CogView: Mastering Text-to-Image Generation via Transformers

Ming Ding[†], Zhuoyi Yang[†], Wenyi Hong[†], Wendi Zheng[†], Chang Zhou[‡], Da Yin[†], Junyang Lin[‡], Xu Zou[†], Zhou Shao[♠], Hongxia Yang[‡], Jie Tang[†♠]
[†]Tsinghua University [‡]DAMO Academy, Alibaba Group [♠]BAAI
`{dm18@mails, jietang@mail}.tsinghua.edu.cn`

Abstract

Text-to-Image generation in the general domain has long been an open problem, which requires both generative model and cross-modal understanding. We propose CogView, a 4-billion-parameter Transformer with VQ-VAE tokenizer to advance this problem. We also demonstrate the finetuning strategies for various downstream tasks, e.g. style learning, super-resolution, text-image ranking and fashion design, and methods to stabilize pretraining, e.g. eliminating NaN losses. CogView (zero-shot) achieves a new state-of-the-art FID on blurred MS COCO, outperforms previous GAN-based models and a recent similar work DALL-E. [1]

Figure 1: Samples from CogView. The text in the first line is either from MS COCO (outside the training set) or user queries on our demo website. The images in the second line are finetuned results for different styles or super-resolution. The actual input text is in Chinese, translated into English here for better understanding. More samples for captions from MS COCO are included in Appendix E.

1 Introduction

> *"There are two things for a painter, the eye and the mind... eyes, through which we view the nature; brain, in which we organize sensations by logic for meaningful expression."* (Paul Cezanne [13])

[1]Codes and models are at `https://github.com/THUDM/CogView`. We also have a demo website running for months at `https://lab.aminer.cn/cogview/index.html` (without post-selection or super-resolution).

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

As contrastive self-supervised pretraining has revolutionized computer vision (CV) [19, 16, 6], visual-language pretraining, which brings high-level semantics to images, is becoming the next frontier of visual understanding [33, 26, 34]. Among various pretext tasks, text-to-image generation expects the model to (1) disentangle shape, color, gesture and other features from pixels, (2) align objects and features with corresponding words and their synonyms, (3) understand the input text and (4) learn complex distributions to generate the overlapping and composite of different objects and features, which, like painting, is beyond basic visual functions (related to eyes and the V1–V4 in brain [17]), representing a higher-level cognitive ability (more related to the angular gyrus in brain [3]).

The attempts to teach machines text-to-image generation can be traced to the early times of deep generative models, when Mansimov et al. [30] added text information to DRAW [15]. Then Generative Adversarial Nets [14] (GANs) began to dominate this task. Reed et al. [37] fed the text embeddings to both generator and discriminator as extra inputs. StackGAN [47] decomposed the generation into a sketch-refinement process. AttnGAN [44] used attention on words to focus on the corresponding subregion. ObjectGAN [25] generated images following a text→boxes→layouts→image process. DM-GAN [48] and DF-GAN [39] introduced new architectures, e.g. dyanmic memory or deep fusion block, for better image refinement. Although these GAN-based models can perform reasonable synthesis in simple and domain-specific dataset, e.g. Caltech-UCSD Birds 200 (CUB), the results on complex and domain-free scenes, e.g. MS COCO [27], are far from satisfactory.

Recent years have seen a rise of auto-regressive generative models. Generative Pre-Training (GPT) models [32, 4] leveraged Transformers [42] to learn language models in large-scale corpus, greatly promoting the performance of natural language generation and few-shot language understanding [28]. Auto-regressive model is not nascent in CV. PixelCNN, PixelRNN [41] and Image Transformer [31] factorized the probability density function on an image over its sub-pixels (color channels in a pixel) with different network backbones, showing promising results. However, a real image usually comprises millions of sub-pixels, indicating an unaffordable amount of computation for large models. Even the biggest pixel-level auto-regressive model, ImageGPT [5], was pretrained on ImageNet at a max resolution of 96×96.

The framework of Vector Quantized Variational AutoEncoders (VQ-VAE) [40] alleviates this problem. VQ-VAE trains an encoder to compress the image into a low-dimensional discrete latent space, and a decoder to recover the image from the hidden variable in the stage 1. Then in the stage 2, an auto-regressive model (such as PixelCNN [41]) learns to fit the prior of hidden variables. This discrete compression loses less fidelity than direct downsampling, meanwhile maintains the spatial relevance of pixels. Therefore, VQ-VAE revitalized auto-regressive model in CV [36]. Following this framework, Esser et al. [11] used Transformer to fit the prior and further switches from L_2 loss to GAN loss for the decoder training, greatly improving the performance of domain-specific unconditional generation.

The idea of our CogView is neat and natural: large-scale generative joint pretraining for both text and image (from VQ-VAE) tokens. We collect 30 million high-quality (Chinese) text-image pairs and pretrain a Transformer with 4 billion parameters. However, large-scale text-to-image generative pretraining could be very unstable due to the heterogeneity of data. We systematically analyze the reasons and solved this problem by proposed *Precision Bottleneck Relaxation* and *Sandwich Layernorm*. As a result, CogView greatly advances the quality of text-to-image generation.

A recent work DALL-E [34] independently proposed the same idea, and released earlier than CogView. Compared with DALL-E, CogView steps forward with the following four aspects:

- CogView outperforms DALL-E and previous GAN-based methods at a large margin according to the Fréchet Inception Distance (FID) [20] on blurred MS COCO, and is the first open-source large text-to-image transformer.
- Beyond zero-shot generation, we further investigate the potential of finetuning the pretrained CogView. CogView can be adapted for diverse downstream tasks, such as style learning (domain-specific text-to-image), super-resolution (image-to-image), image captioning (image-to-text), and even text-image reranking.
- The finetuned CogView enables self-reranking for post-selection, and gets rid of an additional CLIP model [33] in DALL-E. It also provides a new metric *Caption Score* to measure the quality and accuracy for text-image generation at a finer granularity than FID and Inception Score (IS) [38].

- CogView is the first large text-to-image transformer trained with pure FP16 computation, thanks to our simple and effective PB-relaxation and Sandwich-LN. Within three lines of code, these tricks can eliminate overflow in forwarding (characterized as NaN losses), stablizing the training, and can be generalized to the training of other transformers.

2 Method

2.1 Theory

In this section, we will derive the theory of CogView from VAE[2] [22]: *CogView optimizes the Evidence Lower BOund (ELBO) of joint likelihood of image and text.* The following derivation also turns into a clear re-interpretation of VQ-VAE if without text $\mathbf{t}$.

Suppose the dataset $(\mathbf{X}, \mathbf{T}) = \{x_i, t_i\}_{i=1}^N$, consisting of N i.i.d. samples of image variable $\mathbf{x}$ and its description text variable $\mathbf{t}$. We assume the image $\mathbf{x}$ can be generated by a random process involving a latent variable $\mathbf{z}$: (1) t_i is first generated from a prior $p(\mathbf{t}; \theta)$. (2) z_i is then generated from the conditional distribution $p(\mathbf{z}|\mathbf{t} = t_i; \theta)$. (3) x_i is finally generated from $p(\mathbf{x}|\mathbf{z} = z_i; \psi)$. We use a shorthand form like $p(x_i)$ to refer to $p(\mathbf{x} = x_i)$ in the following part.

Let $q(\mathbf{z}|x_i; \phi)$ be the variational distribution, which is the output of an encoder ϕ in VAE. The log-likelihood and the evidence lower bound (ELBO) can be written as:

$$\log p(\mathbf{X}, \mathbf{T}; \theta, \psi) = \sum_{i=1}^{N} \log p(t_i; \theta) + \sum_{i=1}^{N} \log p(x_i|t_i; \theta, \psi) \tag{1}$$

$$\geq -\sum_{i=1}^{N} \Bigg(\underbrace{-\log p(t_i;\theta)}_{\text{NLL loss for text}} + \underbrace{\mathop{\mathbb{E}}_{z_i \sim q(\mathbf{z}|x_i;\phi)} [-\log p(x_i|z_i;\psi)]}_{\text{reconstruction loss}} + \underbrace{\mathrm{KL}\big(q(\mathbf{z}|x_i;\phi) \| p(\mathbf{z}|t_i;\theta)\big)}_{\text{KL between } q \text{ and (text conditional) prior}} \Bigg). \tag{2}$$

The framework of VQ-VAE differs with traditional VAE mainly in the KL term. Traditional VAE fixes the prior $p(\mathbf{z}|t_i; \theta)$, usually as $\mathcal{N}(0, \mathbf{I})$, and learns the encoder ϕ. However, it leads to *posterior collapse* [18], meaning that $q(\mathbf{z}|x_i; \phi)$ sometimes collapses towards the prior. VQ-VAE turns to fix ϕ and fit the prior $p(\mathbf{z}|t_i; \theta)$ with another model parameterized by θ. This technique eliminates posterior collapse, because the encoder ϕ is now only updated for the optimization of the reconstruction loss. In exchange, the approximated posterior $q(\mathbf{z}|x_i; \phi)$ could be very different for different x_i, so we need a very powerful model for $p(\mathbf{z}|t_i; \theta)$ to minimize the KL term.

Currently, the most powerful generative model, Transformer (GPT), copes with sequences of tokens over a discrete dictionary. To use it, we make $\mathbf{z} \in \{0, ..., |V| - 1\}^{h \times w}$, where $|V|$ is the size of dictionary and $h \times w$ is the number of dimensions of $\mathbf{z}$. The sequences z_is can be either sampled from $q(\mathbf{z}|x_i; \phi)$, or directly $z_i = \mathrm{argmax}_{\mathbf{z}}\, q(\mathbf{z}|x_i; \phi)$. We choose the latter for simplicity, so that $q(\mathbf{z}|x_i; \phi)$ becomes a one-point distribution on z_i. The Equation (2) can be rewritten as:

$$-\sum_{i=1}^{N} \Bigg(\underbrace{\mathop{\mathbb{E}}_{z_i \sim q(\mathbf{z}|x_i;\phi)} [-\log p(x_i|z_i;\psi)]}_{\text{reconstruction loss}} \underbrace{-\log p(t_i;\theta)}_{\text{NLL loss for text}} \underbrace{-\log p(z_i|t_i;\theta)}_{\text{NLL loss for } \mathbf{z}} \Bigg). \tag{3}$$

The learning process is then divided into two stages: (1) The encoder ϕ and decoder ψ learn to minimize the reconstruction loss. (2) A single GPT optimizes the two negative log-likelihood (NLL) losses by concatenating text t_i and z_i as an input sequence.

As a result, the first stage degenerates into a pure discrete Auto-Encoder, serving as an *image tokenizer* to transform an image to a sequence of tokens; the GPT in the second stage undertakes most of the modeling task. Figure 2 illustrates the framework of CogView.

2.2 Tokenization

In this section, we will introduce the details about text and image tokenizer.

[2]In this paper, **bold** font denotes a radnom variable, and regular font denotes a concrete value. See this comprehensive tutorial [9] for the basics of VAE.

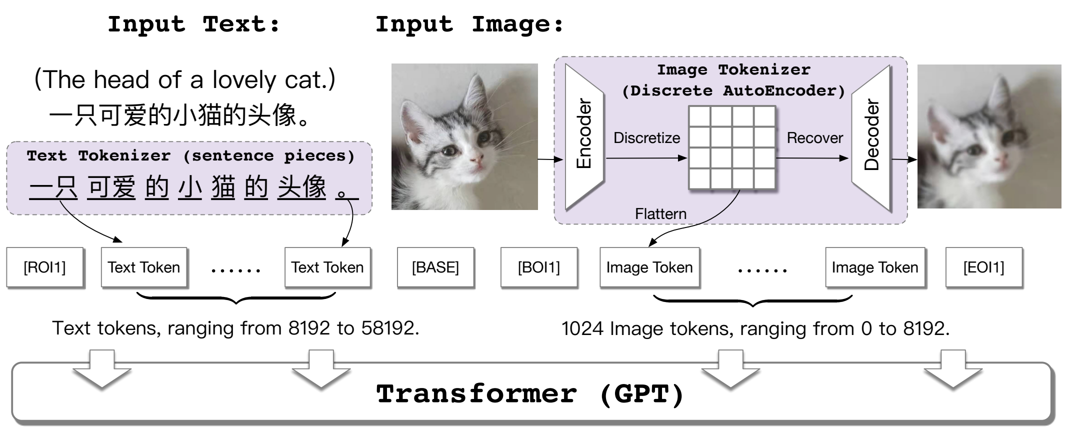


Figure 2: The framework of CogView. `[ROI1]`, `[BASE1]`, etc., are seperator tokens.

Tokenization for text is already well-studied, e.g. BPE [12] and SentencePiece [24]. In CogView, we ran SentencePiece on a large Chinese corpus to extract 50,000 text tokens.

The image tokenizer is a discrete Auto-Encoder, which is similar to the stage 1 of VQ-VAE [40] or d-VAE [34]. More specifically, the Encoder ϕ maps an image x of shape $H \times W \times 3$ into $\text{Enc}_\phi(x)$ of shape $h \times w \times d$, and then each $d-$dimensional vector is quantized to a *nearby* embedding in a learnable dictionary $\{v_0, ..., v_{|V|-1}\}, \forall v_k \in \mathbb{R}^d$. The quantized result can be represented by $h \times w$ indices of embeddings, and then we get the latent variable $\mathbf{z} \in \{0, ..., |V|-1\}^{h\times w}$. The Decoder ψ maps the quantized vectors back to a (blurred) image to reconstruct the input. In our 4B-parameter CogView , $|V| = 8192, d = 256, H = W = 256, h = w = 32$.

The training of the image tokenizer is non-trivial due to the existence of discrete selection. VQ-VAE directly quantizes a vector to its nearest embedding in the dictionary and back-propagates the gradient by straight-through estimator [2], but when the dictionary is large, only a few of embeddings will be used due to curse of dimensionality. A solution in practice is *code revival*, re-initializing the embeddings unused for a long time.

CogView instead follows the original VAE to reparameterize the distribution of latent variable $\mathbf{z}$, where each element in $\mathbf{z}$ are not deterministic at the beginning, but follows a categorical distribution

$$p(\mathbf{z}_{i\times w+j} = v_k|x) = \frac{e^{-\|v_k - \text{Enc}_\phi(x)_{ij}\|_2/\tau}}{\sum_{k=0}^{|V|-1} e^{-\|v_k - \text{Enc}_\phi(x)_{ij}\|_2/\tau}}, \tag{4}$$

where τ is a temperature constant. The standard reparameterization of categorical distribution is *Gumbel-argmax* [21], which means that

$$z_{i\times w+j} = \underset{k}{\text{argmax}}\, g_k - \|v_k - \text{Enc}_\phi(x)_{ij}\|_2/\tau,\ g_k \sim \text{Gumbel}(0,1), \tag{5}$$

is equivalent to sampling from Equation (4). Gumbel-softmax replaces the argmax as differentiable softmax, approximating the one-hot distribution. During training, we gradually decrease the temperature $\tau \to 0$ to make each image correspond a deterministic latent variable, which avoids sampling multiple $\mathbf{z}$ for GPT modeling. This annealing-like method balances the usage of image tokens to achieve a lower reconstruction error.

Compared to DALL-E [34], which uses a uniform distribution over the dictionary for prior in KL-term, we drop the KL-term and train the image tokenizer stably without the scheduling tricks in DALL-E. We also find that competitive reconstruction losses can also be obtained via code revival if implementing carefully.

2.3 Auto-regressive Transformer

The backbone of CogView is a unidirectional Transformer (GPT). The Transformer has 48 layers, with 2560 hidden size, 40 attention heads and 4 billion parameters in total. As shown in Figure 2, four seperator tokens, [ROI1] (reference text of image), [BASE], [BOI1] (beginning of image), [EOI1] (end of image) are added to each sequence to indicate the boundaries of text and image. All the sequences are clipped or padded to a length of 1088.

The pretext task of pretraining is left-to-right token prediction, a.k.a. language modeling. Both image and text tokens are equally treated. DALL-E [34] suggests to lower the loss weight of text tokens; on the contrary, during small-scale experiments we surprisingly find the text modeling is the key for the success of text-to-image pretraining. If the loss weight of text tokens is set to zero, the model will fail to find the connections between text and image and generate images totally unrelated to the input text. We hypothesize that text modeling abstracts knowledge in hidden layers, which can be efficiently exploited during the later image modeling.

We train the model with batch size of 6,144 sequences (6.7 million tokens per batch) for 144,000 steps. The parameters are updated by Adam with max $lr = 3 \times 10^{-4}, \beta_1 = 0.9, \beta_3 = 0.95$, weight decay $= 4 \times 10^{-2}$. The learning rate warms up during the first 2% steps and decays with consine annealing [29]. With hyperparameters in an appropriate range, we find that the training loss mainly depends on the total number of trained tokens (tokens per batch $\times$ steps), which means that doubling the batch size (and learning rate) results in a very similar loss if the same number of tokens are trained. Thus, we use a relatively large batch size to improve the parallelism and lower the percentage of time for communication. We also design a three-region sparse attention to speed up training and save memory without hurting the performance, which is introduced in Appendix B.

2.4 Stabilization of training

Currently, pretraining large models (>2B parameters) usually relies on 16-bit precision to save GPU memory and speed up computation. Many frameworks, e.g. DeepSpeed ZeRO [35], even only support FP16 parameters. However, text-to-image pretraining is very unstable under 16-bit precision. Training a 4B ordinary pre-LN Transformer will quickly result in NaN loss within 1,000 iterations. To stabilize the training is the most challenging part of CogView, which is well-aligned with DALL-E.

We summarize the solution of DALL-E as to *tolerate* the numerical problem of training. Since the values and gradients vary dramatically in scale in different layers, they propose a new mixed-precision framework *per-resblock loss scaling* and store all gains, biases, embeddings, and unembeddings in 32-bit precision, with 32-bit gradients. This solution is complex, consuming extra time and memory and not supported by current frameworks.

CogView is a concurrent work with DALL-E, so that we did not know their solution. We instead *regularize* the values. We find that there are two kinds of instability: overflow (characterized by NaN losses) and underflow (characterized by diverging loss). The following techniques are proposed to solve them.

Precision Bottleneck Relaxation (PB-Relax). After analyzing the dynamics of training, we find that overflow always happens at two *bottleneck* operations, the final LayerNorm or attention.

- In the deep layers, the values of the outputs could *explode* to be as large as $10^4 \sim 10^5$, making the variation in LayerNorm overflow. Luckily, as LayerNorm$(x) =$ LayerNorm$(x/\max(x))$, we can relax this bottleneck by dividing the maximum first[3].
- The attention scores $Q^T K/\sqrt{d}$ could be significantly larger than input elements, and result in overflow. Changing the computational order into $Q^T(K/\sqrt{d})$ alleviates the problem. To eliminate the overflow, we notice that softmax$(Q^T K/\sqrt{d}) =$ softmax$(Q^T K/\sqrt{d} -$ constant), meaning that we can change the computation of attention into

$$\text{softmax}(\frac{Q^T K}{\sqrt{d}}) = \text{softmax}\left(\big(\frac{Q^T}{\alpha\sqrt{d}}K - \max(\frac{Q^T}{\alpha\sqrt{d}}K) \big) \times \alpha \right), \tag{6}$$

  where α is a big number, e.g. $\alpha = 32$. In this way, the maximum (absolute value) of attention scores are also divided by α to prevent it from overflow.

Sandwich LayerNorm (Sandwich-LN). The LayerNorms [1] in Transformers are essential for stable training. Pre-LN [43] is proven to converge faster and more stable than the original Post-LN,

[3]We cannot directly divide x by a large constant, which will lead to underflow in the early stage of training.

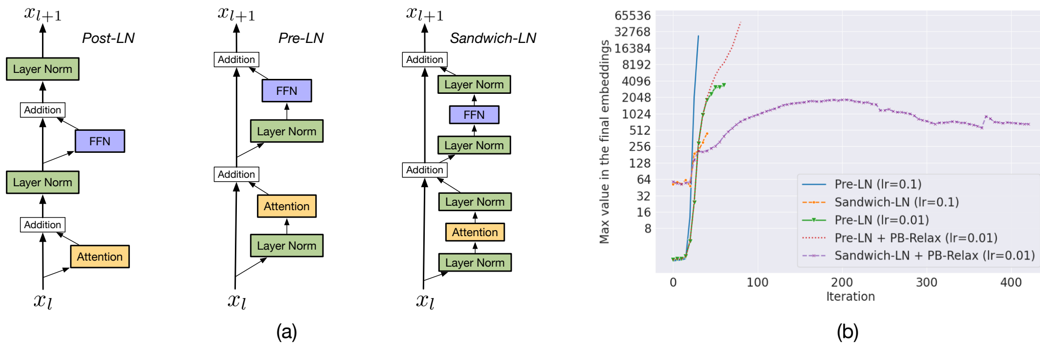


Figure 3: (a) Illustration of different LayerNorm structures in Transformers. Post-LN is from the original paper; Pre-LN is the most popular structure currently; Sandwich-LN is our proposed structure to stabilize training. (b) The numerical scales in our toy experiments with 64 layers and a large learning rate. Trainings without Sandwich-LN overflow in main branch; trainings without PB-relax overflow in attention; Only the training with both can continue.

and becomes the default structure of Transformer layers in recent works. However, it is not enough for text-to-image pretraining. The output of LayerNorm $\frac{(x-\bar{x})\sqrt{d}}{\sqrt{\sum_i (x_i-\bar{x})^2}}\gamma + \beta$ is basically proportional to the square root of the hidden size of x, which is $\sqrt{d} = \sqrt{2560} \approx 50$ in CogView. If input values in some dimensions are obviously larger than the others – which is true for Transformers – output values in these dimensions will also be large ($10^1 \sim 10^2$). In the residual branch, these large values are magnified and be added back to the main branch, which aggravates this phenomenon in the next layer, and finally causes the *value explosion* in the deep layers.

This reason behind value explosion inspires us to restrict the layer-by-layer aggravation. We propose Sandwich LayerNorm, which also adds a LayerNorm at the end of each residual branch. Sandwich-LN ensures the input values of each layer a reasonable scale, and experiments on training 500M model shows that its influence on convergence is negligible. Figure 3(a) illustrates different LayerNorm structures in Transformers.

Toy Experiments. Figure 3(b) shows the effectiveness of PB-relax and Sandwich-LN with a toy experimental setting, since training many large models for verification is not realistic. We find that *deep transformers* (64 layers, 1024 hidden size), *large learning rates* (0.1 or 0.01), *small batch size* (4) can simulate the value explosion in training with reasonable hyperparameters. PB-relax + Sandwich-LN can even stabilize the toy experiments.

Shrink embedding gradient. Although we did not observe any sign of underflow after using Sandwich-LN, we find that the gradient of token embeddings is much larger than that of the other parameters, so that simply shrinking its scale by $\alpha = 0.1$ increases the dynamic loss scale to further prevent underflow, which can be implemented by `emb=emb*alpha+emb.detach()*(1-alpha)` in Pytorch. It seems to slow down the updating of token embeddings, but actually does not hurt performance in our experiments, which also corresponds to a recent work MoCo v3 [7].

Discussion. The PB-relax and Sandwich-LN successfully stabilize the training of CogView and a 8.3B-parameter CogView-large. They are also general for all Transformer pretraining. As an evidence, we used PB-relax successfully eliminating the overflow in training a 10B-parameter GLM [10]. However, in general, the precision problems in language pretraining is not so significant as in text-to-image pretraining. We hypothesize that the root is the heterogeneity of data, because we observed that text and image tokens are distinguished by scale in some hidden states. DALL-E thinks it related to underflow, which is also possible. A thorough investigation is left for future work.

3 Finetuning

CogView steps further on finetuning than DALL-E, which leaves it for future work. Especially, we can improve the text-to-image generation via finetuning CogView for super-resolution and self-reranking. An important finding of us is that we must **load the optimizer states from pretraining** or warmup for a very long period before finetuning large GPT-like models. Since the optimizer states from pretraining are usually not released to the public, their values are largely underestimated. All the finetuning tasks can be completed within one day on a single DGX-2.

3.1 Style Learning

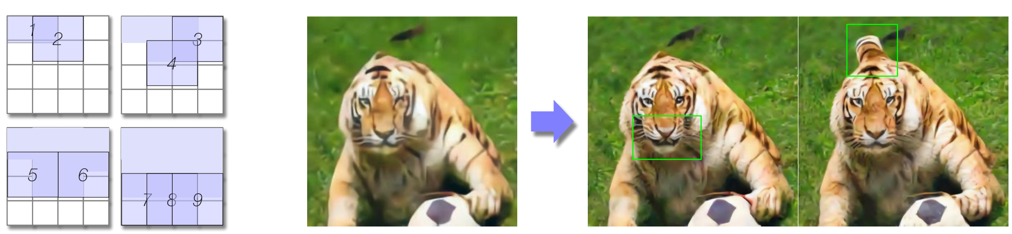
Although CogView is pretrained to cover diverse images as possible, the desire to generate images of a specific style or topic cannot be satisfied well. We finetune models on four styles: Chinese traditional drawing, oil painting, sketch, and cartoon. Images of these styles are automatically extracted from search engine pages including Google, Baidu and Bing, etc., with keyword as "An image of style `{style}`", where `{style}` is the name of style. We finetune the model for different styles separately, with 1,000 images each. During finetuning, the corresponding text for the images are also "An image of `{style}` style". When generating, the text is "A `{object}` of `{style}` style", where `{object}` is the object to generate. Figure 4 shows examples for the styles.

Figure 4: Generated images for "The Oriental Pearl" (a landmark of Shanghai) in different styles.

3.2 Super-resolution

We first finetune a super-resolution model from 16×16 image tokens to 32×32 tokens, and then use it to magnify generated 32×32 tokens to 64×64 tokens by a center-continuous sliding-window strategy in Figure 5, finally resulting in an image of 512×512 pixels.

To prepare data, we crop about 2 million 256×256 patches and downsample them to 128×128. Then we get 16×16 and 32×32 sequence pairs after tokenization for different resolution. The pattern of finetuning sequence is "`[ROI1]` text tokens `[BASE]` `[BOI1]` 256 image tokens `[EOI1]` `[ROI2]` `[BASE]` `[BOI2]` 1024 image tokens `[EOI2]`", exceeding the max position embedding index 1088. As a solution, we recount the position index from 0 at `[ROI2]`. In practice, the model can distinguish the two images well, probably based on whether they can attend to a `[ROI2]` in front.



(a) Center–continuous sliding window (b) Different super–resolution results for "a tiger is playing football".

Figure 5: (a) A 64×64-token image are generated patch-by-patch in the numerical order. The overlaping positions will *not* be overwritten. The key idea is to make the tokens in the 2nd and 4th regions, which are usually faces or other important parts, generated when attending to the whole region. (b) The finetuned super-resolution model does not barely transform the textures, but generates new local structures, e.g. the open mouth or tail in the example.

3.3 Image Captioning and Self-reranking

To finetune CogView for image captioning is straightforward: exchanging the order of text and image tokens in the input sequences. Since the model has already learnt the corresponding relationships between text and images, reversing the generation is not hard. We did not evaluate the performance due to that (1) there is no authoritative Chinese image captioning benchmark (2) image captioning is not the focus of this work. The main purpose of finetuning such a model is for self-reranking.

We propose the *Caption Score* (CapS) to evaluate the correspondence between images and text. More specifically, $\mathrm{CapS}(x,t) = \sqrt[|t|]{\Pi_{i=0}^{|t|} p(t_i|x, t_{0:i-1})}$, where t is a sequence of text tokens and x is the image. $\log \mathrm{CapS}(x,t)$ is the cross-entropy loss for the text tokens, and this method can be seen as an adaptation of inverse prompting [49] for text-to-image generation. Finally, images with the highest CapS are chosen.

Figure 6: 60 generated images for "A man in red shirt is playing video games" (selected at random from COCO), displayed in the order of Caption Score. Most bad cases are ranked in last places. The diversity also eases the concern that CogView might be overfitting a similar image in the training set.

3.4 Industrial Fashion Design

When the generation targets at a single domain, the complexity of the textures are largely reduced. In these scenarios, we can (1) train a VQGAN [11] instead of VQVAE for the latent variable for more realistic textures, (2) decrease the number of parameters and increase the length of sequences for a higher resolution. Our three-region sparse attention (Appendix B) can speed up the generation of high-resolution images in this case.

Figure 7: Generated images for fashion design.

We train a 3B-parameter model on about 10 million fashion-caption pairs, using 50×50 VQGAN image tokens and decodes them into 800×800 pixels. Figure 7 shows samples of CogView for fashion design, which has been successfully deployed on Alibaba Rhino fashion production.

4 Experimental Results

4.1 Machine Evaluation

At present, the most authoritative machine evaluation metrics for general-domain text-to-image generation is the Fréchet Inception Distance (FID) on MS COCO, which is not included in our training set. To compare with DALL-E, we follow the same setting, evaluating CogView on a subset of 30,000 captions sampled from the dataset, after applying a Gaussian filter with varying radius to both the ground-truth images and samples from the models.[4] The captions are translated into Chinese

[4]We use the same evaluation codes with DM-GAN and DALL-E, which is available at `https://github.com/MinfengZhu/DM-GAN`.

for CogView by machine translation. We do not use super-resolution for a fair comparison with DALL-E. Besides, DALL-E generates 512 images for each caption and selects the best one by CLIP, which needs to generate about 15 billion tokens. To save computational resources, we selects the best one from 60 generated images according to their Caption Scores. We finally enhance the constrast of generated images by 1.5. Table 1 shows the metrics for CogView and other methods.

Table 1: Metrics for machine evaluation. Statistics about DALL-E are extracted from their figures. FID-k means that all the images are blurred by a Gaussian Filter with radius k.

Model	FID-0	FID-1	FID-2	FID-4	FID-8	IS	CapS
AttnGAN	35.2	44.0	72.0	108.0	100.0	23.3	0.02763
DM-GAN	**26.0**	39.0	73.0	119.0	112.3	**32.2**	0.02801
DF-GAN	**26.0**	33.8	55.9	91.0	97.0	18.7	0.02802
DALL-E	27.5	28.0	45.5	83.5	85.0	17.9	—
CogView	27.1	**19.4**	**13.9**	**19.4**	**23.6**	18.2	**0.17403**

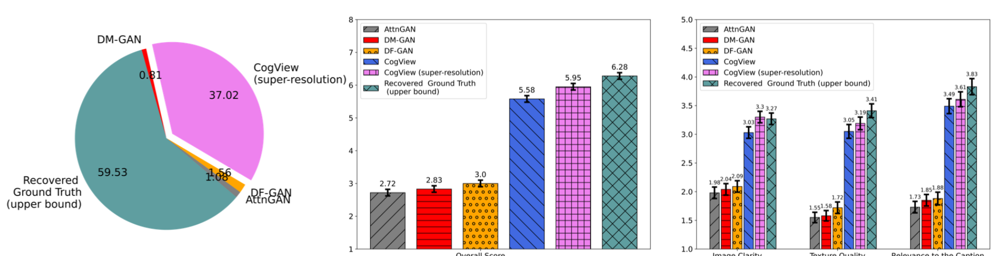
Why does CogView get a better FID on MS COCO than the larger DALL-E? We guess the main reason is that our data for pretraining are mainly photos, similar to the distribution of MS COCO. However, DALL-E learns more rendered and cartoon images, which brings it a better ability to control rare shapes, e.g. pentagon, or spatial position than CogView. The other reason could be the more stable and longer training of CogView.

Caption Score as a Metric. FID and IS are designed to measure the quality of unconditional generation from relatively simple distributions, usually single objects. However, text-to-image generation should be evaluated pair-by-pair. Table 1 shows that DM-GAN achieves the best unblurred FID and IS, but is ranked last in human preference (Figure 8(a)). Caption Score is an absolute (instead of relative like CLIP) score, so that it can be averaged across samples. It should be a better metrics for this task, except samples from CogView itself will be overscored, but it is all right to evalute other models in the future.

4.2 Human Evaluation

Human evaluation is much more persuasive than machine evaluation on text-to-image generation. Our human evaluation consists of 2,950 groups of comparison between images generated by AttnGAN, DM-GAN, DF-GAN, CogView, and recovered ground truth, i.e., the ground truth blurred by our image tokenizer. Details and example-based comparison between models are in Appendix D.

Results in Figure 8 show that CogView outperforms GAN-based baselines at a large margin. CogView is chosen as the best one with probability 37.02%, competitive with the performance of recovered ground truth (59.53%). Figure 8(b)(c) also indicates our super-resolution model consistently improves the quality of images, especially the clarity, which even outperforms the recovered ground truth (256×256).



(a) Human Preference. The percentage of the model to be chosen as best in **all** the questions.

(b) Overall scores (1-10) for the models.

(c) Scores (1-5) for the models on three important aspects.

Figure 8: Human Evaluation results. The recovered ground truth is obtained by first encoding the ground truth image and then decoding it, which is theoretically the upper bound of CogView.

5 Conclusion

We systematically investigate the framework of combining VQVAE and Transformers for text-to-image generation. CogView demonstrates promising results for scalable cross-modal generative pretraining, and also reveals and solves the precision problems probably originating from data heterogeneity. We also introduce methods to finetune CogView for diverse downstream tasks. We hope that CogView could advance both research and application of controllable image generation and cross-modal knowledge understanding.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] H. M. Bonnici, F. R. Richter, Y. Yazar, and J. S. Simons. Multimodal feature integration in the angular gyrus during episodic and semantic retrieval. *Journal of Neuroscience*, 36(20): 5462–5471, 2016.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[5] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[8] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[9] M. Ding. The road from MLE to EM to VAE: A brief tutorial. URL `https://www.researchgate.net/profile/Ming-Ding-2/publication/342347643_The_Road_from_MLE_to_EM_to_VAE_A_Brief_Tutorial/links/5f1e986792851cd5fa4b2290/The-Road-from-MLE-to-EM-to-VAE-A-Brief-Tutorial.pdf`.

[10] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360*, 2021.

[11] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2020.

[12] P. Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994.

[13] J. Gasquet. Cézanne. pages 159–186, 1926.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[15] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015.

[16] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[17] K. Grill-Spector and R. Malach. The human visual cortex. *Annu. Rev. Neurosci.*, 27:649–677, 2004.

[18] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2018.

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6629–6640, 2017.

[21] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] J. Y. Koh, J. Baldridge, H. Lee, and Y. Yang. Text-to-image generation grounded by fine-grained user attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 237–246, 2021.

[24] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL `https://www.aclweb.org/anthology/D18-2012`.

[25] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019.

[26] J. Lin, R. Men, A. Yang, C. Zhou, M. Ding, Y. Zhang, P. Wang, A. Wang, L. Jiang, X. Jia, et al. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*, 2021.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

[29] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[30] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. *ICLR*, 2016.

[31] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[34] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[35] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[36] A. Razavi, A. v. d. Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[37] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016.

[38] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2234–2242, 2016.

[39] M. Tao, H. Tang, S. Wu, N. Sebe, F. Wu, and X.-Y. Jing. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

[40] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017.

[41] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[43] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[44] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. 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*, pages 1316–1324, 2018.

[45] S. Yuan, H. Zhao, Z. Du, M. Ding, X. Liu, Y. Cen, X. Zou, and Z. Yang. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *Preprint*, 2021.

[46] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020.

[47] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[48] M. Zhu, P. Pan, W. Chen, and Y. 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*, pages 5802–5810, 2019.

[49] X. Zou, D. Yin, Q. Zhong, H. Yang, Z. Yang, and J. Tang. Controllable generation from pre-trained language models via inverse prompting. *arXiv preprint arXiv:2103.10685*, 2021.

A Data Collection

We collected about 30 million text-image pairs from the Internet (such as professional Image website, the image search function of search engine, news pictures, a small part of item-caption pairs from Alibaba, etc.) and built a 2.5TB new dataset (After the data was processed into tokens, the size becomes about 250GB). The data are an extension of project WudaoCorpora [45][5]. About 50% of the text is crawled down from Englis websites, whose text is translated into Chinese. In addition, we did not remove the watermarks and white edges in the dataset even though they affect the quality of generated images, because we think it will not influence the conclusions of our paper from the perspective of research.

In order to cover as many common entities as possible, we made a query list consist of 1,200 queries. Every query was an entity name extracted from a large-scale knowledge graph. We chose seven major categories: food, regions, species, people names, scenic, products and artistic works. We extracted top-k entities for each category based on their number of occurrences in the English Wikipedia, where k is manually selected for each category. We collected the top-100 images returned by every major search engine website for each query.

B Sparse Attention

As shown in Figure 9, we design the *three-region sparse attention* (3Rs attention), an implemention-friendly sparse attention for text-to-image generation. Each token attends to all text tokens, all *pivots* tokens and tokens in the blocks in an adjacent window before it.

The pivot tokens are image tokens selected at random, similar to big bird [46]. They are re-sampled every time we enter a new layer. We think they can provide macro information about the image.

The blockwise window attention provides local information, which is the most important region. The forward computation of 1-D window attention can be efficiently implemented inplace by carefully padding and altering the strides of tensors, because the positions to be attended are already continguous in memory. However, we still need extra memory for backward computation if without customized CUDA kernels. We alleviates this problem by grouping adjacent tokens into blocks, in which all the tokens attend to the same tokens (before causally masking). Details about 3Rs attention are included in our released codes.

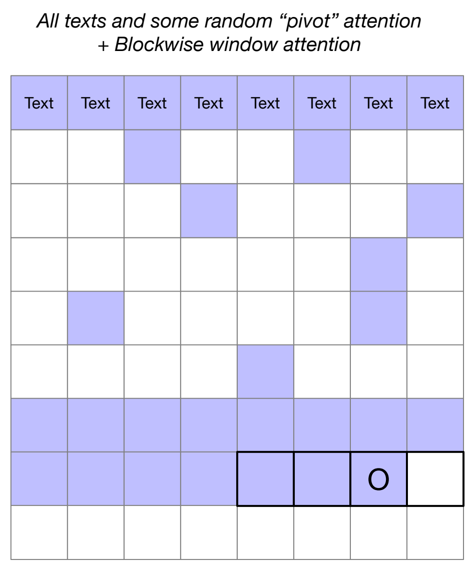


Figure 9: Illustration about our three-region sparse attention. The sequence is shown as a $H \times W$ image and some text tokens in front. Colored grids are all the tokens attended to by the token marked "O". In this case, each block consists of four consecutive tokens.

In our benchmarking on sequences of 4096 tokens, 3Rs attention (768 text and pivot tokens, 768 blockwise window tokens) is $2.5\times$ faster than vanilla attention, and saves 40% GPU memory. The whole training is $1.5\times$ faster than that with vanilla attention and saves 20% GPU memory. With the same hyperparameters, data and random seeds, their loss curves are nearly identical, which means 3Rs attention will not influence the convergence.

However, we did not use 3Rs attention during training the 4-billion-parameter CogView, due to the concerns that it was probably not compatible with finetuning for super-resolution in section 3.2. But 3Rs attention successfully accelerated the training of CogView-fashion without side effects.

[5] `https://wudaoai.cn/data`

C Attention Analysis

To explore the attention mechanism of CogView, we visualize the attention distribution during inference by plotting heat maps and marking the most attended tokens. We discover that our model's attention heads exhibit strong ability on capturing both position and semantic information, and attention distribution varies among different layers.

C.1 Position Information

The attention distribution is highly related to images' position structure. There are a lot of heads heavily attend to fixed positional offsets, especially multiple of 32 (which is the number of tokens a row contains) (Figure 10 (a)). Some heads are specialized to attending to the first few rows in the image (Figure10 (b)) . Some heads' heat maps show checkers pattern (Figure 10 (c)), indicating tokens at the boundary attends differently from that at the center. Higher layers also show some broad structural bias. For example, some heads attend heavily on tokens at top/lower half or the center of images (Figure 10 (d)(e)).

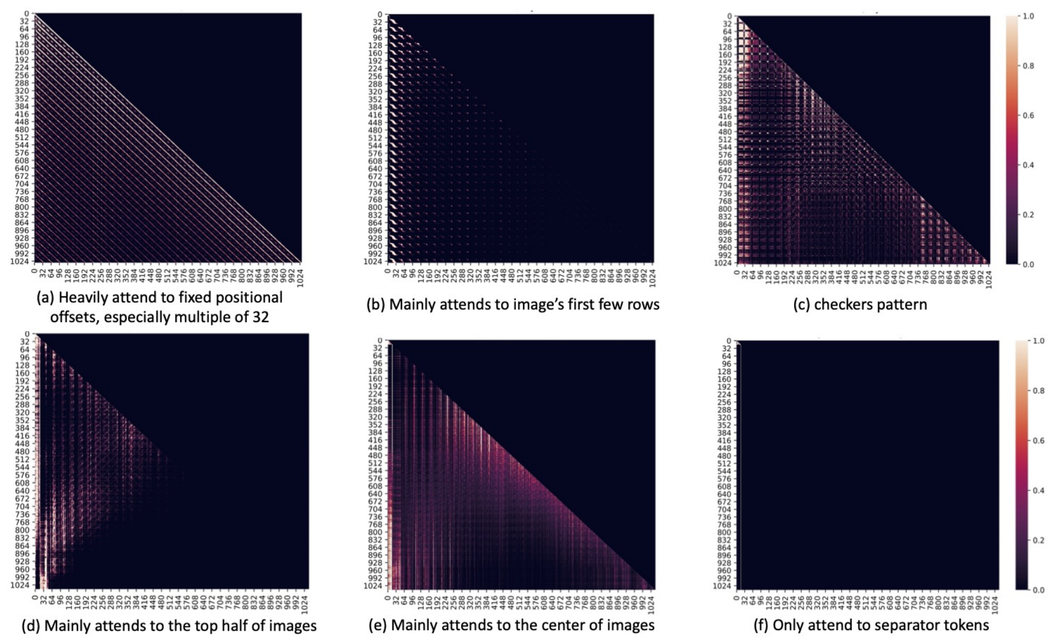


Figure 10: (a, b, c) Our model's attention is highly related to images' positional structure. (d, e) Our model's attention show some broad structural bias. (f) Some heads only attend to a few tokens such as separator token.

C.2 Semantic Information

Our model's attention also relates to semantic information. Some heads highlight major items mentioned in the text. We use "There is an apple on the table, and there is a vase beside it, with purple flowers in it." as input of our experiment. In Figure 11 we marked pixels corresponding to the most highly attended tokens with red dots, and find that attention heads successfully captured items like apple and purple flowers.

C.3 Attention of Different Layers

Attention patterns varies among different layers. Earlier layers focus mostly on positional information, while later ones focus more on image's content. Interestingly, we observe that attention become sparse in the last few layers (after layer 42), with a lot of heads only attend to a few tokens such as separator tokens (Figure 10 (f)). One possible explanation is that those last layers tend to concentrate

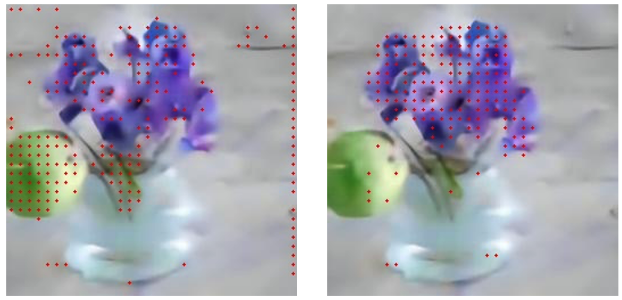


Figure 11: Our model's attention heads successfully captured items like apple and purple flowers. Pixels corresponding to the most highly attended tokens are marked with red dots.

on current token to determine the output token, and attention to separator tokens may be used as a no-op for attention heads which does not substantially change model's output, similar to the analysis in BERT [8]. As the result, the last layers' heads disregard most tokens and make the whole layer degenerate into a feed-forward layer.

D Details about Human Evaluation

To evaluate the performance, we conduct a human evaluation to make comparisons between various methods, similar to previous works [23, 34]. In our designed evaluation, 50 images and their captions are randomly selected from the MS COCO dataset. For each image, we use the caption to generate images based on multiple models including AttnGAN, DM-GAN, DF-GAN and CogView. We do not generate images with DALL-E as their model has not been released yet. For each caption, evaluators are asked to give scores to 4 generated images and the recovered ground truth image respectively. The recovered ground truth image refers to the image obtained by first encoding the ground truth image (the original image in the MS COCO dataset after cropped into the target size) and then decoding it.

For each image, evaluators first need to give 3 scores ($1 \sim 5$) to evaluate the image quality from three aspects: the image clarity, the texture quality and the relevance to the caption. Then, evaluators will give an overall score ($1 \sim 10$) to the image. After all 5 images with the same caption are evaluated, evaluators are required to select the best image additionally.

72 anonymous evaluators are invited in the evaluation. To ensure the validity of the evaluation results, we only collect answers from evaluators who complete all questions and over 80% of the selected best images are accord with the one with the highest overall quality score. Finally, 59 evaluators are kept. Each evaluator is awarded with 150 yuan for the evaluation. There is no time limit for the answer.

To further evaluate the effectiveness of super-resolution, we also introduced a simple A-B test in the human evaluation. Evaluators and captions are randomly divided into two groups E_a, E_b and C_a, C_b respectively. For evaluators in E_a, the CogView images with captions from C_a are generated without super-resolution while those from C_b are generated with super-resolution. The evaluators in E_b do the reverse. Finally, we collected equal number of evaluation results for CogView images with and without super-resolution.

The average scores and their standard deviation are plotted in Figure 8. Several examples of captions and images used in the human evaluation are listed in Figure 12. The evaluation website snapshots are displayed in Figure 13.

E Show Cases for captions from MS COCO

In Figure 14, we provide further examples of CogView on MS COCO.

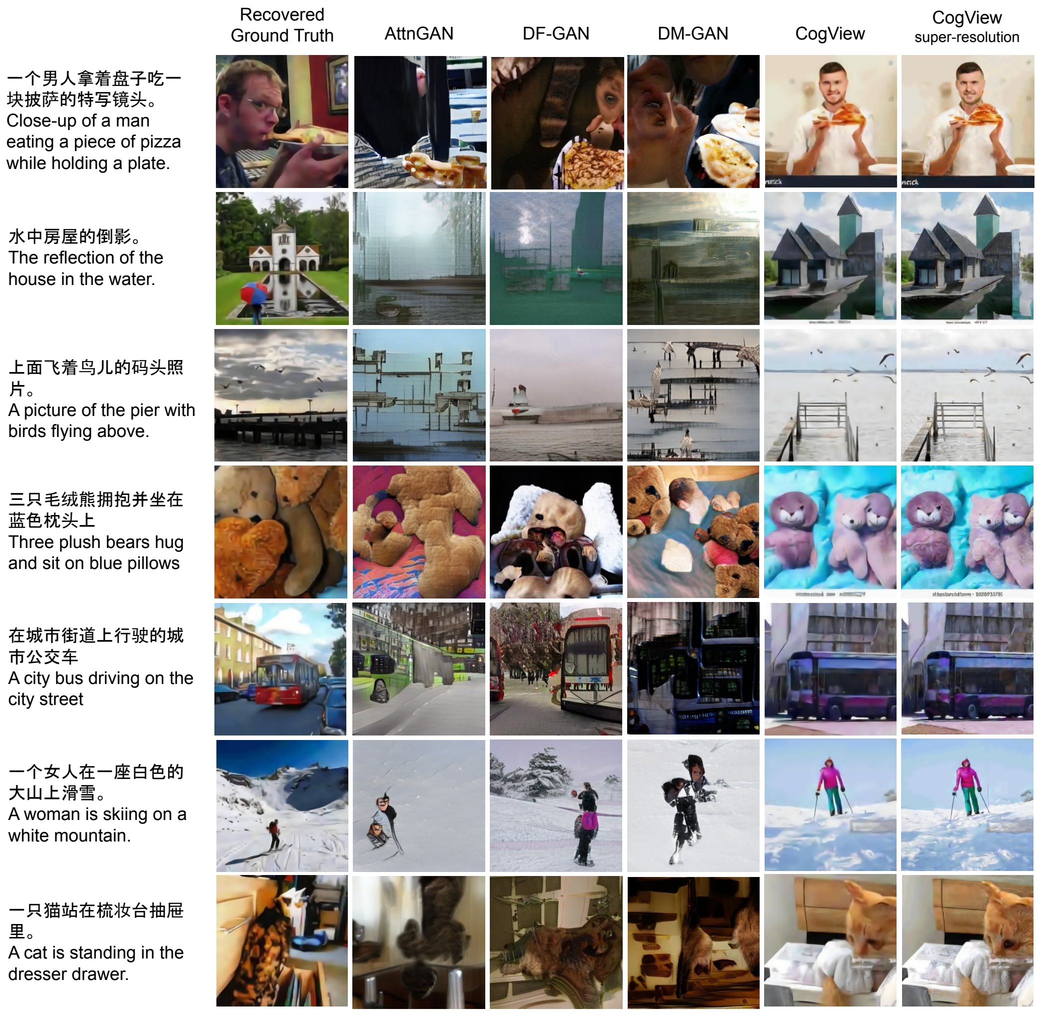


Figure 12: Human Evaluation examples.

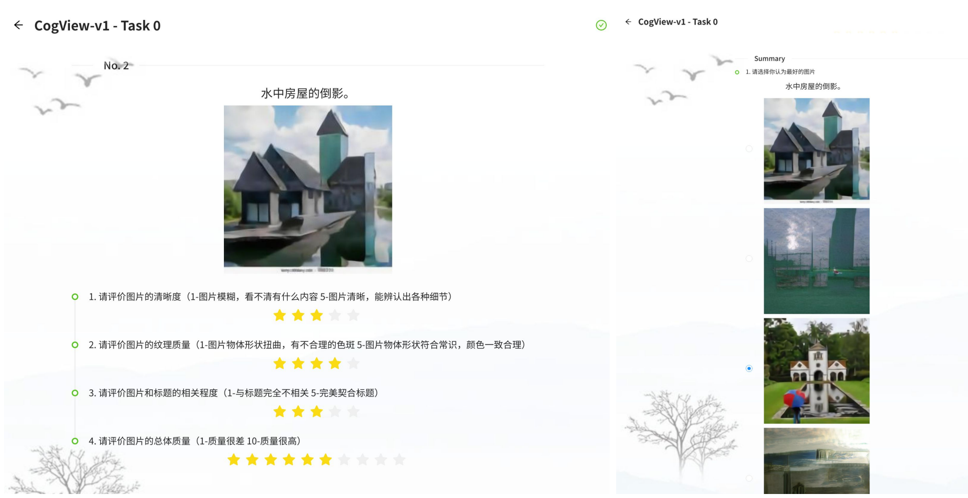


Figure 13: Snapshots of the human evaluation website. The left side is the scoring page for images and the right side is the selection page for all images with the same caption.

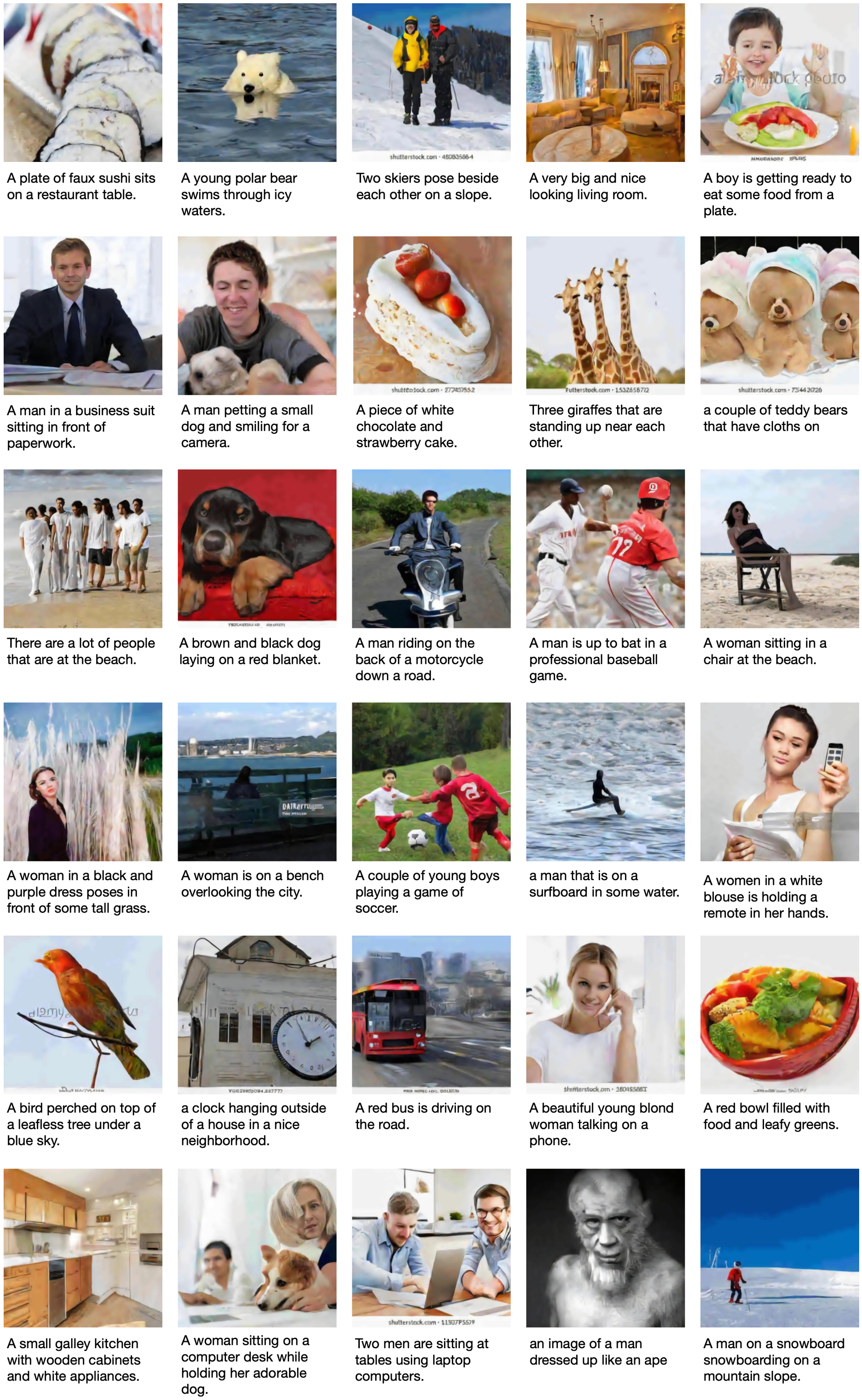


Figure 14: More generated images for COCO captions (after super-resolution).