MarkovGen: Structured Prediction for Efficient Text-to-Image Generation

Sadeep Jayasumana Daniel Glasner Srikumar Ramalingam Andreas Veit
Ayan Chakrabarti Sanjiv Kumar
Google Research, New York
{sadeep, dglasner, rsrikumar, aveit, ayanchakrab, sanjivk}@google.com

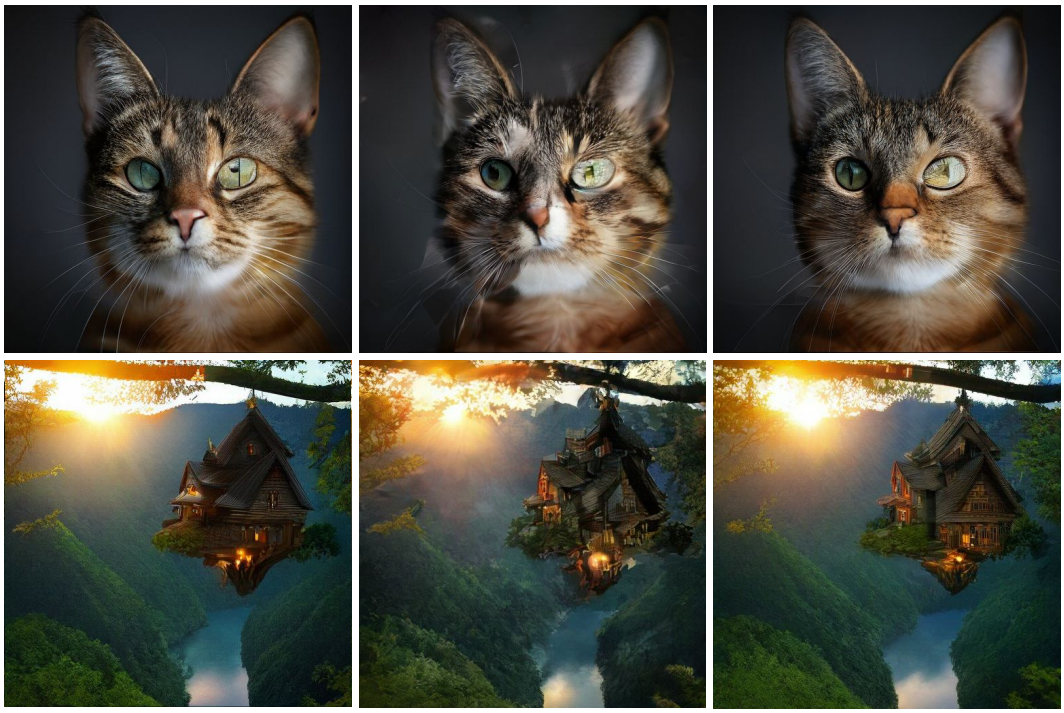

Figure 1. *MarkovGen improves the speed and quality of token-based image generation models such as Muse, by reducing the number of sampling steps and replacing them with a light-weight Markov Random Field (MRF) model.*

Abstract

Modern text-to-image generation models produce high-quality images that are both photorealistic and faithful to the text prompts. However, this quality comes at significant computational cost: nearly all of these models are iterative and require running sampling multiple times with large models. This iterative process is needed to ensure that different regions of the image are not only aligned with the text prompt, but also compatible with each other. In this work, we propose a light-weight approach to achieving this compatibility between different regions of an image, using a Markov Random Field (MRF) model. We demonstrate the effectiveness of this method on top of the latent token-based Muse text-to-image model. The MRF richly encodes the compatibility among image tokens at different spatial locations to improve quality and significantly reduce the required number of Muse sampling steps. Inference with the MRF is significantly cheaper, and its parameters can be quickly learned through back-propagation by modeling MRF inference as a differentiable neural-network layer. Our full model, MarkovGen, uses this proposed MRF model to both speed up Muse by $1.5\times$ *and produce higher quality images by decreasing undesirable image artifacts.*

1. Introduction

Recent image-to-text models [18, 21, 24, 25, 34] are remarkably successful at producing high-quality, photo-realistic images that are faithful to the provided text

prompts, and are poised to drive a new generation of tools for creativity and graphic design. However, the generation process with these models is iterative and computationally expensive, requiring multiple sampling steps through large models. For example, diffusion models [24, 25] require multiple denoising steps to generate the final image, the Parti model [34] auto-regressively generates image tokens one at a time. While the recently proposed Muse model [2] generates multiple tokens at a time, it still requires a large number of sampling steps to arrive at the final image.

This iterative process is needed to ensure that different regions or patches of the images are not only aligned with the provided text prompt, but also *compatible with each other*. Current text-to-image models achieve this spatial compatibility by repeatedly applying their full model multiple times on intermediate image predictions—a process that is computationally very expensive. In this paper, we demonstrate that a significantly lighter-weight approach can achieve the same compatibility.

To this end, we propose a new *structured prediction* approach that applies to image generation models operating in a discrete token space, such as the VQGAN token space [2,3,8]. These models generate images by first selecting tokens in a fixed-size token grid and later detokenizing them into an RGB image. Usual token-based image generation methods select tokens by *independently* sampling from the probability distributions at different patch locations. In contrast, we model the whole image *jointly* using a fully-connected Markov Random Field (MRF) that encodes compatibility between all pairs of tokens (image patches). The tokens at different patch locations are then determined based on this joint distribution. Consequently, as illustrated in Figure 2, a confident token at one location can influence the selected tokens at other locations to enhance the overall compatibility of the token arrangement, and therefore the fidelity of the final image. We use mean-field inference [13, 14, 36] to solve this MRF, which also permits training the compatibility parameters of the model through back-propagation. During image generation with a trained model, the MRF inference comes at a negligible cost compared to the cost of large Transformer models used to predict the initial token probabilities.

To showcase the benefits of our MRF model, we introduce a new text-to-image model, MarkovGen, that can work in conjunction with the Muse model [2]. Muse uses a parallel decoding approach where all tokens of an image are predicted in parallel at each step. Muse has been shown to be much faster (around $3\times$ faster than the closest competitor) than other state-of-the-art image generation models such as DALL-E, Imagen, Parti, and Stable Diffusion, while producing similar or better quality images [2]. Although Muse produces predictions for every patch simultaneously, single-shot parallel decoding leads to serious quality degradation in the generated images [2]. Muse solves this by embracing progressive parallel decoding, where a small incremental number of high confidence tokens are fixed after each iteration. We show that by learning the compatibility of the tokens and applying the MRF inference after limited number of sampling steps with the Muse model, we achieve significant quality and efficiency gains over the Muse's full iterative approach (see Figure 1).

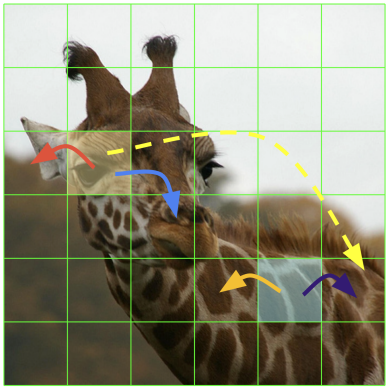

Figure 2. *Benefits of encouraging token compatibility with an MRF model. During MRF inference, a confident token, such as the token representing the giraffe's eye, encourages the neighboring tokens to be compatible to represent other parts of a giraffe face such as ears and nose. Similarly, tokens representing the texture of the giraffe body can influence nearby tokens to encourage consistent patterns. Our formulation also supports long-range connections, such as the one shown with the dashed yellow line.*

Reducing the latency and improving the quality of Muse, one of the fastest text-to-image models, will have important practical implications for real-world deployments. The success of our MRF formulation in modeling spatial and label relationships of image tokens opens up the future possibility of refining predictions of other token-based methods such as Parti [34] and discrete-diffusion models [11] with MRFs.

In summary, our contributions are:

- We propose an MRF model, a type of probabilistic graphical model, that can predict a globally compatible set of image tokens by explicitly modeling spatial and token label relationships. To the best of our knowledge, this is the first work to exploit MRFs to improve the efficiency and quality of text-to-image models.

- Our MarkovGen model, where we replace the last few steps of Muse with a learned MRF layer, leads to a $1.5\times$ speedup as well as improved quality results, as demonstrated by human evaluation and FID distances.

- We show that the MRF model parameters can be trained in just a few hours, allowing us to quickly combine the MRF model with pre-trained Muse models to reap efficiency and quality gains.

2. Related Work

Text-to-Image Generation: In recent years, papers such as [2, 3, 10, 15, 21, 24, 25, 27, 33–35] have proposed a diverse variety of methods to generate high-quality images given a text prompt as input. We discuss some of the most relevant approaches below.

Many text-to-image models [9, 18, 19, 21, 24, 25] use denoising diffusion probabilistic models (DDPM) [12] to generate images, where the model is invoked successively to "denoise" previous intermediate versions and progressively refine the image output. While in theory, we need infinitely small and many denoising steps, only a few hundreds of steps are used in practice [28]. Progressive distillation algorithms are being developed to cut down the number of steps [26]. Most of these models directly operate on and produce pixel intensities, [24, 35] are variants that operate on a lower-dimensional latent representation.

In contrast, the Parti [34], DALL-E [22], and Muse [2] models generate images in a space of discrete token representation. They use a VQGAN [8] model, derived from VQ-VAE [32], to represent non-overlapping image patches with tokens—with values from a discrete vocabulary—and cast the image generation task as that of generating image tokens. The Parti and DALL-E models approach token generation with auto-regressive modeling, generating tokens one at a time in sequence, where each token is generated conditioned on the text input and all previously generated tokens.

The Muse model [2], on the other hand, is trained to take the text prompt and any already generated image tokens as input, and make predictions for all remaining image tokens simultaneously. In particular, it is trained as a BERT-style [6] encoder model operating on a masked set of image tokens (with tokens not already generated being masked), with cross-attention to an encoding of the text prompt input. To generate an image, the model is invoked in multiple sampling steps, with all image tokens being masked in the first step. At each step, the Muse model makes predictions for all masked tokens. A subset of these predictions are selected and added to the set of fixed and non-masked tokens, which are then used as conditioning input for subsequent invocations till the all tokens have been fixed. Similar to Muse, Paella [23] and Cogview2 [7] also exploit progressive parallel decoding to achieve speedup. A similar approach to parallel decoding for text was introduced by [17].

Like many other text-to-image generation models, Muse first generates a low-resolution version of the target image, and then conditions on this low-resolution image to generate the high-resolution version. It uses a similar architecture and sampling approach for the high-resolution generation stage, except in this case, the low-resolution image tokens are provided as additional conditioning input.

For the selected tokens at each sampling stage of Muse, the token values are determined independently for each token from the predicted per-token distributions. Our structured prediction approach, in contrast, considers compatibility between the values of different tokens, and by doing so, is able to improve the quality and reduce the number of sampling steps required—in both the low- and high-resolution stages.

Many of the text-to-image algorithms are also being extended to develop algorithms to handle other conditional inputs [35], and text-to-video generation [10, 27, 33].

Structured Prediction: Markov and Conditional random fields (CRF)s have a long history of being used in computer vision for diverse applications such as stereo, segmentation, and image reconstruction [31]. These MRF and CRF models have typically been used to enforce smoothness constraints, i.e., that semantic labels, pixel intensities, stereo depths, etc. at nearby locations are similar. In neural network-based methods too, they have been a useful post-processing step [4] to yield smooth consistent results.

While early MRF and CRF models considered edges only among immediate pixel neighbors on the image plane, [14] introduced "fully-connected" CRF models that had far longer range connections, and showed that the energy for these models could effectively be minimized using mean-field inference. Using this fully-connected formulation, [36] proposed back-propagating through the mean-field inference steps to jointly train a CRF model with a CNN network to achieve better semantic image segmentation.

In this work, we use an MRF formulation to achieve consistency in predicted image tokens in the context of text-to-image generation, and like [36], we also use a fully connected MRF model and learn its parameters by back-propagation. However, in our case, the MRF is defined over tokenized patches, the label space corresponds to the vocabulary of a VQGAN [8] and the MRF enforces consistency between different token values rather than smoothness.

It is worth mentioning here that CRFs have recently also been proposed to improve text generation [29, 30]. Like our case, these methods also use a Transformer model to generate "unaries" that are then provided as input to a CRF model. However, these methods consider edges only between neighboring tokens, and since text sequences are one-dimensional, are able to use chain decoding techniques (like beam search) for inference. In contrast, our method reasons with a two-dimensional MRF model with edges between all pairs of patches in the image.

3. Structured Token Prediction

In this section, we introduce our MRF formulation for structured token prediction. In token-based image generation, a neural network (often a Transformer model) makes predictions to generate a fixed size (16×16, for example) *token image* containing token labels. This token image is then sent through a *detokenizer* to generate an RGB image [8].

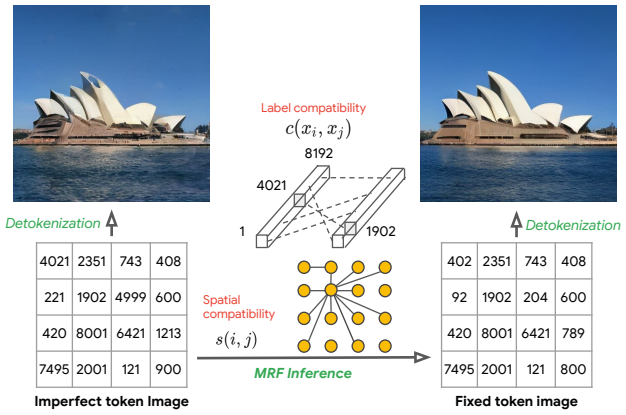

Figure 3. *Given individual token probabilities from an underlying Transformer-based image generation backbone, the MRF improves image quality by utilizing learned spatial and label compatibility relations in the latent token space.*

Consider a common vocabulary size of $V = 8192$. For a full sized 16×16 image, there are $8192^{256} = 6.7 \times 10^{1002}$ different arrangements of tokens, many of which will represent some kind of "garbage" images that lie outside the manifold of photorealistic images. Intuitively, a structured prediction mechanism that accounts for the compatibility of token arrangements could significantly reduce this massive search space of token arrangements and make the token prediction models more efficient.

We propose a probabilistic graphical model for this structured prediction task. Specifically, we formulate finding the token arrangement as maximum a posteriori (MAP) inference of an MRF model, as described in the following. The high-level idea is illustrated in Figure 3.

Let $i \in \{1, 2, \ldots, n\}$ denote the location indices of the token image, arranged in row-major order. Let $\mathcal{L} = \{l_1, l_2, \ldots, l_V\}$ be the token labels, which are used to index each element in the codebook of V tokens. For a 16×16 token image with vocabulary size 8192, we have $n = 256$ and $V = 8192$. Define a random variable $X_i \in \mathcal{L}$ for each $i = 1, 2, \ldots, n$ to hold the token assignment for the i^{th} location. The collection of these random variables $\mathbf{X} = [X_1, X_2, \ldots, X_n]$ then forms a random field, where the value of one variable depends on that of the others. We can then model the probability of an assignment to this random field (and therefore a token arrangement on the grid) with the Gibbs measure:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{x})), \tag{1}$$

where $\mathbf{x} \in \mathcal{L}^n$ is a given token arrangement and $Z = \sum_{\mathbf{x}} \exp(-E(\mathbf{x}))$ is the partition function. The "energy" $E(\mathbf{x})$ of an assignment $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is modeled with two components: the *unary* component $u_i(.)$ and the *pairwise* component $p_{ij}(.,.)$:

Algorithm 1 *The MRF Inference Algorithm*

$Q_i(k) \leftarrow \text{softmax}(f_i(k)), \forall(i,k)$
for `num_iterations` **do**
  $Q_i(k) \leftarrow \sum_{j=1}^{n} \mathbf{W^s}_{ij} Q_j(k), \forall(i,k)$
  $Q_i(k) \leftarrow \sum_{k'=1}^{V} \mathbf{W^c}_{kk'} Q_i(k'), \forall(i,k)$
  $Q_i(k) \leftarrow Q_i(k) + f_i(k), \forall(i,k)$
  $Q_i(k) \leftarrow \text{softmax}(Q_i)(k), \forall(i,k)$
end for
return Q

$$E(\mathbf{x}) = \sum_{i=1}^{n} u_i(x_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}(x_i, x_j). \tag{2}$$

The unary component captures the confidence of the neural network prediction model, such as a Transformer model, for a given token and a location. Therefore, given a condition y, such as a text prompt or pre-fixed tokens, if the neural network's predicted logit value for location i and label x_i is $f_i(x_i, y)$, we set:

$$u_i(x_i) = -f_i(x_i, y). \tag{3}$$

Note that we use negative logits because the energy function is in the log domain and a high energy corresponds to a low probability. We drop conditioning on y hereafter to keep the notation uncluttered. Also note that our MRF formulation is not conditioned on y.

The pairwise component, $p_{ij}(x_i, x_j)$, captures the compatibility of the label x_i assigned to the location i and the label x_j assigned to the location j. It encodes the notion that while some pairs of tokens are highly compatible with each other and can appear in the same image, other pairs are highly incompatible. For example, a token representing an eye of a giraffe is more likely to appear next to a token representing a different part of a giraffe face, than a token representing something completely different like a part of car wheel. We factorize this pairwise compatibility into two parts: the spatial similarity $s(i, j)$ between the locations i and j (for example, if i and j are close to each other in the 2D token image, they will be strongly related) and the label compatibility $c(x_i, x_j)$ between the tokens x_i and x_j (for example, highly compatible tokens are able to coexist with each other). We therefore have:

$$p_{ij}(x_i, x_j) = -c(x_i, x_j)s(i, j). \tag{4}$$

In classic MRFs, the pairwise interactions exist only between neighboring pixels. In contrast, for increased flexibility, we allow interactions between all pairs of locations,

similar to the fully-connected CRFs in the image segmentation setting [14, 36]. However, there are a number of important differences in our formulation compared to the fully-connected CRFs in image segmentation: in the latter, spatial similarity $s(i, j)$ is derived conditioned on the input image (hence the name *conditional* random fields), using Gaussian potentials in spatial and bilateral domains. This Gaussian assumption is crucial for the tractability of their models since the image segmentation CRFs work in a large image grid: in practical implementations pixels that are far away by more than a few standard deviations of the Gaussian kernel are considered not connected [1]. In contrast, we make our graphical model truly fully-connected and learn $s(i, j)$ with backpropagation without fixing them to be Gaussian. Furthermore, the CRFs in image segmentation can assume a Potts model for label compatibility because assigning the same label to nearby pixels generally improves the smoothness of the segmentation. In our application, on the other hand, it is not straightforward to assign semantic meanings to tokens and Potts model does not intuitively makes sense since the same token at similar locations does not increase the meaningfulness of a token assignment. We therefore learn the pairwise connections $p_{ij}(.,.)$, completely from data without using any priors or heuristics. Thus, our MRF formulation has two learnable weight matrices: $\mathbf{W^s}$, with $\mathbf{W^s}_{ij} := s(i, j)$ and $\mathbf{W^c}$, with $\mathbf{W^c}_{kk'} := c(k, k')$.

Given our probabilistic graphical model, finding the final token arrangement amounts to finding the assignment $\mathbf{x}$ that maximizes $P(\mathbf{X} = \mathbf{x})$. This can be done efficiently via mean-field inference, where we approximate $P(\mathbf{X}) \approx Q(\mathbf{X}) := \prod_i Q_i(X_i)$, with $Q_i(.)$ being the marginal distribution for X_i. The distribution $Q(\mathbf{X})$ is then iteratively refined to minimize the KL divergence between P and Q. We refer the reader to [13] and [14] for more details on the derivations. The resulting inference algorithm is summarized in Algorithm 1. Note that all operations of this algorithm can be implemented via simple matrix multiplication and other common operations such as $\mathrm{softmax}(.)$, which are readily available in any deep learning library. Importantly, the cost our MRF inference is negligible compared to prediction with a large Transformer model.

4. MarkovGen

We now demonstrate the benefits of the proposed MRF model by using it to speed up the state-of-the-art Muse image generation model [2]. We achieve this speed-up by replacing the last few sampling steps of Muse with MRF inference. Specifically, we let Muse execute the first few steps and then use our extremely lightweight MRF inference to fast-forward the remaining steps. This model, dubbed MarkovGen, improves the speed of image generation by $1.5\times$ while simultaneously also improving quality.

The Muse model works in the discrete VQGAN token space [8], which is gradually emerging as a centerpiece of many text-to-image generation algorithms. Muse generates images by first performing a series of inference steps with the base model to predict a small grid of 16×16 image tokens, conditioned on text embeddings generated by a T5-XXL [20] text encoder. This is followed by a few steps of the super-resolution (SR) model to predict a larger grid of 32×32 image tokens by conditioning on both the text embeddings and the tokens generated by the base model. Due to the larger set of tokens, a single iteration of the SR transformer model is substantially more computationally expensive than that of the base model. We exploit this multi-scale approach to speed up inference efficiency by using more steps in the base model, followed by much fewer steps with the SR model. Once SR tokens are generated, the VQGAN [8] detokenizer is used to render the image in pixel space.

Model	Time (ms)
Muse base (single step)	10.40
Muse super-resolution (single step)	24.00
MRF inference on base	0.29
MRF inference on super-resolution	0.29
T5-XXL inference	0.30
Detokenizer	0.15
Muse	442.05
MarkovGen (ours)	281.03

Table 1. *Average inference times for different components of the MarkovGen models on a TPUv4 device. The* MRF *inference is almost free compared to the costs of the Muse Transformer models. Furthermore, MRF inference is independent of the image resolutions (rows 3 and 4). We make Muse inference* $1.5\times$ *faster by introducing the MRF model.*

The goal of MarkovGen is to fast-forward the later part of the Muse model and replacing it with the MRF outlined in the previous section. To achieve this, we train the MRF to match the final predictions of the Muse model, given the output at an intermediate step. By fast-forwarding after the step k, out of a total of n, we instantly save $(n - k)/n \times 100\%$ of the Muse model's inference time. This is because the inference time of the MRF is negligible compared to that of the Muse steps as shown in Table 1. The same strategy is used for both the base model and the SR model, to achieve an overall boost of $1.5\times$ in inference speed.

Muse determines the token values independently at each sampling stage, and our structured prediction, enforces the learned compatibility relations jointly on the different token values, and thereby leading to improved quality as demonstrated in the experiments.

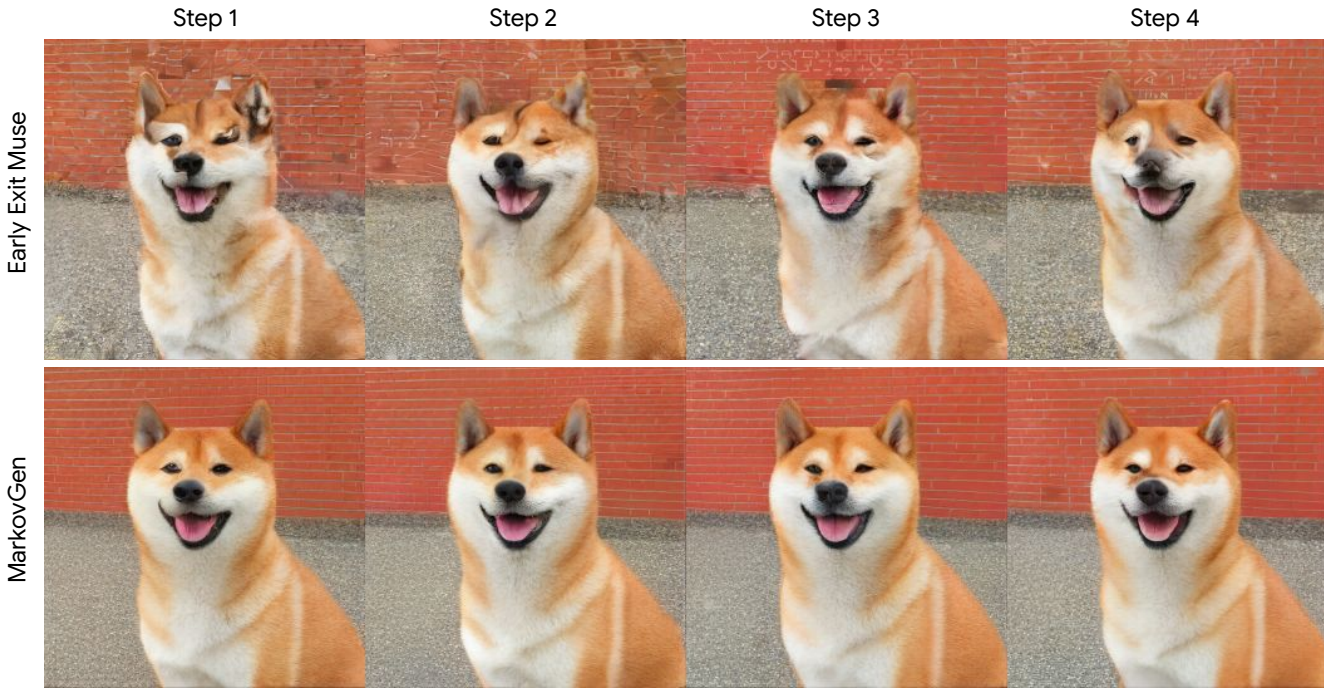

Figure 4. *The first four steps of the Muse super-resolution model without (top) and with (bottom) the application of the MarkovGen MRF model. Note that the MRF fixes complex object structures such as the dog's face as well as texture-inconsistencies in areas such as the brick wall. MarkovGen generates good looking high quality images starting from the first step.*

5. Experiments

In this section, we show that MarkovGen achieves both faster inference and improved image quality compared to Muse, which it uses as its backbone. Muse by itself has already been shown to be much faster than other state-of-the-art text-to-image models such as Dall-E, Dall-E 2, Parti, Imagen, and Stable Diffusion [2], outperforming the second-fastest method by a factor of approximately $3\times$ (Table 3 of [2]). Furthermore, as evidenced by Table 1 & 2 of [2], Muse achieves better quality results compared to these methods, as measured with the FID scores. Human evaluation results for image quality in [2] showed that humans preferred Muse outputs for 70.6% prompts while Stable Diffusion was preferred for only 25.4%. Since Muse is already shown to outperform other state-of-the-art methods in terms of both speed and quality, we focus on comparing our results to that of Muse.

Model and Dataset: We use a Muse model with approximately 1.7B parameters, trained on the WebLI dataset [5]. This model was generously made available to us by the authors of the Muse paper. We refer the reader to [2] for more details on the architecture and the training setup of Muse. The same WebLI dataset was used to train the MRF model.

MRF Training: We train two MRF models, one for fast-forwarding the base and one for SR model respectively. Each model contains two weight matrices for spatial and label compatibilities: 256×256 $\mathbf{W^s_{base}}$ and 8192×8192 $\mathbf{W^c_{base}}$ for thr base, and 1024×1024 $\mathbf{W^s_{SR}}$ and 8192×8192 $\mathbf{W^c_{SR}}$ for the SR model. All MRF weights are trained with back-propagation and gradient descent, with the ADAM optimizer. We use a two-stage approach for MRF training. First, we pre-train the MRF model using a self-supervised masked-token prediction loss [2, 6]. Specifically, we obtain VQGAN tokens for an image, randomly mask 20% of them and train the MRF model to predict the masked tokens using the categorical cross-entropy loss. Second, we fine-tune the MRF model to imitate the last $n - k$ steps of the Muse model: Given the output of the Muse model after the kth iteration, the spatial and label compatibility matrices are learned such that the MRF inference matches the final predictions of the Muse model after n iterations using the KL divergence loss. Both base and SR MRF models are trained in the same manner. Both MRF models complete training in just a few hours on TPUv4 chips.

Experimental Setup: The base model operates on a 16×16 token grid with 24 sampling steps to produce 256×256 images. The SR model works on a 32×32 token grid and produces 512×512 images in 8 additional steps. MarkovGen uses both the base and SR MRF models to trade with the base and SR sampling steps of the Muse model, respec-

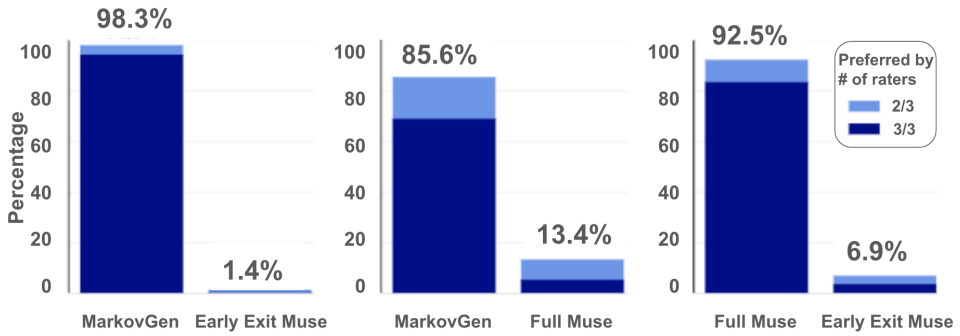

Figure 5. *Percentage of prompts for which human raters prefer images by a given model in a side-by-side comparison. We observe that human raters strongly prefer the images generated by MarkovGen over those of both early exit Muse (left) and even the more expensive and slower full Muse model (center).*

Model	FID
Early Exit Muse base (18 iters)	14.37
Full Muse base (24 iters)	13.13
MarkovGen (18 iters)	12.28

Table 2. *Quantitative evaluation of FID scores on the MS-COCO [16] dataset for* 256×256 *image resolution. MarkovGen outperforms both the Early Exit as well as the full Muse model.*

tively. We apply the base MRF after step 20 of the base Muse model, and the SR MRF after 3 steps of the SR Muse model, cutting down 4 and 5 steps respectively. This results in a speedup of $1.5\times$ for MarkovGen compared to Muse.

In our experiments we compare 3 different models: (A) full Muse, (B) MarkovGen, and (C) an early exit Muse model that also stops Muse iterations early to have comparable speed to MarkovGen, but without the application of the MarkovGen MRF model.

Qualitative Evaluation:

In Figure 4 we study the progression of the generated images during the invocation of the Muse SR model. At each step, we show the output of early exit Muse (top) and the improved result after the application of the MarkovGen MRF (bottom). The results show that while the Muse model slowly improves result quality, MarkovGen provides high quality results already after the first step. We observe that MarkovGen with just 3 SR steps already consistently produces images comparable or better than the full Muse results after the total of 8 SR steps. Figure 6 demonstrates this using a series of Parti Prompts [34], where we compare the results of the full Muse model (left), early exit Muse (middle) and, our MarkovGen model with a $1.5\times$ speed-up (right). We observe that the model is able to produce a wide variety of images ranging from artistic to natural images.

Quantitative Evaluation: Figure 5 summarizes the results of our human evaluation study. Using the 1633 prompts from the Parti promts dataset, we generated three images with full Muse, early exit Muse, and our proposed MarkovGen model respectively. To allow raters to focus on image quality, we use the same random seed across models to ensure that image content and degree of alignment to the prompt are the same across the generated images. Asked to evaluate which image is of higher quality, we present human raters with two generated images side-by-side. Raters are given the option of choosing either image or that they are indifferent. All image pairs are evaluated by 3 independent raters that were hired through a high-quality crowd computing platform. The raters and the authors of this paper were anonymous to each other. For each pairwise comparison, we consider an image to be of higher quality if it is selected by at least 2 raters.

From the results we observe that human raters strongly prefer the images generated by MarkovGen over those of both early exit Muse (left) and even the more expensive and slower full Muse model (center). We also compared early exit Muse to full Muse (right) and verified that human raters can clearly identify the quality improvement achieved by the last stages of the Muse model. This result demonstrates that MarkovGen not only achieves a drastic speed-up of 1.5x over Muse, but also significantly improves image quality.

In addition to human evaluation, Table 2 shows single-shot FID scores on the MS-COCO dataset [16]. We use the base Muse model for this evaluation. Again, in line with the human evaluation, we observe that MarkovGen achieves better results than full Muse, which in turn outperforms early exit Muse.

6. Conclusion

The proposed MarkovGen model showed a significant inference speed-up of $1.5\times$ and a clear quality gain over Muse by fast-forwarding the last few steps of Muse model with MRF inference. The MRF model achieves this by

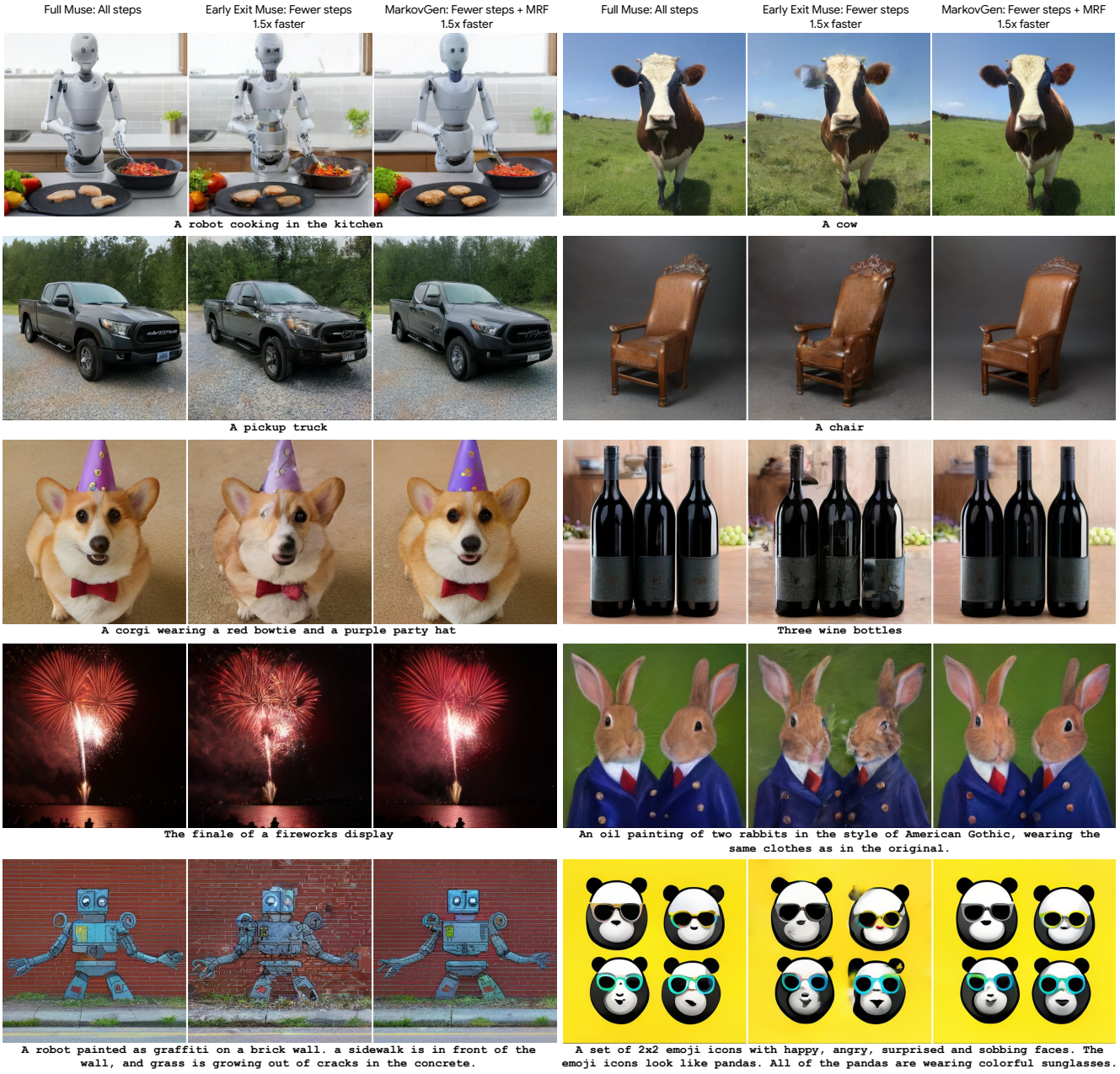

Figure 6. *Within each set of three, MarkovGen (right) speeds up Muse (left) by* $1.5\times$ *and improves image quality. A similar speed up by only reducing the step count with early exit Muse (middle) results in a significant loss of quality.*

learning the spatial and token label compatibility relationships in the discrete VQGAN token space. Our MRF model can be trained in just a few hours, allowing us to use it in conjunction with pre-trained Muse models, to observe almost immediate improvements.

While providing clear benefits over independent per-patch token selection, our current MRF model does not yet utilize the provided text prompt, with text guidance coming solely through the unaries. An interesting direction of future work would be to make the spatial and token compatibility weights be dependent on the text prompt, allowing the MRF (or in this case, the CRF) to adapt to text input.

Another direction of future work lies in training the Muse model itself jointly with the MRF layers, so as to ensure that the unaries produced by Muse are optimal for use with MRF-based decoding.

Acknowledgment

We would like to thank Apurv Suman, Dilip Krishnan, Jarred Barber, Huiwen Chang, Jason Baldridge, and the anonymous reviewers for their valuable feedback.

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Eurographics*, 2010. 5

[2] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. *ICML*, 2023. 2, 3, 5, 6

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 2, 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 3

[5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2022. 6

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 6

[7] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. In *NeurIPS*, 2022. 3

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3, 5

[9] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022. 3

[10] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer, 2022. 3

[11] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2021. 2

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[13] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. 2, 5

[14] Philipp Krahenbuhl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 2, 3, 5

[15] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion, 2023. 3

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[17] Y. Liu M. Ghazvininejad, O. Levy and L. Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*, 2019. 3

[18] Midjourney, 2022. https://www.midjourney.com. 1, 3

[19] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 3

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 2020. 5

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *preprint*, 2022. [arxiv:2204.06125]. 1, 3

[22] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[23] Dominic Rampas, Pablo Pernias, Elea Zhong, and Marc Aubreville. Fast text-conditional discrete denoising on vector-quantized latent spaces. *preprint*, 2022. [arXiv:2211.07292]. 3

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3

[25] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *preprint*, 2022. [arXiv:2205.11487]. 1, 2, 3

[26] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 3

[27] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 3

[28] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. 3

[29] Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. Non-autoregressive text generation with pre-trained language models. In *EACL*, 2021. 3

[30] Zhiqing Sun, Zhuohan Li, Haoqing Wang, Zi Lin, Di He, and Zhi-Hong Deng. Fast structured decoding for sequence models. In *NeurIPS*, 2019. 3

[31] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008. 3

[32] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *preprint*, 2017. [arXiv:1711.00937]. 3

[33] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description, 2022. 3

[34] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. In *ICML*, 2022. 1, 2, 3, 7

[35] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[36] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 3, 5