CommonCanvas: Open Diffusion Models Trained on Creative-Commons Images

Supplementary Material

A. Details on Data Scarcity Analysis

A.1. Hypothesis: Diffusion models are too small

A back-of-the-envelope calculation provides some insight on why this is the case. Consider a training dataset consisting of N images with resolution $H \times W$ and c channels. To completely memorize the training data, the model must be capable of storing $c \times H \times W \times N$ numbers. Given a number of trainable parameters N_p, it is natural to assume that on average each parameter is capable of storing roughly enough information to reconstruct a single number from the training dataset. Under this assumption, complete memorization is only possible if the size of the training dataset is at or below a critical size N_c ($N \leq N_c$) with N_c given by $N_c = \frac{N_p}{cHW}$. Note that this critical size assumes the data cannot be further compressed, which is obviously not the case for natural images. However, SD2 and SDXL are latent diffusion models, which first use a pretrained encoder to compress images by a factor of 8 in both H and W, and so when we train LDMS like SD2 and SDXL, we are training on data that has been significantly compressed already.

In our experiments, $c = 4$ and $H = W = 32$, corresponding to 256×256 resolution RGB images in the SD2 and SDXL latent space. The SD2 UNet has $N_p = 866 \times 10^6$ trainable parameters, and SDXL's UNet has $N_p = 2567 \times 10^6$. So we calculate $N_c \approx 0.2 \times 10^6$ for SD2 and $N_c \approx 0.6 \times 10^6$ for CommonCanvas-Large; both of these numbers are several orders of magnitude below the size of our YFCC derived datasets, and so even with significant additional data compression we expect that our CommonCatalog datasets should be sufficient to train both SD2 and SDXL. Additionally, this argument predicts that we should only begin to see significant overfitting in these models for datasets of size $N \sim 10^6$. These estimates are resolution dependent, and as image resolution increases we expect that N_c will decrease as more information is provided per image.

A.2. Increasing model capacity

We also train a variant of SD2 with more trainable parameters, taking the UNet from SDXL. We refer to this model as CommonCanvas-LNC. We adapt the SDXL UNet architecture to SD2 by changing the cross-attention dimensionality to match that of the SD2 text encoder hidden state dimensionality (1024 for SD2 vs. 2048 for SDXL). SDXL also retrains the VAE component in their model, and we use this improved performance VAE as well. Except for these changes, the architecture is identical to that of SD2.

B. Training Dataset Details

B.1. LAION-2B

The fact that LAION is not a stable benchmark can lead to multiple reproducability and security issues. Data poisoning attacks would be difficult to detect at the scale of 2 billion parameters. While this could be mitigated by using hash values of the images, then any time the a site decide to re-encode the image, those images would now need to be excluded from the dataset. Furthermore, targeted data poisoning attacks for diffusion models are no longer just academic conjecture. Last year after the release of Stable Diffusion, a protest was launched on ArtStation that had uses upload images that said "NoAI" to taint future training data for generative models after artists felt as though their work had been unfairly used to train the models. With the high degree of link rot, targeted attacks are fairly easy. Furthermore, reproduction of the experiments becomes virtually impossible. This means any benchmarks that use copies of LAION as ground truth are are likely using differing subsets of the full dataset.

B.1.1 Sourcing Creative-Commons images

Table 1. CC licenses in YFCC100M. ND means derivative works are not licensed or the license doesn't allow the user to create derivative works. NC means images cannot be used in commercial contexts. CommonCatalog-C only contains data from the bottom two (yellow) rows, reflecting images licensed for commercial contexts (i.e., roughly 25 million images). CommonCatalog-NC contains CommonCatalog-C, and additionally includes the middle two (blue) rows, reflecting images licensed for non-commercial purposes. We do not include the roughly 30 million images in the top two (pink) rows in CommonCatalog, as they are non-derivative licenses. We do not train on these images. We do, however, produce BLIP-2 captions for them and release those captions as an evaluation set.

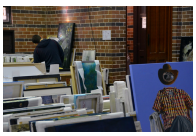
CC License	# Images	% Captioned
CC-BY-NC-ND-2.0	25,790,117	33.52%
CC-BY-ND-2.0	4,827,970	30.23%
CC-BY-NC-2.0	12,468,229	31.39%
CC-BY-NC-SA-2.0	28,314,685	31.57%
CC-BY-SA 2.0	9,270,079	34.05%
CC-BY 2.0	16,962,338	28.96%

B.1.2 Release and documentation

C. YFCC Example Images

Table 2. Randomly sampled images from the YFCC [68] training set. Our synthetic BLIP2 captions are also provided below.

a person riding a bike on a dirt road



a paintings on the wall

an orange and blue race car driving on a track

Model Architecture

We follow the model architecture and training recipe of Stable Diffusion 2 as closely as we can to best reproduce the model for CC-Small. The model has an identical number of params and structure as the original model. In fact, we can even load SD2's model weights into our framework due to the identical architecture and naming scheme. We are able to achieve virtually identical performance with SD2 in a much shorter training time with less data. We use the same VAE, tokenizers, and UNet archicture as SD2 except for reducing the precision of the normalization layers.

Our CC-Large model takes SD2's model and replaces the UNet with the SDXL architecture [49]. Like CC-Small, we also replace the normalization layers with their low-precision version. The replacement of all the normalization layers is handled automatically by MosaicML's Composer library [45]. We perform all dataloading through MosaicML's streaming library [46].

D. Details on Efficiency Optimizations

In this section we provide additional details on the optimizations we implemented to achieve SD2 training speedups. We also report the approximate cost of training our implementation of SD2 on various hardware configurations in Table 5.

Flash Attention. Cross attention operations are a very expensive part of training that occurs in dozens of layers in diffusion model UNets [53]. Flash Attention is an efficient implementation that is optimized to work well with reduced precision and GPU hardware [13], which was implemented using the XFormers library [36], allowing us to save compute and memory usage.

Precomputing latents. Each forward pass of SD2 requires computing a latent representation of the input image, as well as transforming the caption into a text embedding. Instead of computing the latents for each example during training, we can precompute latents for the entire dataset, amortizing the cost. Doing so speeds up training of the model, especially at lower resolutions, in exchange for a one-time fixed cost of precomputing all the latents over 1 epoch.

Reduced-precision GroupNorm and LayerNorm. Most layers in SD2 are implemented in float16 precision, but GroupNorm and LayerNorm are implemented in float32, in part because it was assumed to be necessary for training stability. The resulting, frequent upcasting causes a major bottleneck in training speed. Recent work shows that it is safe to implement LayerNorm using float16 precision [50], and we found the same to be true of GroupNorm. We thus cast all GroupNorm and LayerNorm operators to float16 and are able to further reduce total memory consumption and accelerate training.

Fully-Sharded Data Parallelism (FSDP). FSDP is a variant of data-parallel training that shards the models parameters, gradients and optimizer state across multiple devices. When training data batches do not fit into memory, we do several forward and backward passes on smaller microbatches, followed by a single gradient update. At GPU scale, there may only be a single microbatch, so the time for the gradient update can become a significant bottleneck. In standard data distributed training, each GPU communicates all its gradients to every other GPU, and then each GPU updates its local copy of the model. Instead, we use a different paradigm inspired by [74] where each GPU only gets the gradients and updates the weights for a small part of the model before sending the updated weights for that part of the model to all of the other GPUs. By dividing the update step across all the GPUs, we can ensure that the amount of work per GPU decreases as we increase the number of GPUs, helping us achieve linear scaling. To tackle this problem, we use PyTorch's experimental support for Fully Sharded Data Parallelism (FSDP), specifically, FSDP's SHARD_GRAD_OP mode.

Scheduled Exponential Moving Average (EMA). SD2 uses EMA, which maintains an exponential moving average of the weights at every gradient update for the entire training period. This can be slow due to the memory operations required to read and write all the weights at every step. Since the old weights are decayed by a factor of 0.9999 at every batch, the early iterations of training only contribute minimally to the final average. We decide to only apply EMA for the final 50K steps (about 3.5% of the training period), and are able to avoid adding overhead and still achieve a nearly equivalent EMA model.

E. Telephoning

We dub our solution for handling the lack of captions in CC images as *telephoning*, a type of transfer learning (Figure 3). Telephoning assumes the existence of a large labeled dataset $\mathcal{D}_1 = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, consisting of pairs

Table 3. Top 10 highest frequency captions in the YFCC dataset. The most common captions are not user generated and are not very descriptive of the corresponding image.

YFCC Original Caption	**Count**
OLYMPUS+DIGITAL+CAMERA	184889
SONY+DSC	123128
Exif_JPEG_PICTURE	104480
Barclays+Center+Arena%0AAtlantic+Yards%0A6th+and+Atlantic+A	68832
Olympus+digital+camera	54805
Effortlessly+uploaded+by Eye-Fi	48388
.	43227
-+Camera+phone+upload+powered+by ShoZu	38856
Sony+dsc	32709
Photo+by @Kmeron —Facebook page is this way—	23754

Table 4. Number of usable captions from OpenAI's YFCC14M dataset [51]. This table is actually a subset from 1 for which either the user description or image title were deemed usable. These figures provide an estimate on how many images in each category are actually potentially usable as captions.

License Name	count
CC-BY 2.0	2448002
CC-BY-ND 2.0	682273
CC-BY-NC 2.0	1925854
CC-BY-NC-ND 2.0	4058817
CC-BY-NC-SA 2.0	4146113
CC-BY-SA 2.0	1568336

of high-dimensional $x^{(i)}$ (e.g., images, audio) that map to a compact, structured label $y^{(i)}$ (e.g., caption, audio transcript). Telephoning trains a forward model $q(y|x)$ on $\mathcal{D}_1$ to learn the mapping of y given x via maximum likelihood learning $\max_{q\in\mathcal{Q}} \sum_{i=1}^{n} \log q(y^{(i)}|x^{(i)})$. It then uses q as training signal for a reverse model $p(x|y)$ trained on a *separate* dataset $\mathcal{D}_2 = \{x^{(i)}\}_{i=1}^{m}$ by maximizing $\sum_{i=1}^{m} \mathbb{E}_{y\sim q(y|x^{(i)})}[\log p(x^{(i)}|y^{(i)})]$, the likelihood of the data $\mathcal{D}_2$ and the predicted label y under q. This forms a type of knowledge transfer from the forward labeling task defined by $\mathcal{D}_1$ to the reverse task of inverting x from y on a separate $\mathcal{D}_2$.

While telephoning can be viewed as a type of synthetic labeling, it becomes particularly interesting when x is a type of protected modality (e.g., a copyrighted image), while y is a compact representation of x that does not encode sensitive aspects of y (e.g., a generic caption). Effectively, telephoning performs a type of "lossy compression" or "distillation" from a high-dimensional or information-rich x (e.g., an image of Snoopy) to a low-dimensional or information-poor y that loses the sensitive content in x (e.g., the visual characteristics of Snoopy). Because this compression step is "lossy", a reconstruction x' of x from $p(x|y)$ via y often does not remotely resemble the original input, just like in a game of telephone [43]. We derive the term telephoning from the above intuition, and employ it as useful shorthand to denote instances of transfer learning that solve data-scarcity problems in multimodal generative modeling.

Telephoning for text-to-image modeling. In this work, we apply telephoning to the image and text domains, where CC images are the high-dimensional inputs x, and we use a pre-trained BLIP-2 model [39] for "lossy compression" to short-text captions y (Figure 3a). Together, these CC-image-caption pairs comprise the CommonCatalog dataset, which we use to train our CommonCanvas T2I models (Figure 3b). Even though BLIP-2 was pre-trained on LAION-400M [58], CommonCatalog and CommonCanvas never have direct access to LAION-400M or, importantly, anything that is similar to the images that BLIP-2 was trained on. Instead, we only have access to the mapping in the model, which, given an image input, produces lossy output text that inherently does not literally resemble its image counterpart (Figure 3c).[2]

[2]We draw on the example of Snoopy from [55]. Figure 3's Snoopy is CC-licensed [60].

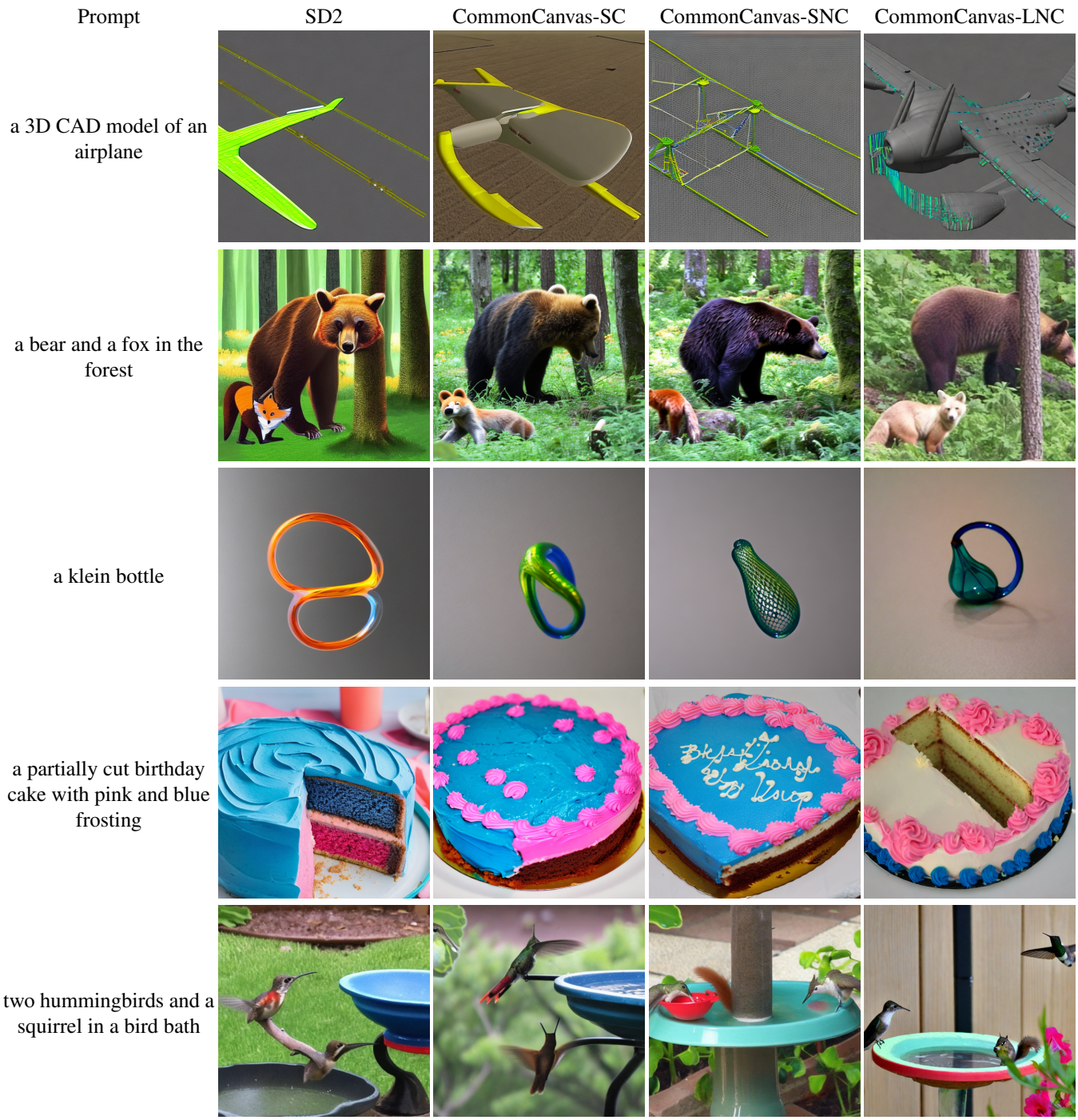


Figure 13. Additional qualitative examples comparing SD2 to our model trained on the commerical split (CommonCanvas-SC), non-commerical split (CommonCanvas-SNC), and the larger UNet model trained on the non-commercial (CommonCanvas-LNC).

Figure 14. Additional qualitative examples of our CommonCanvas models.

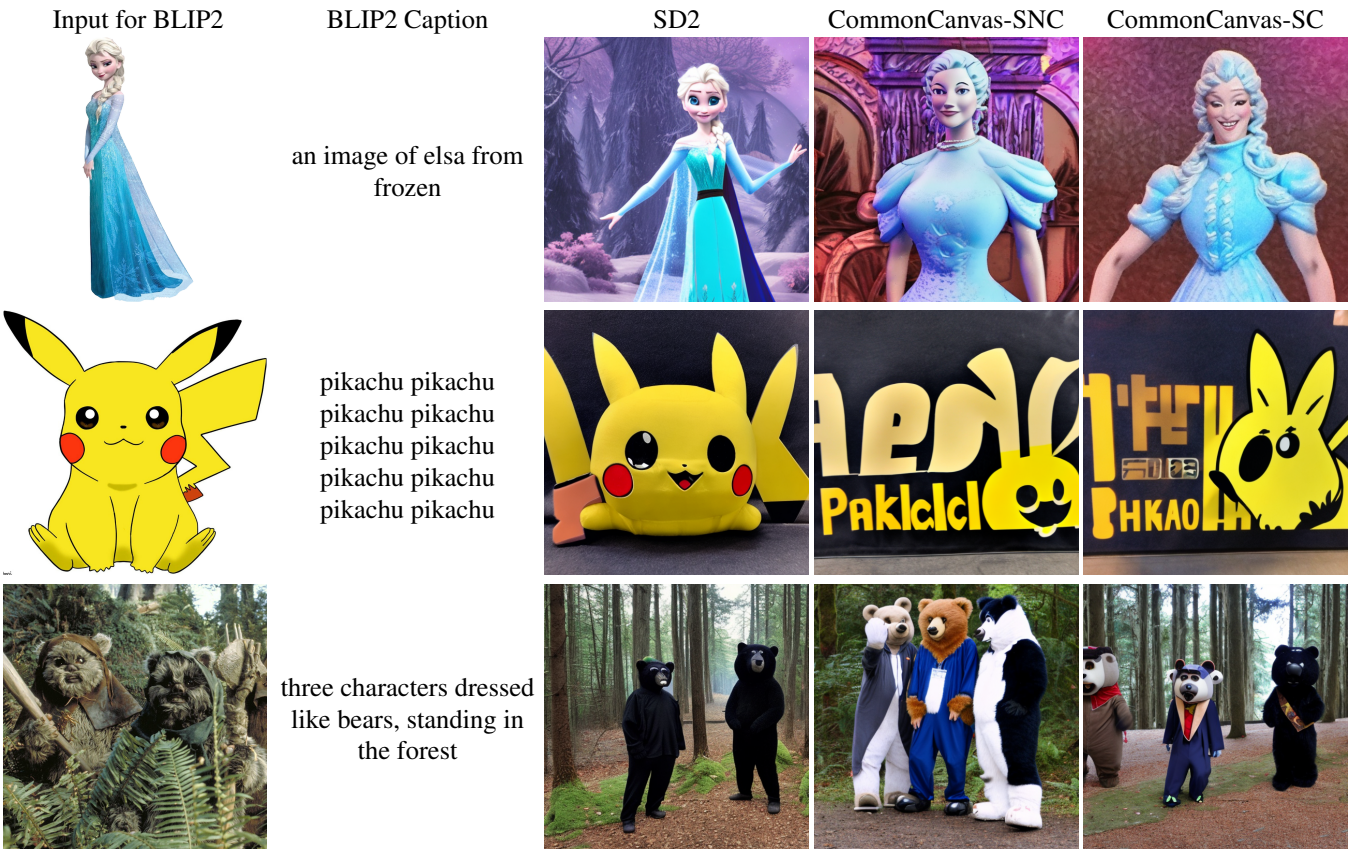


Figure 15. Additional qualitative examples comparing our CommonCanvas models to SD2, given synthetic BLIP2 captions as prompts. While not perfect, our models are better at avoiding generating potentially problematic data.

Table 5. Performance (throughput) and approximate cost of training SD2 UNet with our optimizations. Depending on the number of GPUs used, the cost to train the same models without these optimizations range from $90,000-$140,000

Number of A100s	256x256 (img/s)	512x512 (img/s)	512x512 with EMA (img/s)	Days to Train	Cost ($)
8	1100	290	290	101.04	$38,800.00
16	2180	585	580	50.29	$38,630.00
32	4080	1195	1160	25.01	$38,420.00
64	8530	2340	2220	12.63	$38,800.00
128	11600	4590	3927	6.79	$41,710.00

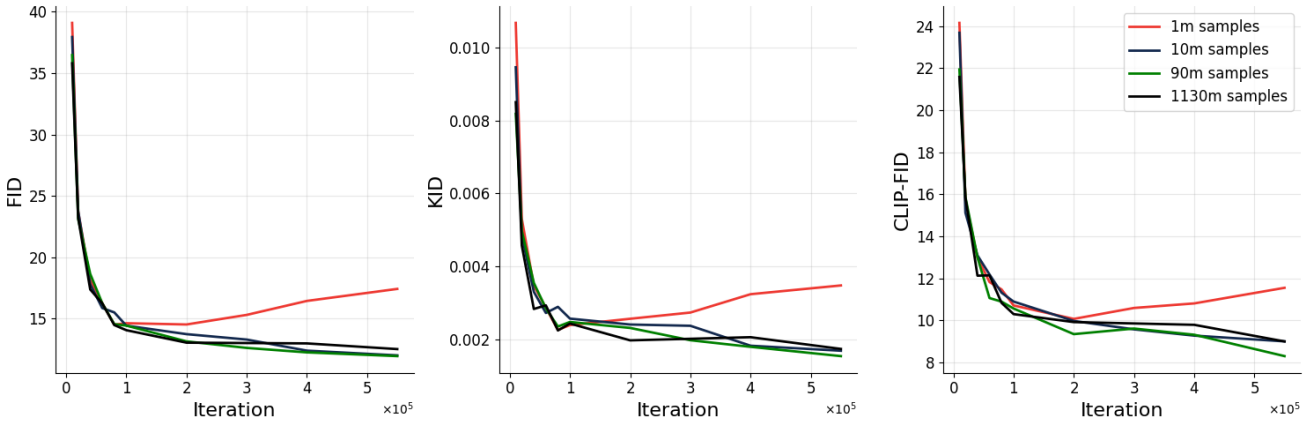


Figure 16. MS COCO metrics over training duration for various dataset sizes. We investigate how reducing the size of the training dataset affects training dynamics, and find that performance is largely unchanged until dropping below 10 million samples. We show that the FID of the eval set remains stable as training progresses. However, reducing the number of samples in our training dataset to 1 million leads to divergence. This finding suggests that only 10 million to 1 million synthetic image caption pairs are needed for good performance on MS COCO.