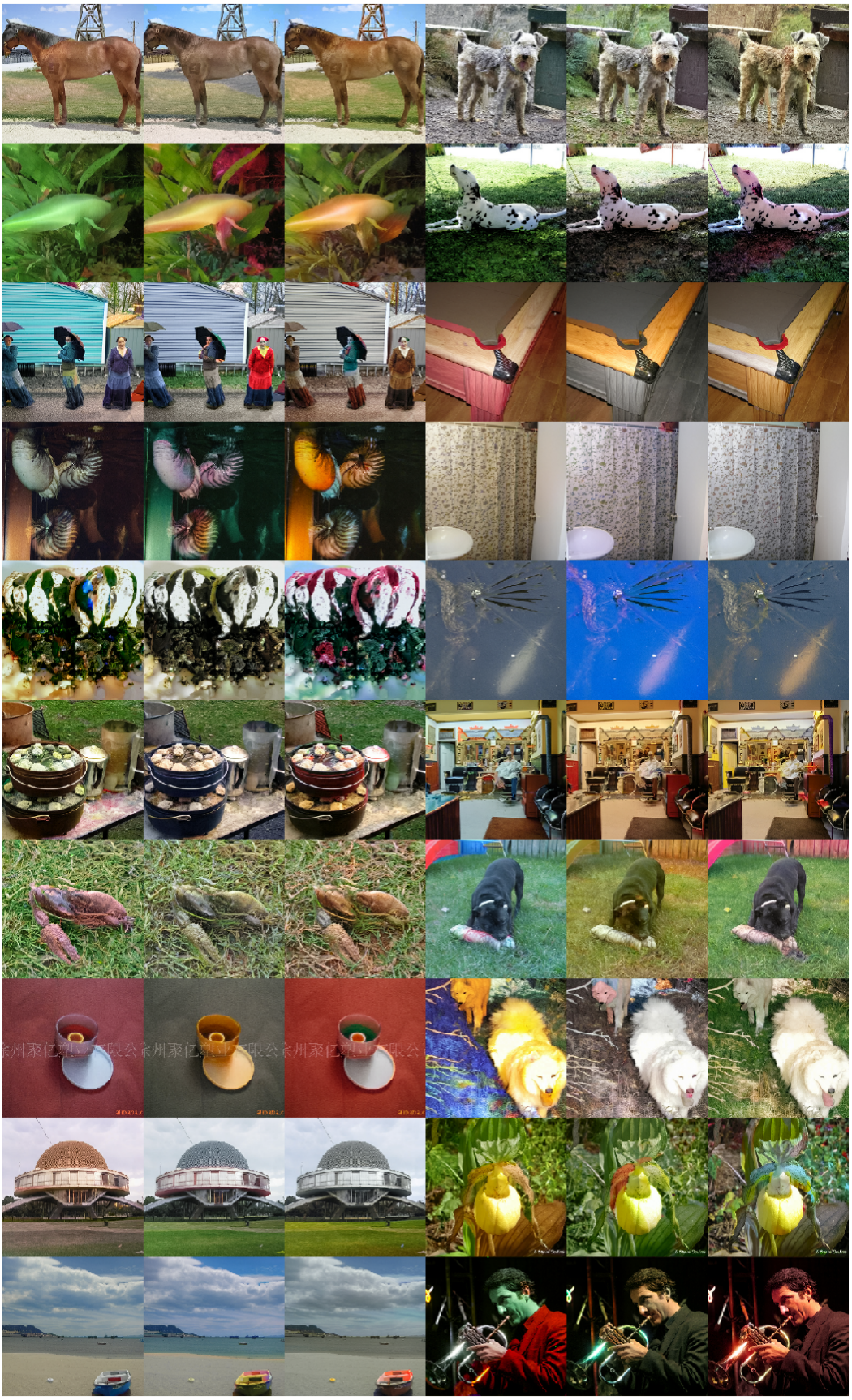
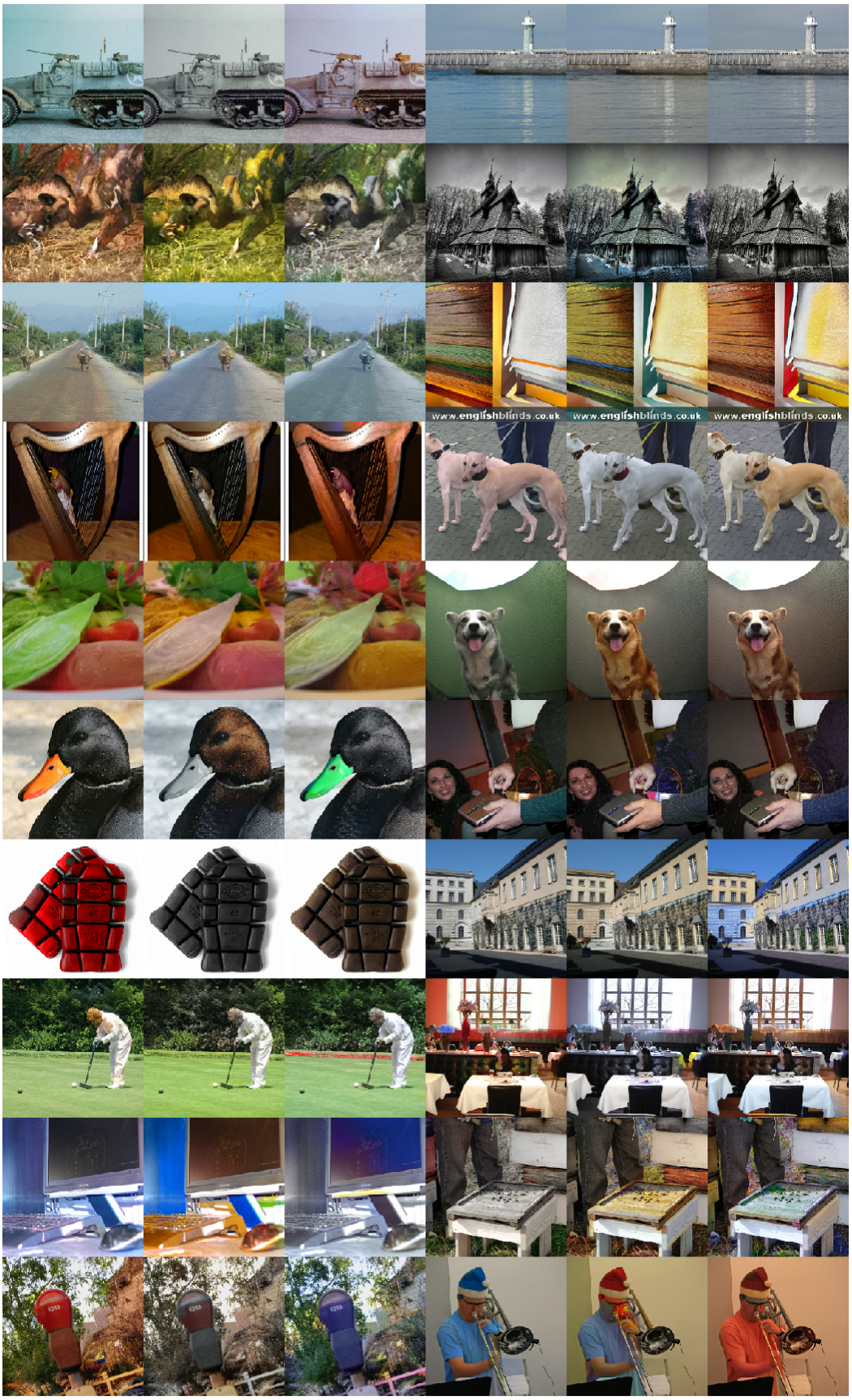


Figure 9: We display the per-pixel, maximum predicted probability over 512 colors as a proxy for uncertainty.

F MORE SAMPLES




www.englishblinds.co.uk
www.englishblinds.co.uk
www.englishblinds.co.uk