MARKUP-TO-IMAGE DIFFUSION MODELS WITH SCHEDULED SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Building on recent advances in image generation, we present a fully data-driven approach to rendering markup into images. The approach is based on diffusion models, which parameterize the distribution of data using a sequence of denoising operations on top of a Gaussian noise distribution. We view the diffusion denoising process a sequential decision making process, and show that it exhibits compounding errors similar to exposure bias issues in imitation learning problems. To mitigate these issues, we adapt the scheduled sampling algorithm to diffusion training. We conduct experiments on four markup datasets: formulas (LaTeX), table layouts (HTML), sheet music (LilyPond), and molecular images (SMILES). These experiments each verify the effectiveness of diffusion and the use of scheduled sampling to fix generation issues. These results also show that the markup-to-image task presents a useful controlled compositional setting for diagnosing and analyzing generative image models.

1 INTRODUCTION

Recent years have witnessed rapid progress in text-to-image generation with the development and deployment of pretrained image/text encoders such as CLIP (Radford et al., 2021) and powerful generative processes such as denoising diffusion probabilistic models (Ho et al., 2020). Most existing image generation research focuses on generating natural images conditioned on possibly ambiguous natural language (Saharia et al., 2022; Ramesh et al., 2022). In this work, we instead study the task of markup-to-image generation, where the presentational markup describes exactly one-to-one what the final image should look like.

While the task of markup-to-image generation can be accomplished with standard renderers, we argue that this task has several nice properties for acting as a benchmark for evaluating and analyzing text-to-image generation models. First, the deterministic nature of the problem enables exposing and analyzing generation issues in a setting with known ground truth. Second, the compositional nature of markup language is nontrivial for neural models to capture, making it a challenging benchmark for relational properties. Finally, developing a model-based markup renderer enables interesting applications such as markup compilers that are resilient to typos, or even enable mixing natural and structured commands (Glennie, 1960; Teitelman, 1972).

We build a collection of markup-to-image datasets shown in Figure 1: mathematical formulas, simple tables, sheet music, and molecules (Nienhuys & Nieuwenhuizen, 2003; Weininger, 1988). These datasets can be used to assess the ability of generation models to produce coherent outputs in a structured environment. We then experiment with utilizing diffusion models, which represent the current state-of-the-art in conditional generation of natural images, on these tasks.

The markup-to-image challenge exposes a new class of generation issues for the model. For example, when generating formulas, current models generate perfectly formed output, but often generate duplicate or misplaced symbols (see Figure 2). This type of error is similar to the widely studied *exposure bias* in token-based generation (Ranzato et al., 2015). To help the model fix this class of errors during the generation process, we propose to adapt scheduled sampling (Bengio et al., 2015). Specifically we train diffusion models by using the model's own generations as input such that the model learns to correct its own mistakes.

Math

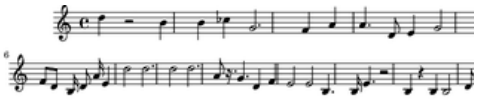
$$\widetilde{\gamma}_{\mathrm{hopf}} \simeq \sum_{n>0} \widetilde{G}_n \frac{(-a)^n}{2^{2n-1}}$$

```
\widetilde \gamma _ { \mathrm { h o p f } } \simeq \sum _ { n > 0 } \widetilde { G } _ { n } { \frac { ( - a ) ^ { n } } { 2 ^ { 2 n - 1 } } }
```

Simple Tables

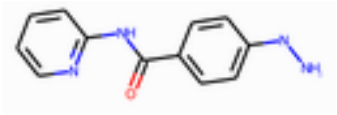
```
... <span style=" font-weight:bold; text-align:center; font-size:150%; " > f j </span> </div> ...
```

Music



```
\relative c'' { \time 4/4 d4 |  r2 b4 b2 | ces4 b4~ g2 f4 | a4 d8 | e4 g16 g2 f2 r4 | des2 d8 d8 f8 e4 d8 a16 b16 | d4 e2 d2. a8~ g4 r16~ e16. d2 f4 b4 e2 | f4. | b16 a16 e4. r2~ c4 r4 b4 d8 b2 | d4 | r8. e8 e2 | r8~ e2 }
```

Molecules



```
COc1ccc(cc1N)C(=O)Nc2ccccc2
```

Figure 1: Markup-to-Image suite with generated images. Tasks include: mathematical formulas (LaTeX), simple tables (HTML), sheet music (LilyPond), and molecules (SMILES). Each example is conditioned on markdown (bottom) and produces a rendered image (top). Evaluation is done by direct comparison with the ground truth image.

Experiments on all four datasets show that the proposed scheduled sampling approach improves the generation quality compared to baselines, and produces surprisingly good image generation for these tasks. Models produce clearly recognizable images for all domains, and often do very well at representing the semantics of the task. Still, there is more to be done to ensure faithful and consistent generation in these difficult deterministic settings. All models, data, and code are publicly available at `https://anonymous.4open.science/r/markup2im/`.

2 Motivation: Diffusion Models for Markup-to-Image Generation

Task Define the task of markup-to-image generation as converting a source in a markup language describing an image to that target image. The input is a sequence of M tokens $x = x_1, \cdots, x_M \in \mathcal{X}$, and the target is an image $y \in \mathbb{R}^{H \times W}$ of height H and width W (for simplicity we only consider grayscale images here). The task of rendering is defined as a mapping $f : \mathcal{X} \to \mathcal{Y}$. Our goal is to approximate the rendering function using a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ trained on supervised examples $\{(x^i, y^i) : i \in \{1, 2, \cdots, N\}\}$. To make the problem tangible, we show several examples of x, y pairs in Figure 1.

Challenge The markup-to-image task contains several challenging properties that are not present in other image generation benchmarks. While the images are much simpler, they act more discretely than typical natural images. Layout mistakes by the model can lead to propagating errors throughout the image. For example, including an extra mathematical symbol can push everything one line further down. Some datasets also have long-term symbolic dependencies, which may be difficult for non-sequential models to handle, analogously to some of the challenges observed in non-autoregressive translation (Gu et al., 2018).

Generation with Diffusion Models Denoising diffusion probabilistic models (DDPM) (Ho et al., 2020) parameterize a probabilistic distribution $P(y_0|x)$ as a Markov chain $P(y_{t-1}|y_t)$ with an initial distribution $P(y_T)$. These models conditionally generate an image by sampling iteratively from the

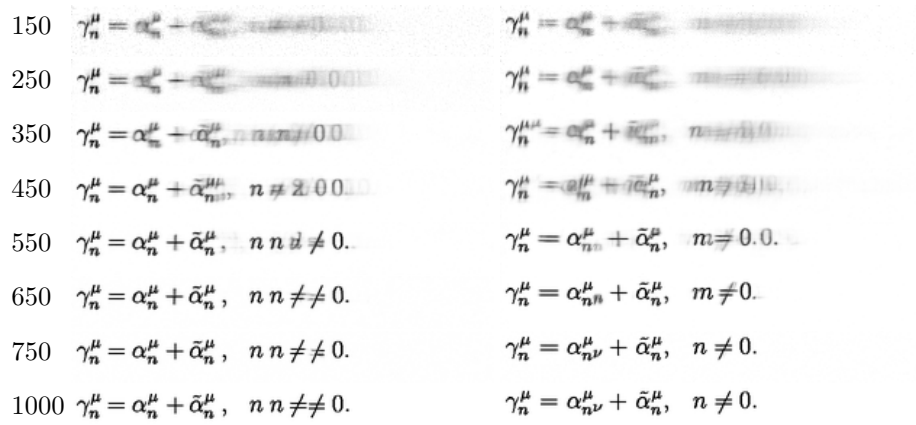


Figure 2: The generation process of diffusion (left) versus diffusion+schedule sampling (right). The numbers on the y-axis are the number of diffusion steps ($T - t$). The ground truth LaTeX is `\gamma_{n}^{\mu}=\alpha_{n}^{\mu}+\tilde{\alpha}_{n}^{\mu},~~~n\neq0.`

following distribution (we omit the dependence on x for simplicity):

$$P(y_T) = \mathcal{N}(0, I)$$
$$P(y_{t-1}|y_t) = \mathcal{N}(\mu_\theta(y_t, t); \sigma_t^2 I)$$

where $y_1, y_2, \cdots, y_T$ are latent variables of the same size as $y_0 \in \mathcal{Y}$, $\mu_\theta(\cdot, t)$ is a neural network parameterizing a map $\mathcal{Y} \to \mathcal{Y}$.

Diffusion models have proven to be extremely effective for generating realistic natural images and are more stable to train than alternative approaches for image generation. These models are surprisingly effective on the markup-to-image dataset as well. However, despite generating realistic images, they make major mistakes in the layout and positioning of the markdown. For an example of these mistakes see Figure 2(left).

We attribute these mistakes to error propagation in the sequential Markov chain. Small mistakes early in the sampling process can lead to intermediate y_t states that may have diverged significantly far from the model's observed distribution. This issue has been widely studied in the inverse RL and autoregressive token generation literature, where it is referred to as *exposure bias* (Ross et al., 2011; Ranzato et al., 2015).

3 Scheduled Sampling for Diffusion Models

In this work, we adapt scheduled sampling, a simple and effective method based on DAgger (Ross et al., 2011; Bengio et al., 2015) from discrete autoregressive models to the training procedure of diffusion. The core idea is to replace the standard training procedure with a biased sampling approach that mimics the test-time trajectories of the model itself. Before describing this approach, we first give a short background on training diffusion models.

Training Diffusion Models Diffusion models maximize an evidence lower bound (ELBO) on the above Markov chain. We introduce an auxiliary Markov chain $Q(y_1, \cdots, y_T|y_0) = \prod_{t=1}^{T} Q(y_t|y_{t-1})$ to compute the ELBO:

$$\log P(y_0) \geq \mathbb{E}_{y_1,\cdots,y_T \sim Q} \log \frac{P(y_0, \cdots, y_T)}{Q(y_1, \cdots, y_T)}$$
$$= \mathbb{E}_Q \left[\log P(y_0|y_1) - \sum_{t=1}^{T} D_{\mathrm{KL}}(Q(y_{t-1}|y_t, y_0) \| P(y_{t-1}|y_t)) - D_{\mathrm{KL}}(Q(y_T|y_0) \| P(y_T)) \right]. \quad (1)$$

Diffusion models fix Q to a predefined Markov chain:

$$Q(y_t|y_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}y_{t-1}, \beta_t I) \quad Q(y_1, \cdots, y_T|y_0) = \prod_{t=1}^{T} Q(y_t|y_{t-1}),$$

where $\beta_1, \cdots, \beta_T$ is a sequence of predefined scalars controlling the variance schedule.

Since Q is fixed, the last term $-\mathbb{E}_Q D_{\mathrm{KL}}(Q(y_T|y_0)\|P(y_T))$ in Equation (1) is a constant, and we only need to optimize

$$\begin{aligned}
&\mathbb{E}_Q\left[\log P(y_0|y_1) - \sum_{t=1}^{T} D_{\mathrm{KL}}(Q(y_{t-1}|y_t, y_0)\|P(y_{t-1}|y_t))\right] \\
&= \mathbb{E}_{Q(y_1|y_0)} \log P(y_0|y_1) - \sum_{t=1}^{T} \mathbb{E}_{Q(y_t|y_0)} D_{\mathrm{KL}}(Q(y_{t-1}|y_t, y_0)\|P(y_{t-1}|y_t)). \qquad (2)
\end{aligned}$$

With large T, sampling from $Q(y_t|y_0)$ can be made efficient since $Q(y_t|y_0)$ has an analytical form:

$$Q(y_t|y_0) = \int_{y_1, \cdots, y_{t-1}} Q(y_{1:t}|y_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}y_0, \sqrt{1-\bar{\alpha}_t}I),$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$.

To simplify the $P(y_{t-1}|y_t)$ terms. Ho et al. (2020) parameterize this distribution by defining $\mu_\theta(y_t, t)$ through an auxiliary neural network $\epsilon_\theta(y_t, t)$:

$$\mu_\theta(y_t, t) = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(y_t, t)).$$

With P in this form, applying Gaussian identities, reparameterization (Kingma & Welling, 2013), and further simplification leads to a final MSE training objective,

$$\max_\theta \sum_{t=1}^{T} \mathbb{E}_{y_t \sim Q(y_t|y_0)} \left\| \frac{y_t - \sqrt{\bar{\alpha}_t y_0}}{\sqrt{1-\bar{\alpha}_t}} - \epsilon_\theta(y_t, t) \right\|^2, \qquad (3)$$

where y_t is the sampled latent, $\bar{\alpha}_t$ is a constant derived from the variance schedule, y_0 is the training image, and ϵ_θ is a neural network predicting the update to y_t that leads to y_{t-1}.

Scheduled Sampling Our main observation is that at training time, for each t, the objective function in Equation (3) takes the expectation with respect to a $Q(y_t|y_0)$. At test time the model uses the learned $P(y_t)$ leading to exposure bias issues like Figure 2.

Scheduled sampling suggests alternating sampling in training from the standard distribution and the model's own distribution based on a schedule that increases model usage through training. Ideally, we would sample from,

$$P(y_t) = \int_{y_{t+1}, \cdots, y_T} P(t_T) \prod_{s=t+1}^{T} P(y_{s-1}|y_s),$$

However, sampling from $P(y_t)$ is expensive since it requires rolling out the intermediate steps $y_T, \cdots, y_{t+1}$[1].

We propose an approximation instead. First we use Q as an approximate posterior of an earlier step $t+m$, and then roll out a finite number of steps m from $y_{t+M} \sim Q(y_{t+m}|y_0)$:

$$\tilde{P}(y_t|y_0) \triangleq \int_{y_{t+1}, \cdots, y_{t+m}} Q(y_{t+m}|y_0) \prod_{s=t+1}^{t+m} P(y_{s-1}|y_s).$$

[1]There is no analytical solution since the transition probabilities in this Markov chain are parameterized by a neural network μ_θ.

Note that when $m = 0$, $\tilde{P}(y_t|y_0) = Q(y_t|y_0)$ and we recover normal diffusion training. When $m = T - t$, $\tilde{P}(y_t|y_0) = P(y_t)$ if $Q(y_T|y_0) = \mathcal{N}(0, I)$ (which in practice is usually the case). Substituting back, the objective becomes

$$\sum_{t=1}^{T} \mathbb{E}_{y_t \sim \tilde{P}(y_t|y_0)} \left\| \frac{y_t - \sqrt{\bar{\alpha}_t} y_0}{\sqrt{1 - \bar{\alpha}_t}} - \epsilon_\theta(y_t, t) \right\|^2 . \tag{4}$$

To compute its gradients, in theory we need to back-propagate through $\tilde{P}$ since it depends on θ, but in practice to save memory we ignore $\frac{\partial \tilde{P}}{\partial \theta}$ and only consider the original term.

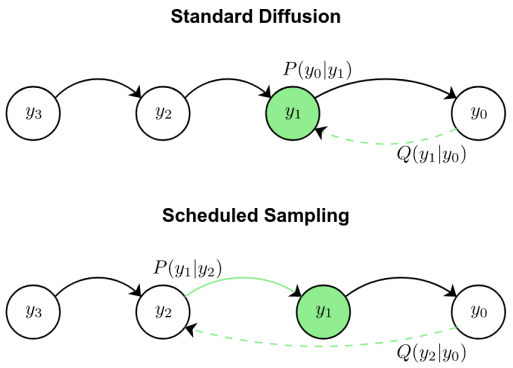


Figure 3: Diffusion samples y_1 from Q. Scheduled sampling instead samples an upstream latent variable y_2 and then y_1 based on the model's Markov chain $P(y_1|y_2)$.

4 Markup-to-Image Setup

4.1 Data

We adapt datasets from four domains to the task of markup-to-image. Table 1 provides a summary of dataset statistics.

Math Our first dataset, LaTeX-to-Math, is a large collection of real-world mathematical expressions written in LaTeX markups and their rendered images. We adapt IM2LATEX-100K introduced in Deng et al. (2016), which is collected from Physics papers on arXiv. IM2LATEX-100K is originally created for the visual markup decompiling task, but we adapt this dataset for the reverse task of markup-to-image. Considering the majority of diffusion-based image generation models require specifying the size of images, we pad all images to size 64×320 and remove images larger than that size. For faster evaluation, we form a smaller test set by subsampling 1,024 examples from the original test set in IM2LATEX-100K .

Simple Tables The second dataset we use is based on the 100k synthesized HTML snippets and corresponding rendered webpage images from Deng et al. (2016). Each HTML snippet contains a nested with a solid border, a random width, and a random float. The maximum depth of a nest is limited to two. We make no change to this dataset, except that we subsample 1,024 examples from the original test set to form a new test set.

Sheet Music We generate a third dataset of sheet music. The markup language LilyPond is a file format for music engraving (Nienhuys & Nieuwenhuizen, 2003). LilyPond is a powerful language for writing music scores: it allows specifying notes using letters and note durations using numbers. One challenge in the LilyPond-to-Sheet music task is to deal with the possible "relative" mode, where the determination of each note relies on where the previous note is. We generate 35k synthetic LilyPond files, and compile them into sheet music. We downsample images by a factor of 2, and then filter out images greater than 192×448.

Molecules The last dataset we use is from the chemistry domain. The input is a string of Simplified Molecular Input Line Entry System (SMILES) which specifies atoms and bonds of a molecule (Weininger, 1988). The output is a scheme of the input molecule. We use a solubility dataset by Wilkinson et al. (2022), containing 19,925 SMILES strings. The dataset is originally proposed to improve the accessibility of chemical structures for deep learning research. 2D molecules images are rendered from SMILES strings using the Python package RDKIT (Landrum et al., 2016). We partition this dataset into training, validation, and test sets. We downsample images by a factor of 2.

4.2 Evaluation

Popular metrics for conditional image generation such as Inception Score (Salimans et al., 2016) or Frëchet Inception Distance (Heusel et al., 2017) evaluate fidelity and high-level semantics of gen-

Dataset	Input Format	Input Length	# Train	# Val	# Test	Image Size	Grayscale
Math	LaTeX Math	113	55,033	6,072	1,024	64×320	Y
Simple Tables	HTML Snippet	481	80,000	10,000	1,024	64×64	Y
Sheet Music	LilyPond File	240	30,902	989	988	192×448	Y
Molecules	SMILES String	30	17,925	1,000	1,000	128×128	N

Table 1: Markup-to-image datasets. Inputs to each dataset are described in Section 4.1 in detail. Input length is measured as the median number of characters in the validation set.

erated images. In markup-to-image tasks, we instead emphasize the pixel-level similarity between generated and ground-truth images as markup describes exactly what the image should look like.

Pixel Metrics Our primary evaluation metric is Dynamic Time Warping (DTW) (Müller, 2007), which calculates the pixel-level similarities of images by treating them as a column time-series. We preprocess images by binarizing them. We treat binarized images as time-series by having image rows as a sequence and image columns as feature vectors representing each row. We evaluate the similarity of generated and ground-truth images by calculating the cost of alignment between two time-series using DTW [2]. We use Euclidean distance as a feature matching metric. We allow minor translations of generated images by allowing some translations during feature matching.

Our secondary evaluation metric is the root squared mean error (RMSE) of pixels between generated and ground-truth images. While RMSE compares two images at the pixel level, one drawback is that RMSE heavily penalizes symbolically equivalent images with minor perturbations. We convert all images to grayscale before calculating RMSE.

Complimentary Metrics Complementary to the above two main metrics, we report one learned and seven classical image similarity metrics. We use CLIP score (Radford et al., 2021) as a learned metric to calculate the similarity between CLIP embeddings of generated and ground-truth images. While CLIP score is robust to minor perturbations of images, it is unclear if CLIP embeddings capture the symbolic meaning of the images in the domains of rendered markups. For classical image similarity metrics [3], we report SSIM (Wang et al., 2004), PSNR (Wang et al., 2004), UQI (Wang & Bovik, 2002), ERGAS (Wald, 2000), SCC (Zhou et al., 1998), and RASE (González-Audícana et al., 2004).

4.3 Experimental Setup

Model For Math, Simple Tables, and Sheet Music datasets, we use GPT-Neo-175M (Black et al., 2021; Gao et al., 2020) as the input encoder, which incorporates source code in its pre-training. For the Molecules dataset, we use ChemBERTa-77M-MLM from DeepChem (Ramsundar et al., 2019; Chithrananda et al., 2020) to encode the input. To parameterize the diffusion decoder, we experiment with three variants of U-Net (Ronneberger et al., 2015): 1) a standard U-Net conditioned on an average-pooled encoder embedding, 2) a U-Net alternating with cross-attention layers over the full resolution of the encoder model (denoted as "+Attn"), and 3) a U-Net with both cross-attention and additional position embeddings on the query marking row ids and column ids (denoted as "+Attn,+Pos") (Vaswani et al., 2017).

Hyperparameters We train all models for 100 epochs using AdamW optimizer (Kingma & Ba, 2014; Loshchilov & Hutter, 2018). The learning rate is set to $1e-4$ with a cosine decay schedule over 100 epochs and 500 warmup steps. We use a batch size of 16 for all models. For scheduled sampling, we used $M = 1$. We linearly increase the scheduled sampling from 0% to 50% since the beginning of the training to the end.

[2] We use the DTW implementation by `https://tslearn.readthedocs.io/en/stable/user_guide/dtw.html`.

[3] We use the similarity metric implementation by `https://github.com/andrewekhalel/sewar`.

Approach	Pixel		Complimentary						
	DTW↓	**RMSE↓**	CLIP↑	SSIM↑	PSNR↑	UQI↑	ERGAS↓	SCC↑	RASE↓
Math									
-Attn, -Pos	27.73	44.72	0.95	0.70	15.35	0.97	2916.76	0.02	729.19
+Attn, -Pos	20.81	39.53	0.96	0.76	16.62	0.98	2448.35	0.06	612.09
+Attn, +Pos	19.45	37.81	0.97	0.78	17.12	0.98	2314.31	0.07	578.58
Scheduled Sampling	**18.81**	**37.19**	0.97	0.79	17.25	0.98	2247.41	0.07	561.85
Simple Tables									
+Attn, -Pos	6.09	22.89	0.95	0.92	38.55	0.98	2497.51	0.44	624.38
+Attn, +Pos	5.91	22.17	0.95	0.93	38.91	0.98	2409.28	0.44	602.32
Scheduled Sampling	**5.64**	**21.11**	0.95	0.93	40.20	0.98	2285.83	0.45	571.46
Sheet Music									
+Attn, -Pos	81.21	45.23	0.97	0.67	15.10	0.97	3056.72	0.02	764.18
+Attn, +Pos	80.63	45.16	0.97	0.68	15.11	0.97	3032.40	0.02	758.10
Scheduled Sampling	**79.76**	**44.70**	0.97	0.68	15.20	0.97	2978.36	0.02	744.59
Molecules									
+Attn, -Pos	24.87	38.12	0.97	0.61	16.66	0.98	2482.08	0.00	620.52
+Attn, +Pos	24.95	38.15	0.96	0.61	16.64	0.98	2455.18	0.00	613.79
Scheduled Sampling	**24.80**	**37.92**	0.96	0.61	16.69	0.98	2467.16	0.00	616.79

Table 2: Evaluation results of markup-to-image generation across four datasets. (+/-)Attn indicates a model with or without attention, and (+/-)Pos is a model with or without positional embeddings. Scheduled Sampling is applied to the training of models with attention and positional embeddings.

Implementation Details Our code is built on top of the HuggingFace diffusers library[4]. We use 1 Nvidia A100 GPU to train on the Math, Simple Tables, and Molecules datasets; We use 4 A100's to train on the Sheet Music dataset. Training takes approximately 25 minutes per epoch for Math and Simple Tables, 30 minutes for Sheet Music, and 15 minutes for Molecules. Although one potential concern is that the scheduled sampling approach needs more compute due to the extra computation to get $\tilde{P}$ for $m > 0$, in practice we find that the training speed is not much affected: on the Math dataset, scheduled sampling takes 24 minutes 59 seconds per training epoch, whereas without scheduled sampling it takes 24 minutes 13 seconds per epoch.

5 RESULTS

Table 2 summarizes the results of markup-to-image tasks across four domains. We use DTW and RMSE as our primary evaluation metrics to make our experimental conclusions. First, we train and evaluate the variations of diffusion models on the math dataset. Comparing the model with attention or without attention, using attention in the model results in a significant improvement by reducing DTW (25% reduction) and RMSE (12% reduction). Therefore, we always use attention for experiments in other datasets. We observe that additionally, using positional embeddings is helpful for the math dataset. The proposed scheduled sampling approach improves the model's performance using attention and positional embeddings.

We observe a similar trend in the other three datasets–simple tables, music, and molecules. Using positional embeddings improves the performance measured by DTW and RMSE (except the Molecules dataset). Training models with the proposed scheduled sampling achieve the best results consistently across all the datasets. As noted in Figure 2, we can qualitatively observe that schedule sampling, which exposes the model to its own generations during training time, comes with the benefits of the model being capable of correcting its own mistakes at inference time.

Absolute Evaluation Our evaluation metrics enable relative comparisons between models in the markup-to-image task. However, it remains unclear how capable the models are in an absolute

[4] https://github.com/huggingface/diffusers

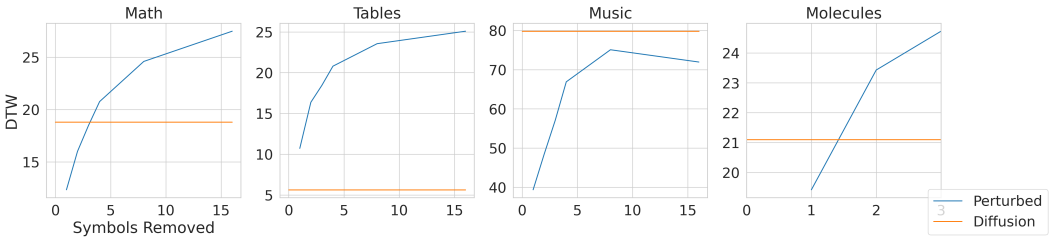


Figure 4: Perturbation results.

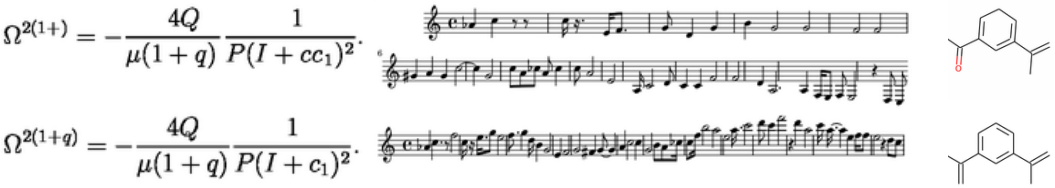


Figure 5: Qualitative results. The top row is model-generated images in Math, Music and Molecules domains (from left to right). The bottom row is the gold images.

sense–if the models are generating near-perfect images or even the best model is missing a lot of symbols. We investigate this question by removing an increasing number of symbols from the gold images and evaluating the perturbed images against the gold images. Our results in Figure 4 highlight our best model in expectation, performing roughly equivalent to the gold images with three symbols removed in the math domain. On the other hand, our best model performs better than gold images with only a single symbol removed in expectation, indicating that our best model adapts to the simple table's domain well.

Qualitative Analysis We perform qualitative analysis on the results of our best models, and we observe that the diffusion models show a different level of adaptation to four domains. First, we observe diffusion models fully learn the simple tables domain, where generated images are equivalent to the gold images for human eyes. Diffusion models perform moderately well for the math and molecules domain. In the math domain, diffusion models manage to generate images similar to the gold images most of the time, less frequently in the chemical domain. We spot frequent frequent failure modes such as dropping a few symbols, adding extra symbols and repeating symbols as illustrated in Figure 5.

In the music domain, diffusion models struggle by generating images that deviate significantly from the gold images. Despite this, we observe diffusion models manage to generate the first few symbols for most of the cases. The intrinsic difficulty of the music domain is a long chain of dependency of symbols from left to right, and the limited number of denoising steps might be a bottleneck to generating images containing this long chain.

We provide the additional qualitative results for all four datasets in the appendix.

6 Related Work

Text-to-Image Generation Text-to-image generation has been broadly studied in the machine learning literature, and several model families are adopted to approach the task. Generative Adversarial Networks (Goodfellow et al., 2014) is one of the popular choices to generate realistic images from text prompts. Initiating from the pioneering work by Reed et al. (2016a) on text-to-image generation in the bird and flower domains, researchers develop methods to improve the quality of text-to-image generation via progressive refinement (Zhang et al., 2017; 2018; Zhu et al., 2019; Tao et al., 2020), cross-modal attention mechanisms (Xu et al., 2018; Zhang et al., 2021), as well as spatial and semantic modeling of objects (Koh et al., 2021; Reed et al., 2016b; Hong et al., 2018; Hinz et al., 2019). Another common method is VQ-VAE-based (Van Den Oord et al., 2017). In

this approach, text-to-image generation is treated as a sequence-to-sequence task of predicting discretized image tokens autoregressively from text prompts (Ramesh et al., 2021; Ding et al., 2021; Gafni et al., 2022; Gu et al., 2022; Aghajanyan et al., 2022; Yu et al., 2022).

Diffusion models (Ho et al., 2020) are the most recent progress in text-to-image generation. The simplicity in training diffusion models introduces significant utility, which often reduces to the minimization of mean-squared error for estimating noises added to images (Ho et al., 2020). Diffusion models are free from training instability or model collapses (Brock et al., 2018; Dhariwal & Nichol, 2021), and yet manage to outperform Generative Adversarial Networks on text-to-image generation in the MSCOCO domain (Dhariwal & Nichol, 2021). Diffusion models trained on large-scale image-text pairs demonstrate impressive performance in generating creative natural or artistic images (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022).

So far, the demonstration of successful text-to-image generation models is centered around the scenario with flexible interpretations of text prompts (e.g., artistic image generation). When there is an exact interpretation of given text prompt (e.g., markup-to-image generation), text-to-image generation models are understudied (with a few exceptions such as Liu et al. (2021) who studies controlled text-to-image generation in CLEVR (Johnson et al., 2017) and iGibson (Shen et al., 2021) domains). Prior work reports that state-of-the-art diffusion models face challenges in the exact interpretation scenario. For example, Ramesh et al. (2022) report unCLIP struggles to generate coherent texts based on images. In this work, we propose a controlled compositional testbed for the exact interpretation scenario across four domains. Our study brings potential opportunities for evaluating the ability of generation models to produce coherent outputs in a structured environment, and highlights open challenges of deploying diffusion models in the exact interpretation scenario.

Scheduled Sampling In sequential prediction tasks, the mismatch between teacher forcing training and inference is known as an exposure bias or covariate shift problem (Ranzato et al., 2015; Spencer et al., 2021). During teacher forcing training, a model's next step prediction is based on previous steps from the ground truth sequence. During inference, the model performs the next step based on its own previous predictions. Training algorithms such as DAgger (Ross et al., 2011) or scheduled sampling (Bengio et al., 2015) are developed to mitigate this mismatch problem, primarily by forcing the model to use its own previous predictions during training with some probability. In this work, we observe a problem similar to exposure bias in diffusion models, and we demonstrate that training diffusion models using scheduled sampling improves their performance on markup-to-image generation.

7 Conclusion

In this work, we propose the task of markup-to-image generation, which is different from the typical natural image generation problems in that there are ground truth images and that there is strong compositionality in the task. We adapted/created four instances of this task and showed that they can be used to analyze state-of-the-art diffusion-based image generation models. Motivated by the observation that the diffusion model cannot correct its own mistakes at inference time, we propose to use scheduled sampling to expose it to its own generations during training. Experiments confirm the effectiveness of the proposed approach, yet there is still a big gap between model generations and ground truth, showing that the proposed suite of tasks is suitable for benchmarking image generation approaches.

References

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Yuntian Deng, Anssi Kanervisto, and Alexander M Rush. What you get is what you see: A visual markup decompiler. *arXiv preprint arXiv:1609.04938*, 10:32–37, 2016.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

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. *arXiv preprint arXiv:2203.13131*, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

AE Glennie. On the syntax machine and the construction of a universal compiler. Technical report, CARNEGIE INST OF TECH PITTSBURGH PA COMPUTATION CENTER, 1960.

María González-Audícana, José Luis Saleta, Raquel García Catalán, and Rafael García. Fusion of multispectral and panchromatic images using improved ihs and pca mergers based on wavelet decomposition. *IEEE Transactions on Geoscience and Remote sensing*, 42(6):1291–1299, 2004.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

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. *Advances in neural information processing systems*, 30, 2017.

Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. *arXiv preprint arXiv:1901.00686*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7986–7994, 2018.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Text-to-image generation grounded by fine-grained user attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 237–246, 2021.

Greg Landrum et al. Rdkit: Open-source cheminformatics software. 2016. URL `https://github.com/rdkit/rdkit/`.

Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. *Advances in Neural Information Processing Systems*, 34:23166–23178, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pp. 69–84, 2007.

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. *arXiv preprint arXiv:2112.10741*, 2021.

Han-Wen Nienhuys and Jan Nieuwenhuizen. Lilypond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, volume 1, pp. 167–171. Citeseer, 2003.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

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*, pp. 8821–8831. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. `https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837`.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pp. 1060–1069. PMLR, 2016a.

Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Advances in neural information processing systems*, 29, 2016b.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7520–7527. IEEE, 2021.

Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J Andrew Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.

Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

Warren Teitelman. Automated programmering: the programmer's assistant. In *Proceedings of the December 5-7, 1972, fall joint computer conference, part II*, pp. 917–921, 1972.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Lucien Wald. Quality of high resolution synthesised images: Is there a simple criterion? In *Third conference" Fusion of Earth data: merging point measurements, raster maps and remotely sensed images"*, pp. 99–103. SEE/URISCA, 2000.

Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9 (3):81–84, 2002.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Matthew R Wilkinson, Uriel Martinez-Hernandez, Chick C Wilson, and Bernardo Castro-Dominguez. Images of chemical structures as molecular representations for deep learning. *Journal of Materials Research*, 37(14):2293–2303, 2022.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1316–1324, 2018.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 833–842, 2021.

Jie Zhou, Daniel L Civco, and JA Silander. A wavelet transform method to merge landsat tm and spot panchromatic data. *International journal of remote sensing*, 19(4):743–757, 1998.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5802–5810, 2019.

A QUALITATIVE RESULTS

We provide additional qualitative results from the model trained with scheduled sampling for all the four datasets. The qualitative examples are randomly selected.

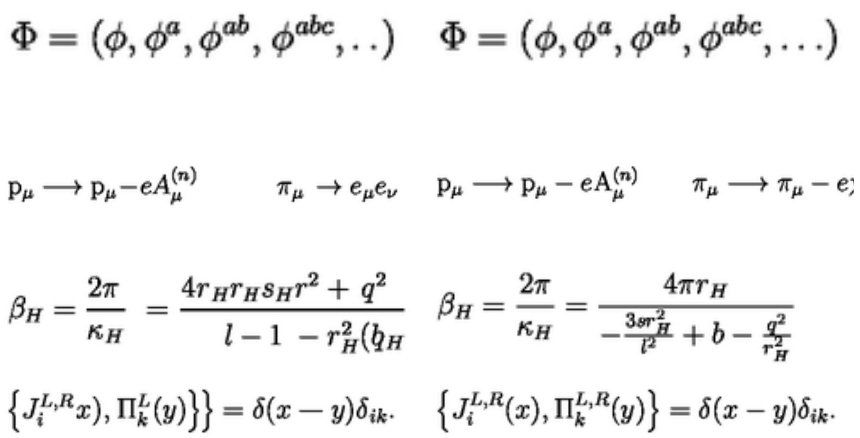


Figure 6: Qualitative results in Math domain. The left column is model-generated images, and the right column is the gold images.

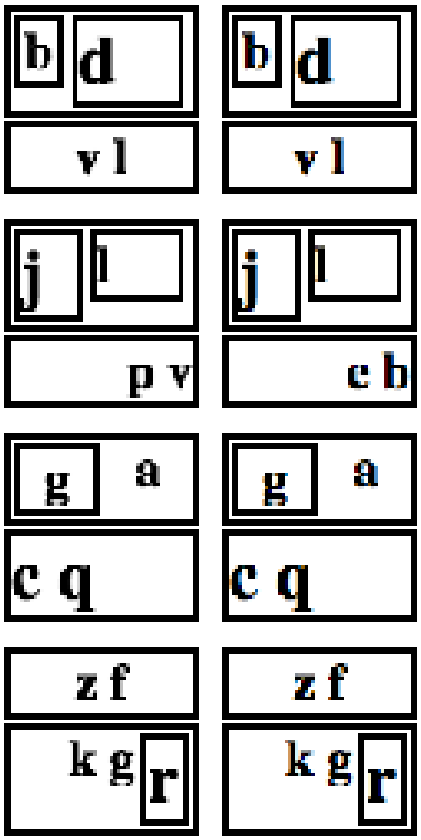


Figure 7: Qualitative results in Simple Tables domain. The left column is model-generated images, and the right column is the gold images.

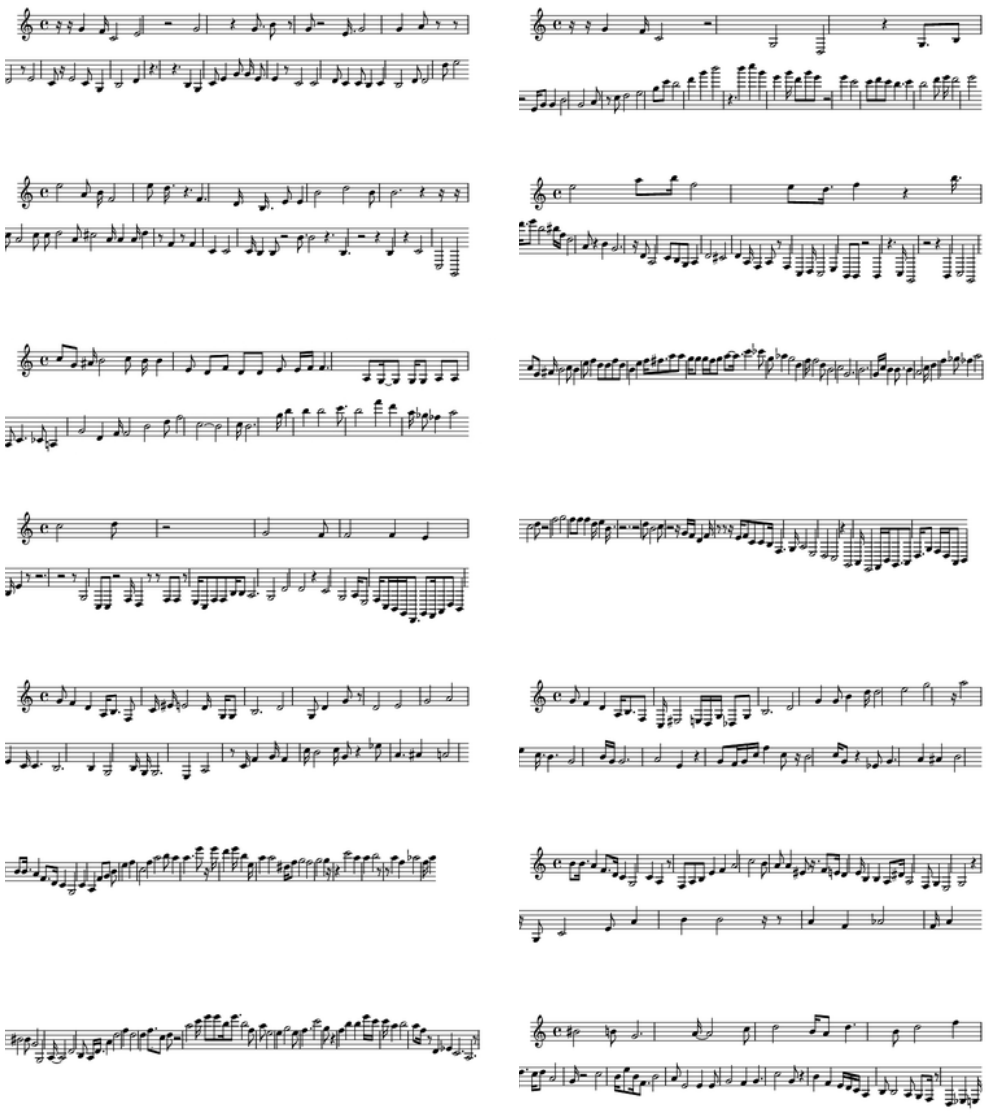


Figure 8: Qualitative results in Music domain. The left column is model-generated images, and the right column is the gold images.

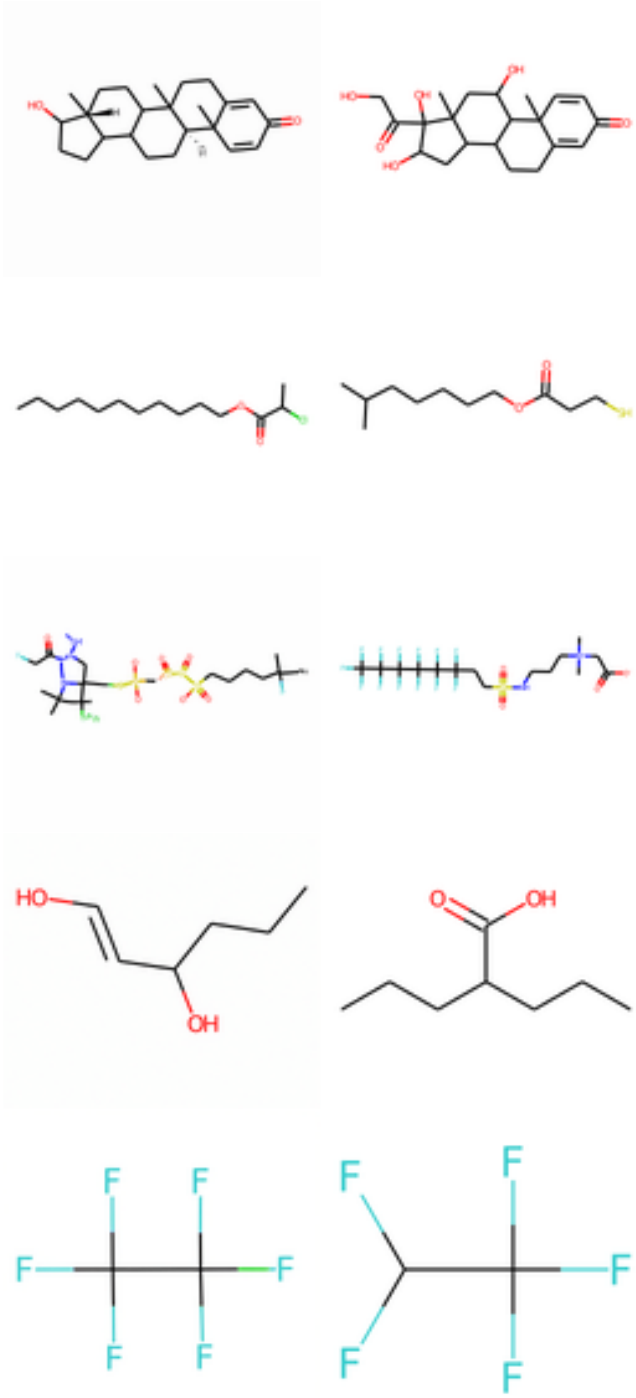


Figure 9: Qualitative results in Molecules domain. The left column is model-generated images, and the right column is the gold images.