

UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

---

# ControlNet Scaling Laws

---

*Author:*  
Diana KAPATSYN

*Supervisors:*  
Jack LANGERMAN,  
Dmytro MISHKIN

*A thesis submitted in fulfillment of the requirements  
for the degree of Master of Science*

*in the*

Department of Computer Sciences  
Faculty of Applied Sciences



Lviv 2024

## Declaration of Authorship

I, Diana KAPATSYN, declare that this thesis titled, "ControlNet Scaling Laws" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

**ControlNet Scaling Laws**

by Diana KAPATSYN

## *Abstract*

Effectively training large-scale deep learning models is costly, and requires careful planning and resource allocation. One strategy involves fitting simple parametric functions, such as logarithmic functions, on smaller-scale experiments and extrapolating them to predict model performance and associated costs. These empirical "scaling laws" are then used to predict the required resources for achieving a given level of performance. This approach is widely used for Large Language Models but is insufficiently investigated for computer vision generative models. Today, diffusion models dominate image generation and ControlNet is one of the most popular ways to customize and control them.

This work makes three contributions. First, we have estimated the scaling laws for ControlNet quality depending on the dataset size. Second, we have shown that task-specific metrics, such as edge detection metrics for Canny edges are more suitable for predicting image quality compared to the ControlNet training and validation loss itself. Finally, we present a practical recommendations for dataset size for ControlNet training.

The code and data are available on GitHub<sup>1</sup> and HuggingFace<sup>2</sup> respectively.

---

<sup>1</sup><https://github.com/Diana3101/ControlNetScalingLaws>

<sup>2</sup><https://huggingface.co/scaling-laws-diff-exp>

## *Acknowledgements*

I am thankful to the Armed Forces of Ukraine for the defence and possibility of writing this master's thesis in Independent Ukraine.

I am grateful to my awesome supervisors Jack Langerman and Dmytro Mishkin for the guidance and support throughout the whole project. Thanks to Dmytro Mishkin for providing a lot of useful advices, weekly meetings and active engagement in the all processes related to this master's thesis. Thanks to Jack Langerman for the all meaningful proposed ideas and making possible an efficient download of the large dataset in short period of time.

I am extremely thankful to **HOVER Inc.** for providing computational resources, that makes this project possible.

I want to thank Ukrainian Catholic University, Faculty of Applied Sciences, and Ruslan Partsey, Data Science Academic Program Director for remarkable management during the educational process.

I am grateful to my boyfriend for extraordinary belief in me and his continuous care along the way.

Finally, I want to note that tools like **ChatGPT** and **Grammarly** were used only for reviewing my own text correctness and fixing grammatical, stylish, or vocabulary errors. Thanks to developers of them.



# Contents

<b>Declaration of Authorship</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>2</b>
2.1 Scaling laws for conditional generative models. Key research gap . . .	2
2.2 Diffusion models . . . . .	3
2.3 Stable Diffusion . . . . .	4
2.4 Personalization of the text-to-image diffusion models . . . . .	5
2.5 Scaling laws . . . . .	9
2.6 Metrics for scaling laws . . . . .	10
<b>3 Methodology</b>	<b>12</b>
3.1 Research goals . . . . .	12
3.2 Research hypothesis . . . . .	12
3.3 Experiment Setting . . . . .	12
3.4 Evaluation . . . . .	13
<b>4 Data</b>	<b>19</b>
4.1 Small-scale fill50k dataset . . . . .	19
4.2 Large-scale dataset . . . . .	20
<b>5 Experiments</b>	<b>26</b>
5.1 Small-scale dataset . . . . .	26
5.2 Large-scale dataset . . . . .	28
<b>6 Conclusions</b>	<b>36</b>
6.1 Discussion . . . . .	36
6.2 Future Work . . . . .	37
<b>Bibliography</b>	<b>38</b>

# List of Figures

2.1	The impact of dataset size on the generation quality of ControlNet. Figure from (Zhang, Rao, and Agrawala, 2023).	2
2.2	Two processes in diffusion model training: forward diffusion process, where noise is added to the signal, and reverse diffusion process, where the noise is estimated and subtracted.	3
2.3	Reverse diffusion process of Stable Diffusion: denoising. $Z_T$ is a noisy latent vector at the denoising step $T$ . The $Z_{T-1}$ is a latent vector after one denoising step. $Z$ is the last denoised latent vector. $D$ is a fixed VAE decoder, which projects from latent to image space. $\tau_\theta$ is a domain-specific encoder for text prompts.	4
2.4	Personalized generation results by Textual Inversion. The figure is taken from (Gal et al., 2022).	5
2.5	Conditional image generation with ControlNet.	6
2.6	Reverse diffusion process of ControlNet over Stable Diffusion: denoising. $Z_T$ is a noisy latent vector at the denoising step $T$ . The $Z_{T-1}$ is a latent vector after one denoising step. $Z$ is the last denoised latent vector. $D$ is a fixed VAE decoder, which projects from latent to image space. $\tau_\theta$ is a domain-specific encoder for text prompts. $E$ is an encoder for mapping control signal to the latent space. The encoder is trained jointly with the ControlNet. $c_f$ is encoded control signal. Detailed architecture of the ControlNet is presented in the Figure 2.7.	7
2.7	ControlNet over Stable Diffusion architecture	8
2.8	The scaling laws proposed by Kaplan et al., 2020. The figure is from (Kaplan et al., 2020).	9
3.1	"Raw" (top) and smoothed (bottom) ControlNet training loss.	13
3.2	Hypothesis 1: the conditional image generation is of good quality and aligns with the control signal when the estimated contours/segmentation/etc on the generated image are the same as the control signal itself. Such alignment can be estimated with task-specific metrics.	14
3.3	Edge detection metrics for different target and prediction pairs: top – unrelated images, center – similar, bottom – identical.	15
3.4	<b>Blurred</b> edge detection metrics for different target and prediction pairs. Top – unrelated images, center – similar, bottom – identical	16
3.5	Depth map metrics for various target and prediction pairs: top – unrelated, center – similar, bottom – identical.	18
4.1	Training examples of the fill50k dataset	19
4.2	Training sample from small-scale experiment on fill50k dataset	19
4.3	Distribution of initial 160M dataset after images downloading and two-iteration filtering by DETR labels	21
4.4	The size distribution of images in the 72M dataset	22
4.5	Training images examples with respective depth maps	22

4.6	Training images examples with respective Canny edges . . . . .	23
4.7	The size distribution of images in the 1M dataset - the largest subset, we have finished training on, at the time of submission . . . . .	23
4.8	Entire test set of 1000 images . . . . .	25
5.1	Correlation between AP metric and generated images correspondence to the target image. The metric also shows a so-called sudden convergence phenomenon. . . . .	26
5.2	Correlation between metrics, such as AP and training loss, and different dataset sizes of the fill50k dataset. . . . .	27
5.3	Validation-generated images from experiments with varying dataset sizes, predicted with prompt and in a 'no prompt' mode. . . . .	27
5.4	Correlation between edge detection metrics and generated images correspondence to the target image during the training process using 1M images with Canny edges as control signals. The metrics clearly show a "sudden convergence: phenomenon. . . . .	28
5.5	Correlation between RMSE Log metric and generated images correspondence to the target image during the training process using 1M images with depth maps as control signals. The metrics clearly represent a sudden convergence phenomenon. . . . .	29
5.6	Full range of depth metrics utilized for establishing correspondence in depth maps during training on a 1M dataset. Their plots are highly similar, while the scale of the metrics is different. . . . .	30
5.7	Comparison of task-specific metrics and validation losses. Task-specific metrics look more stable, while the validation loss exhibits random spikes. . . . .	31
5.8	Significant decrease in ControlNet training loss might be an indication of the overfitting. This both can be observed in validation loss graph (going up), and the visual inspection of the generated images. The model is trained on 1k dataset . . . . .	32
5.9	Canny edge condition: AP metric on the test set for ControlNet trained on various dataset sizes throughout the training process . . . . .	32
5.10	Depth condition: RMSE Log metric for ControlNet trained on various dataset sizes throughout the training process. . . . .	33
5.11	Canny edge condition: AP metric and validation loss depending on the dataset sizes. . . . .	33
5.12	Depth condition: metrics depending on the dataset sizes for the ControlNet. Top: RMSE Log and validation loss, bottom: a1 (threshold accuracy). . . . .	34
5.13	Canny edge condition: scaling laws for ControlNet based on AP metric . . . . .	34
5.14	Depth condition: scaling laws for ControlNet, with RMSE Log metric (top) and a1 (bottom)) metrics . . . . .	35

# List of Tables

4.1	Training images examples with captions . . . . .	24
4.2	Test images examples with BLIP2 (Li et al., 2023a) generated captions .	24

# List of Abbreviations

<b>LLM</b>	<b>Large Language Models</b>
<b>GAN</b>	<b>Generative Adversarial Network</b>
<b>LDM</b>	<b>Latent Diffusion Model</b>
<b>SD</b>	<b>Stable Diffusion</b>
<b>T2I</b>	<b>Text to Image model</b>
<b>LoRA</b>	<b>Low-Rank Adaptation</b>
<b>ResNet</b>	<b>Residual neural Network</b>
<b>ViT</b>	<b>Vision Transformer</b>
<b>PSNR</b>	<b>Peak Signal-to-Noise Ratio</b>
<b>SSIM</b>	<b>Structural Similarity Index Measure</b>
<b>FID</b>	<b>Fréchet Inception Distance</b>
<b>ODS</b>	<b>Optimal Dataset Scale</b>
<b>AP</b>	<b>Average Precision</b>
<b>DETR</b>	<b>DEtection TRansformer</b>
<b>ARE</b>	<b>Absolute Relative Error</b>
<b>SRE</b>	<b>Squared Relative Error</b>
<b>RMSE</b>	<b>Root Mean Squared Error</b>
<b>RMSELog</b>	<b>Root Mean Squared Error Log scale</b>
<b>ABSLog<sub>10</sub></b>	<b>Absolute Log<sub>10</sub></b>
<b>SILog</b>	<b>Scale Invariant Log Error</b>
<b>A1 (<math>\delta_1</math>)</b>	<b>Threshold Accuracy 1</b>
<b>A2 (<math>\delta_2</math>)</b>	<b>Threshold Accuracy 2</b>
<b>A3 (<math>\delta_3</math>)</b>	<b>Threshold Accuracy 3</b>



## Chapter 1

# Introduction

Training large-scale deep learning models (and gathering large-scale datasets) is a costly endeavor (Shen et al., 2023). For example, Stable diffusion version 1.5 (Rom-bach et al., 2022) was trained on 5 billion images, with a help of approximately 250 A100 GPUs. The training took around 150k GPU-hours (Mostaque, 2022). Doing this on Google Cloud Platforms would cost approximately 700 000 dollars.

Before starting a project, cost and resource estimation must be considered. Researchers and practitioners have to determine how much data is necessary to reach the desired performance level and what level of model performance can be achieved within the allocated budget for data labeling and compute. For example, to label 60 thousand images for image classification task on Amazon costs 8k dollars, whereas tasks such as object detection or image segmentation require even higher labeling expenses (Lee, 2023). The ability to estimate required resources (compute, dataset size) to achieve a given level of performance helps minimize the risks in such a process. This implies that researchers and practitioners would have a performance and cost estimate in the early stages of a project, rather than conducting experiments blindly.

One set of approaches to performing this estimation which have recently come to the fore are collectively referred to as "scaling laws" (Hestness and al, 2017). Application of scaling laws is common in the training of Large Language Models (LLMs) like GPT-3/4 (Brown and al., 2020, OpenAI, 2023). Scaling laws are also studied in computer vision, focusing mainly on discriminative models (Zhai et al., 2022, Cherti et al., 2023).

There is less research on scaling laws for generative models in computer vision, such as Generative adversarial networks (GANs) (Brock, Donahue, and Simonyan, 2018), diffusion models (Jiaming Song, 2023), etc. To the best of our knowledge, the exploration of scaling laws for generative, and, in particular, diffusion models is limited and concentrated mostly on the size of the model (Nichol and Dhariwal, 2021, Peebles and Xie, 2023). There is a growing interest in text-to-image generative models and approaches to personalize them, both in research (Jiaming Song, 2023, Ruiz et al., 2023, Li et al., 2023b, Zhang, Rao, and Agrawala, 2023, Gal et al., 2022, Mou et al., 2023), and industry (*Midjourney 2022, HOVER Inc. 2023, getimg.ai 2023, Hotpot.ai 2023*). One of the most popular ways to control the geometric structure of a generated image is the recently proposed ControlNet (Zhang, Rao, and Agrawala, 2023), where the generated image is conditioned on the input image data in the form of a drawing, segmentation, depth map, etc.

## Chapter 2

# Related Work

The following related works provide an overview into both the models we are exploring and the scaling laws within the domains they were investigated. First, we overview the domain of the model we intend to develop scaling laws for, and then move on to the scaling laws. We begin by discussing diffusion models broadly, then narrow down to the text-to-image diffusion models, specifically Stable Diffusion, which we will use in our approach. Additionally, we explore methods for personalizing text-to-image diffusion models, one of which is ControlNet.

### 2.1 Scaling laws for conditional generative models. Key research gap

The conditional text-to-image modeling with diffusion is a young area of research (ControlNet paper appeared on arXiv in February 2023 and was officially published at ICCV2023, October 2023) and we have failed to find papers about this specific topic. The authors of ControlNet (Zhang, Rao, and Agrawala, 2023) presented only a qualitative evaluation of the model scaling without quantitative results (see Fig. 2.1); we aim to fill this gap. We are focusing on the dataset size aspect of the scaling laws, as it was repeatedly shown that the simpler model, trained on bigger and better dataset outperforms larger or more complex model, trained on the smaller dataset (Sun et al., 2017; Kolesnikov et al., 2020; Schuhmann et al., 2022). However, the data aspect is often overlooked, as most of publications focus on the compute and architectures, rather than data.

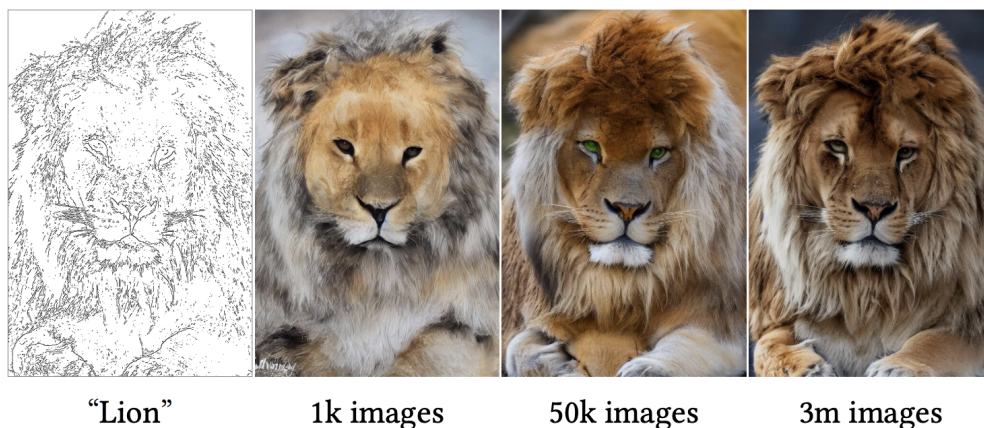


FIGURE 2.1: The impact of dataset size on the generation quality of ControlNet. Figure from (Zhang, Rao, and Agrawala, 2023).



## 2.2 Diffusion models

Generative models give a lot of possibilities by creating synthetic data based on structures and patterns from original data. It is widely used both for improving the efficiency and accuracy of existing AI models, and for creating new original content for advertising and entertainment. Diffusion models (Sohl-Dickstein et al., 2015) are deep generative models, which are used in a variety of domains (Ho, Jain, and Abbeel, 2020, Jiaming Song, 2023).

At inference time, they iteratively transform noise randomly sampled from a simple distribution into samples from complex data distributions, which resemble real data. The training consists of two main processes: forward and reverse. During the forward process, noise is gradually added to the original data until it becomes a simple distribution such as Gaussian or Uniform. During the reverse process, at each step the model predicts either the noise that was added to the image at this step during the forward process, or directly the denoised version of the image (signal). The process is visualized in the Figure 2.2.

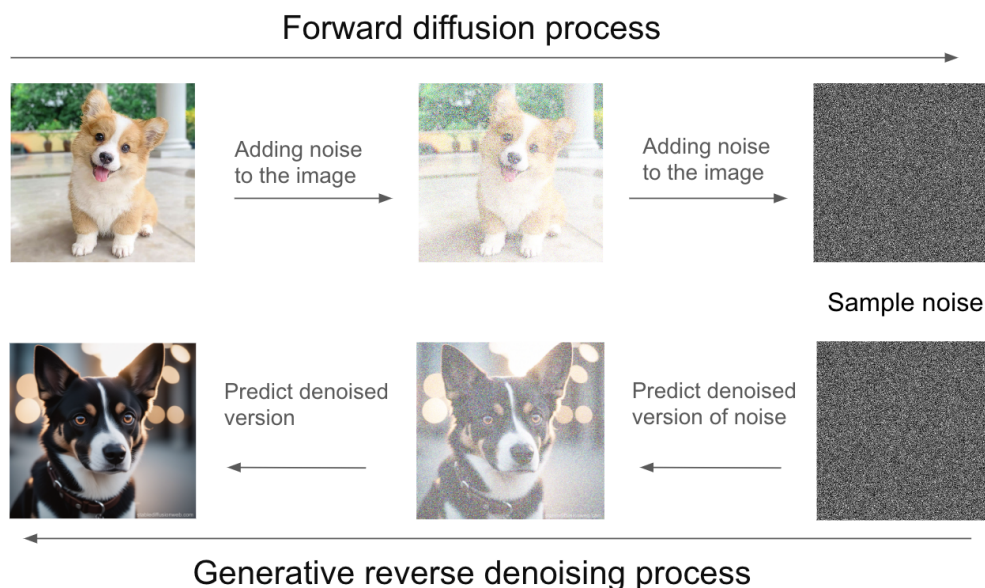


FIGURE 2.2: Two processes in diffusion model training: forward diffusion process, where noise is added to the signal, and reverse diffusion process, where the noise is estimated and subtracted.

The drawback of working in pixel space is two fold. First, pixel space is enormous (e.g. a relatively small RGB image of resolution  $512 \times 512$  pixels has 786432 parameters) driving up computational cost. Additionally, most bits in an image are allocated to representing details which are imperceptible to humans. To overcome these limitations, latent diffusion models (LDM) (Rombach et al., 2022) were proposed. Rombach et al. use pre-trained autoencoders to map images from pixel space to latent space, which is much smaller, e.g.  $4 \times 64 \times 64 = 16384$  for Stable Diffusion ( $48\times$  more compact). It increases the efficiency of both the training and inference of diffusion models. While we focus on images, LDMs can be applied across a wide range of data modalities (Jiaming Song, 2023).

Unconditional diffusion models generate images that resemble the training data distribution (Graham et al., 2023). Conditional diffusion models allow more control of the output of the diffusion model by adding conditions like text, image, or scalar

to the generation process (Saharia et al., 2021, Ramesh et al., 2022, Chitwan and al, 2022, Ho, 2022, Dhariwal and Nichol, 2021, Nie, Vahdat, and Anandkumar, 2021).

Some of the popular text-to-image (T2I) models are DALL-E2 (OpenAI, 2022), Imagen (Chitwan and al, 2022), Midjourney (Midjourney 2022), and Stable Diffusion (Rombach et al., 2022).

## 2.3 Stable Diffusion

We are focusing on Stable Diffusion (SD) (Rombach et al., 2022). It is a widely used open-source latent text-to-image diffusion model.

The input image is initially encoded into latent space using a pre-trained encoder. It then undergoes a fixed forward diffusion process. During the generative reverse denoising process, a U-net neural network architecture (Ronneberger, Fischer, and Brox, 2015) is utilized for noise prediction. Additionally, the U-net receives a text prompt along with the input noise. A text is converted to the CLIP embedding (Radford et al., 2021) with a domain-specific encoder  $\tau_\theta$ . But, the encoder can be also adopted for conditions from other domains. Then, the intermediate representation of the text is combined with the intermediate feature maps of the U-Net via a cross-attention mechanism (Vaswani et al., 2017). The U-net receives the noise from step  $Z_T$  and the text embedding, and predicting the noise that should be subtracted from  $Z_T$  to obtain the denoised version  $Z_{T-1}$ . This process repeats iteratively for  $T$  steps. Finally, the pre-trained decoder converts the last denoised representation  $Z$  from latent space to pixel space. The generative reverse process during training is illustrated in Figure 2.3.

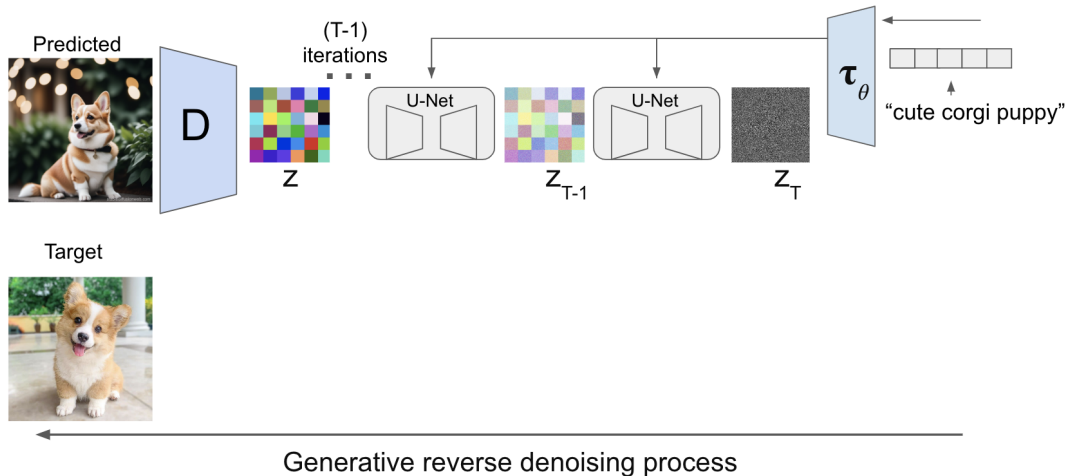


FIGURE 2.3: Reverse diffusion process of Stable Diffusion: denoising.  $Z_T$  is a noisy latent vector at the denoising step  $T$ . The  $Z_{T-1}$  is a latent vector after one denoising step.  $Z$  is the last denoised latent vector.  $D$  is a fixed VAE decoder, which projects from latent to image space.  $\tau_\theta$  is a domain-specific encoder for text prompts.

Controlling the output of a diffusion model is more straightforward when using an additional text prompt, as it is done with SD. For instance, with an unconditional diffusion model, it is difficult to control the specific breed of dog that is generated (see the example image in Figure 2.2). It might generate the target breed or a different one. But, with a text-to-image diffusion model, we can specify the breed, such as by

using the prompt "cute corgi puppy" (see Figure 2.3) and receive the corresponding generated image.

## 2.4 Personalization of the text-to-image diffusion models

One of the problems in real-world applications is the customization of T2I models when the user wants to generate an image based on their reference image but in a different setting and context (see Figure 2.4, Figure 2.5).

**DreamBooth** (Ruiz et al., 2023) allows generating images with the same custom object in different contexts. It fine-tunes a T2I model by using 3-5 images of the personal object and the text prompt that contains the unique identifier for the object and the corresponding object class. Simultaneously it uses a specific objective that maintains the characteristics specific to the class, encouraging the model to generate varied instances related to that class.

**Textual Inversion** (Gal et al., 2022) resembles DreamBooth (Ruiz et al., 2023) in its approach. But, it differs in that it doesn't fine-tune the entire T2I model; rather, it focuses on refining the text embedding component. In this method, a new pseudo-word is incorporated into the text prompt. The model learns a new text embedding associated with this new word, employing a reconstruction objective to represent the images provided by the user accurately. Textual Inversion demonstrates effective performance both for artistic style and personal objects.



FIGURE 2.4: Personalized generation results by Textual Inversion. The figure is taken from ( Gal et al., 2022).

**LoRA** (Low-Rank Adaptation) was first proposed for LLMs (Hu et al., 2021). The study revealed that concentrating solely on the attention layers, fine-tuning quality using LoRA matched that of fine-tuning the entire model, but with the notable advantages of being significantly faster and demanding less computational resources. This approach was adopted for Stable Diffusion (cloneofsimon, 2022).

Another method that fine-tunes solely cross-attention layers and text embedding for new concept is **Custom Diffusion** (Kumari et al., 2023). What sets Custom Diffusion apart is its unique capability to learn multiple concepts concurrently. This is achieved by consolidating the training datasets tailored to each concept and employing specific tokens for each of them. Also, the authors suggested using the regularization images as a preventive measure against overfitting.

**Perfusion** (Tewel et al., 2023) is one more technique for personalizing T2I models. It introduced a unique mechanism known as "Key-Locking" to mitigate the risk of attention overfitting. It associates the cross-attention keys of the new concepts with their subject categories, preventing personalized examples from overpowering other words across the entire attention map. Moreover, the authors created a gated rank-1 method that allows merging various concepts and controlling the impact of an acquired concept during inference.

All methods above do not allow for spatial control over the resulting image. The methods, that will be described below, do allow such control (see Fig. 2.5).



FIGURE 2.5: Conditional image generation with ControlNet.

**T2I Adapter** (Mou et al., 2023) is a compact model linked to the frozen T2I model's U-Net (Ronneberger, Fischer, and Brox, 2015) encoder. It comprises layers that capture the features of the condition image by reducing its resolution. It can use various conditions, similar to GLIGEN (Li et al., 2023b), and unite different adapters trained under varied conditions without requiring retraining.

**GLIGEN** (Li et al., 2023b) is a method where the weights of the diffusion model are frozen, and new trainable layers are incorporated. This approach can be employed with various inputs such as bounding boxes with corresponding entities in the text, semantic maps, canny maps, and more. The authors integrated a new gated self-attention layer between the attention layers of the model at each transformer block to include new conditional input. During the inference step, the model has the flexibility to decide whether to utilize recently trained layers or not. This decision is made to strike a balance between the quality of generation and the model's ability to establish new conditions.

**ControlNet** (Zhang, Rao, and Agrawala, 2023) is a neural network structure to add image-based, spatial conditions to diffusion models. It proves useful when the



user knows the shapes and structures they want in the generated image but desires to observe them in various colors, textures, or environments. It trains specifically for each input modality, like Canny edges, depth maps, human poses, scribbles etc. One also can combine individual ControlNets into a multi-conditioning model during inference. Moreover, ControlNet shows a strong recognition ability as it identifies control signals and produces meaningful images even without prompts (Zhang, Rao, and Agrawala, 2023). The authors chose to conduct experiments with ControlNet over Stable diffusion, while it can be applied to any neural network blocks.

Although sufficient number of methods exist for personalizing T2I diffusion models, we focus our research on those that enable precise spatial control over image generation (see Figure 2.5). Specifically, there are three methods: ControlNet (Zhang, Rao, and Agrawala, 2023), T2I Adapter (Mou et al., 2023), and GLIGEN (Li et al., 2023b). Other described methods allow generating the object from input images in various styles and settings (see Figure 2.4), but they do not provide control over the specific placement, shapes, or structure of the object in the generated images. ControlNet, T2I Adapter, and GLIGEN also share similarities in the concept of locking parameters of the original model and incorporating new trainable layers for fine-tuning T2I models. But, **ControlNet** is much widely used in research and industry, which is evidenced by the number of citations and GitHub stars. As of May 29, 2024, ControlNet has 1555 citations and 28300 stars, compared to T2I Adapter with 395 citations and 3200 stars, and GLIGEN with 275 citations and 1800 stars. ControlNet is the de-facto standard for spatial control over the diffusion-based image generation. That is why we chose to focus our research on scaling laws for ControlNet and not other approaches.

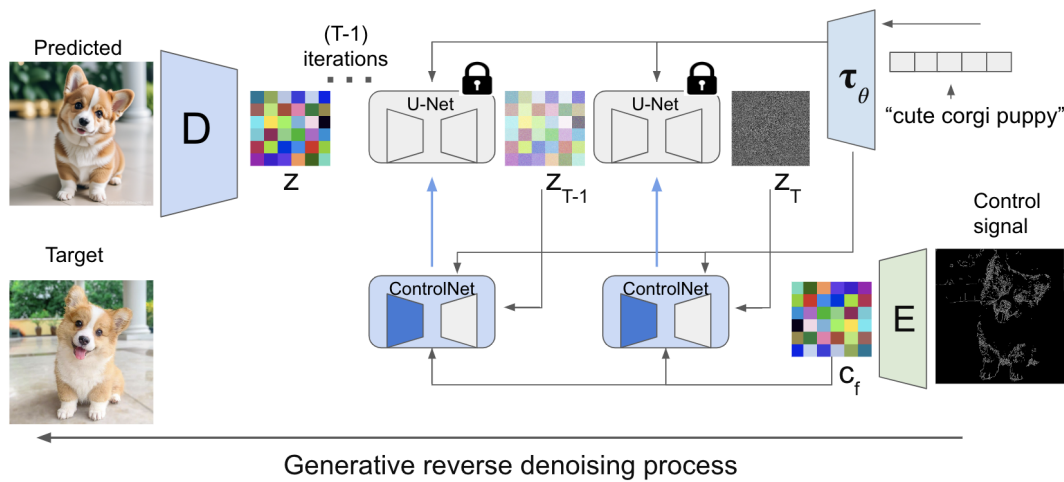


FIGURE 2.6: Reverse diffusion process of ControlNet over Stable Diffusion: denoising.  $Z_T$  is a noisy latent vector at the denoising step  $T$ . The  $Z_{T-1}$  is a latent vector after one denoising step.  $Z$  is the last denoised latent vector.  $D$  is a fixed VAE decoder, which projects from latent to image space.  $\tau_\theta$  is a domain-specific encoder for text prompts.  $E$  is an encoder for mapping control signal to the latent space. The encoder is trained jointly with the ControlNet.  $c_f$  is encoded control signal. Detailed architecture of the ControlNet is presented in the Figure 2.7.

**ControlNet over Stable Diffusion.** In our experiments, we investigated ControlNet over SD. The training pipeline for ControlNet is presented in Figure 2.6. ControlNet is a trainable copy of the U-Net used in SD, with its encoder consisting of the same blocks as the U-Net’s encoder, while the decoder is composed of zero convolution blocks. So, the encoders are visualized with the same colors, whereas the decoders are shown in different colors.

Unlike Stable Diffusion, ControlNet incorporates an additional image condition - Canny edges of the target image - referred to as the Control signal in Figure 2.6. This control signal is encoded from the pixel space to the latent space using a specific encoder, which trains jointly with the ControlNet. Consequently, ControlNet receives noise  $Z_T$ , prompt embedding, and the encoded control signal  $c_f$  to predict the noise that should be subtracted from  $Z_T$  to obtain the denoised version  $Z_{T-1}$ . Similar to SD, this process is repeated for  $T$  iterations, with the final generated denoised representation  $Z$  ultimately being converted back to pixel space by the decoder.

With SD we can specify the breed of the dog, such as predicted image with "cute corgi puppy" in the Figure 2.3. In contrast, ControlNet can generate a "cute corgi puppy" with specific contours defined by the control signal, as shown in the predicted image in Figure 2.6.

The architecture of the ControlNet over Stable Diffusion is presented in the Figure 2.7.

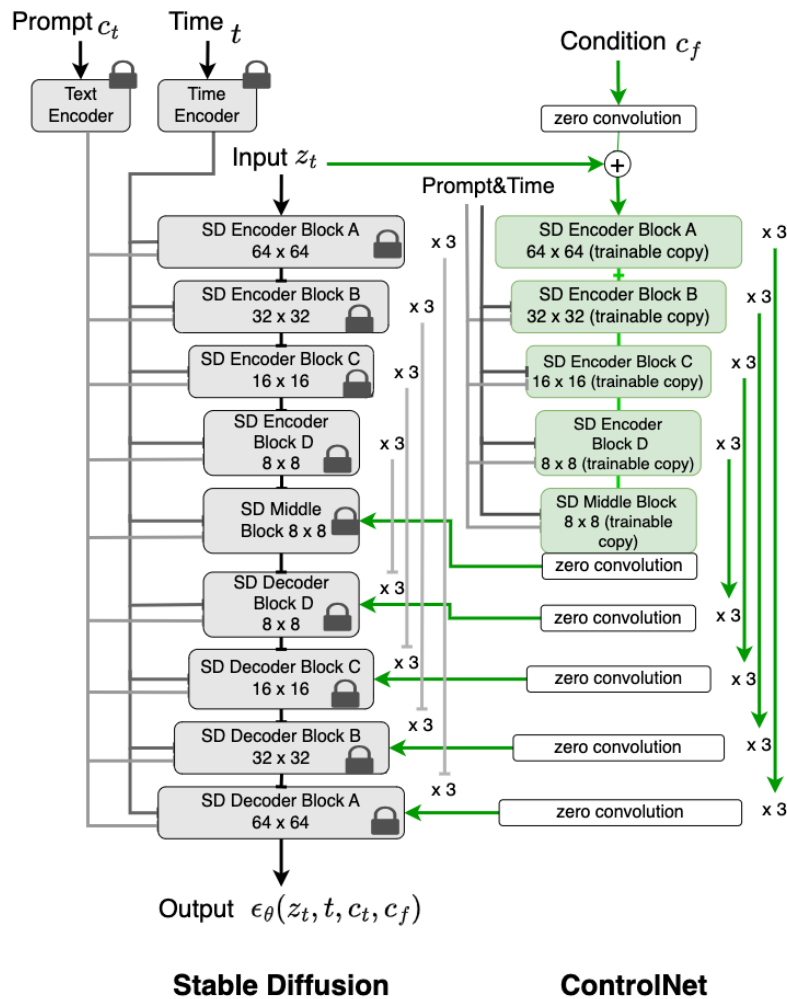


FIGURE 2.7: ControlNet over Stable Diffusion architecture

As we mentioned above, ControlNet has the same U-Net encoder layers as SD, including the middle block, but has zero convolution layers, except decoder blocks. And also an additional zero convolution layer for the middle block. As the input ControlNet takes condition vector  $c_f$  (encoded control signal) and combines it with the noisy latent representation of the input image  $z_t$  through the zero convolution layer. The convolution layers initialized with zeros prevent adding any harmful noise to the pre-trained features of the original model. Then the combined input goes through encoder blocks, where it combines with encoded prompt  $c_t$  and encoded time (step)  $t$  in each block with the cross-attention mechanism (Vaswani et al., 2017), as it is done in the original SD model. After it passes through the encoder middle block, it goes to the first zero convolution layer, which is connected to the SD middle block, and then - to the next zero convolution layers, which are connected to the SD decoder blocks. Finally, it returns noise for subtracting from the input latent vector  $z_t$  in this step.

The ControlNet trains using the learning objective that is presented in Equation 2.1

$$\mathcal{L} = \mathbb{E}_{z_0, t, c_t, c_f, \epsilon \sim N(0,1)} [\|\epsilon - \epsilon_\theta(z_t, t, c_t, c_f)\|_2^2], \quad (2.1)$$

where  $z_0$  is the input (target) image, for which the noise  $\epsilon$  is added  $t$  times and we get a noisy image  $z_t$ .  $c_t$  is a text-condition (prompt),  $c_f$  is a task-specific condition, e.g. Canny edges of the target image  $z_0$ .  $\epsilon_\theta$  - the predicted by neural network noise.

## 2.5 Scaling laws

The term "Scaling laws" describes the functional relations between the characteristics of the deep learning model quality (loss, performance metrics) and hyperparameters, such as dataset size, model size, amount of compute, etc. They can be useful in designing the optimal model architecture. Such experiments started long before the deep learning era (Banko and Brill, 2001), focusing on the popular machine learning models of that time. The empirical research on the impact of deep learning model size on performance gained a focus a few years ago (Hestness and al, 2017, Hestness, Ardalani, and Diamos, 2019, Rosenfeld et al., 2019).

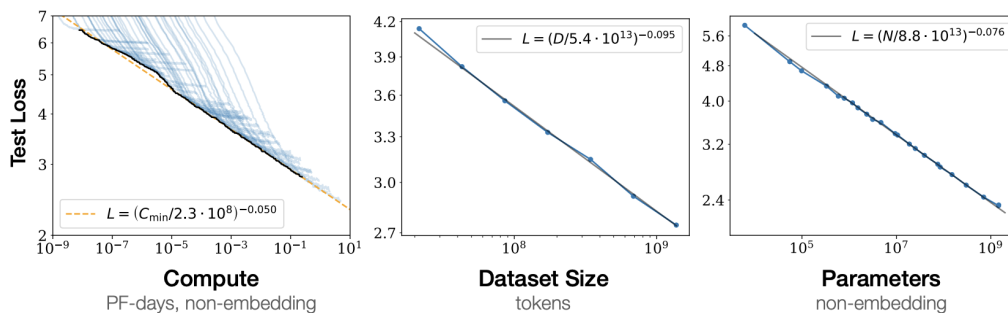


FIGURE 2.8: The scaling laws proposed by Kaplan et al., 2020. The figure is from (Kaplan et al., 2020).

One of the most influential works in scaling laws was released by Kaplan et al., 2020, who found power-law scaling for the Transformer (Vaswani et al., 2017) language models using the cross-entropy loss. The experiments have shown that the loss scales with the quantity of non-embedding parameters, dataset size, and computational resources (see Fig. 2.8) whereas the model's structural parameters such as

depth and width do not have a notable impact. Another essential discovery is that when increasing computational resources, the majority of these resources should be allocated to expanding the size of the model. Moreover, larger models achieve superior performance in less training time. The proposed scaling laws are similar for autoregressive generative modeling (Henighan et al., 2020) and transfer learning (Hernandez et al., 2021).

Hoffmann et al., 2022 focused on identifying the trade-off between the sizes of the language model and the dataset while maintaining a fixed computational budget. They have shown that for optimal performance, both the number of model parameters and the number of tokens should be scaled proportionally. The resulting Chinchilla model outperforms their previous Gopher model (Rae et al., 2021). But, the Chinchilla has four times fewer parameters and was trained on a dataset four times larger, while utilizing the same computational resources.

Sorscher et al., 2022 proposed a concept of data pruning, suggesting that a well-designed small dataset beats larger dataset, which are not designed carefully, which could result in exponential loss scaling with the pruned dataset size. They validated this theory using ResNets (He et al., 2015) trained on ImageNet (Russakovsky et al., 2014), CIFAR (Krizhevsky, 2009), and SVHN (Netzer et al., 2011). The central claim asserts that pruning effectiveness lies in retaining examples that provide the most information, measured by the rate of change of entropy concerning the amount of data.

The scalability of ten language task architectures was investigated by Tay et al., 2022. They note significant variations in the scaling exponent among different architectures, highlighting the vanilla transformer as having the most favorable exponent, even though its performance may not consistently be the best. Moreover, the experiments showed that certain models demonstrate suboptimal scalability and the upstream loss does not consistently serve as an accurate predictor of downstream performance when comparing diverse architectures.

Zhai et al., 2022 developed a saturating (with two additional constants) power law for Vision Transformers (ViT) (Dosovitskiy et al., 2020) scaling.

Also, the scaling laws were researched in other fields such as Audio and Speech Processing (Droppo and Elibol, 2021), Machine Translation (Gordon, Duh, and Kaplan, 2021, Bansal et al., 2022), Recommendation Models (Ardalani et al., 2022), Reinforcement learning (Gao, Schulman, and Hilton, 2023, Neumann and Gros, 2022).

## 2.6 Metrics for scaling laws

**Metrics** commonly used to evaluate discriminative models are also used for the scaling laws of large language models, making experiments straightforward. These metrics can be a cross-entropy loss, accuracy, perplexity (Kaplan et al., 2020, Tay et al., 2022). Unlike the tasks above, image generation is harder to evaluate numerically. Even for the tasks, where the ground truth is available (deblurring, super-resolution, etc), there are many metrics, which are at odds with each other (Vasu, Thekke Madam, and Rajagopalan, 2018).

Such metrics include traditional image quality metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) (Wang et al., 2004), as well as more goal-oriented metrics, like Fréchet inception distance (FID) (Heusel et al., 2017) and Perceptual Loss (Johnson, Alahi, and Fei-Fei, 2016).



---

However, for the diverse image generation model the "ground truth" is not available, as many variants of the generated image are acceptable. This makes the formulation of scaling laws for condition-based image generation models even harder.

## Chapter 3

# Methodology

### 3.1 Research goals

We aim to formulate scaling laws for ControlNet, which will show the relationship between the dataset size and the quality of generated images. In addition, we evaluate whether the scaling laws are task-specific or remain consistent for different control signals (such as contours and depth maps).

### 3.2 Research hypothesis

**Hypothesis 1** *The conditional image generation is of good quality and aligns with the control signal when the estimated contours/segmentation/etc on the generated image are the same as the control signal itself. Such alignment can be estimated with task-specific metrics and off-the-shelf models.*

**Hypothesis 2** *Training the ControlNet on a larger dataset size is expected to result in an improvement in the quality of the generated images.*

**Hypothesis 3** *There exist simple functional relationships between the size of the dataset and the performance metrics.*

**Hypothesis 4** *Scaling laws are consistent across different types of control signals, such as Canny edges and depth maps.*

### 3.3 Experiment Setting

First, we performed experiments on a small-scale dataset, which is described in Section 4.1. This synthetic dataset presents significantly simpler patterns to learn compared to real-world data. Trainings with the small-scale dataset give a sense of how the training is going and help ensure the model is working correctly in significantly less time and with fewer computational resources than training on real data. Such approach gave us the possibility to conduct numerous experiments and explore various hyperparameters within the neural network. Beginning with the baseline model, we iteratively experimented, leading to pre-validation of hypotheses 1, 2. This process enabled us to obtain insightful results even before working with large-scale dataset. Moreover, beyond providing clarity, it also allowed for more precise planning of experiments on a larger dataset.

Datasets of different sizes were created, with each smaller subset a precise fraction of the larger set. These different subsets were used to train the ControlNet keeping the maximum number of training steps constant.

The hypothesis 4 implies to examine the consistency of experimental results across various control signals, such as edges and depth maps, since each ControlNet model is trained for a specific condition. The hypothesis was validated only on the large-scale dataset; only models with edge control signals were trained on the small-scale dataset. Extraction of the control signal from the image is accomplished using off-the-shelf methods. We used the Canny edge detector (Canny, 1986) as implemented in OpenCV (Bradski, 2000) for edges extraction from the images, and the ZoeDepth model (Bhat et al., 2023), as implemented by authors for generating depth maps.

Training ControlNet for 15000 steps, which involves processing 7680000 samples, takes approximately 3 days on an A100 GPU. We trained ControlNets on nine different dataset sizes (1k, 5k, 10k, 25k, 50k, 100k, 500k, 1M images) and for two different control signals, Canny edges and depth maps. In total, we trained 18 models, which required 54 GPU-days. Due to time and computational constraints, we focused on two variations of control signals to validate our hypothesis 4. However, comparing scaling laws for more than two control signals would be an interesting direction for future work.

### 3.4 Evaluation

In (section 2.6), we discussed the difficulty of evaluating image generation tasks quantitatively compared to assessing discriminative models and LLMs. Selecting informative metrics is important for filling our key research gap. We considered several metrics options for this. The most obvious option is a training and validation loss of ControlNet (Zhang, Rao, and Agrawala, 2023); see equation 2.1.

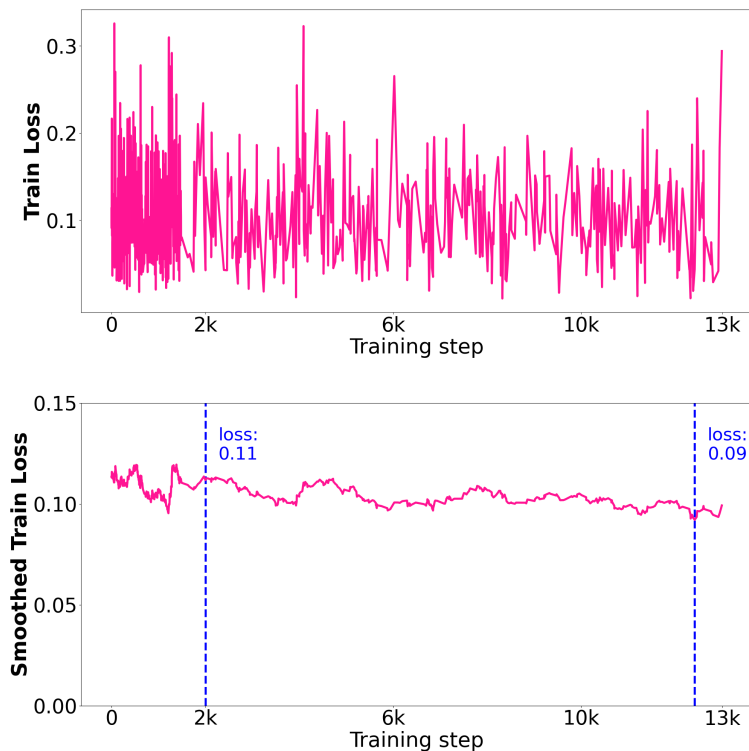


FIGURE 3.1: "Raw" (top) and smoothed (bottom) ControlNet training loss.

On the positive side, the training loss shows a slow decreasing trend during training, which one can see after smoothing it (see Fig. 3.1). For example, ControlNet that was trained on 1M images has 0.11 loss on the step  $2k$  and 0.09 loss on the step  $12.5k$ . Another signal which one can get from the ControlNet loss, is when severe overfitting starts to occur (more details in Section 5.2). On the negative side, the ControlNet neither provides a signal about the quality of the generated image, nor it tells much about how the generated image is following (or not) the task-specific condition (control signal).

The second option is traditional image quality metrics, like PSNR, SSIM (Wang et al., 2004), FID (Heusel et al., 2017) and PerceptualLoss (Johnson, Alahi, and Fei-Fei, 2016). They can be useful, but similarly to loss, they do not measure the correspondence of the generated image to the control signal.

So, we propose an alternative option, as outlined in Hypothesis 1 – to explore task-specific metrics for each type of the control signal, such as edge detection metrics for Canny edges, to assess the alignment between the control signal and its estimated counterpart derived from the generated image (see Fig. 3.2).

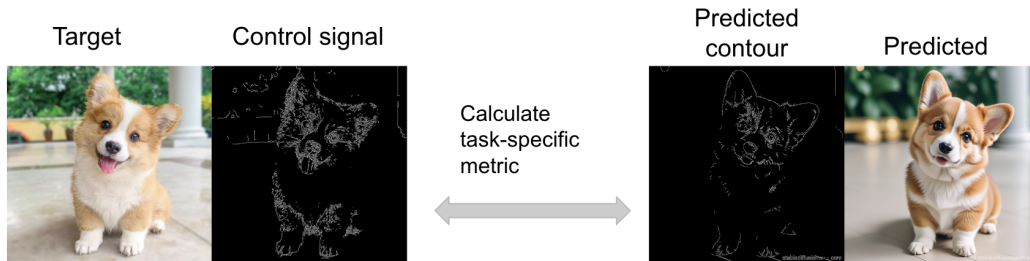


FIGURE 3.2: Hypothesis 1: the conditional image generation is of good quality and aligns with the control signal when the estimated contours/segmentation/etc on the generated image are the same as the control signal itself. Such alignment can be estimated with task-specific metrics.

**Task-specific (Canny edge) Evaluation metrics.** We employed two evaluation metrics for ControlNet which used Canny edges as input conditions: Optimal Dataset Scale (ODS) and Average Precision (AP) to compare the control signal with the predicted edges (see equations 3.1, 3.2). These metrics are usually used for evaluating the quality of edge detection algorithms (Xie and Tu, 2015).

$$AP = \frac{\sum_t \text{precision}_t}{\text{count of thresholds}} \quad (3.1)$$

$$ODS = \max_t (F1_t) \quad (3.2)$$

where  $t$  : threshold.

The metrics are computed between the Canny edges of the target image (used as the control signal) and the predicted Canny edges extracted from the predicted image. The Canny edges in the target and predicted images are extracted using hard Canny mode, resulting in two possible values after normalization: 0 for background

and 1 for edges. It renders the thresholds useless in metrics calculations. Consequently, the AP and ODS metrics become the standard precision and F1-score, respectively. The precision score is computed as the ratio of correctly predicted edges to all predicted edges, while recall is the ratio of correctly predicted edges to all target edges. F1-score, as usual, represents the harmonic mean between precision and recall.

We also consider a variant employing blurred predicted edges in the metrics computation to avoid penalization in metric assessments due to slight shifts (e.g. couple of pixels) in edge positions. So, in this case, while the target Canny edges images still consist solely of zeros and ones, the predicted edges are not. Therefore, the thresholds are beneficial in this context. Thresholds are sequential values from the range  $[0, 1)$  with the specified step when the edge image pixel values are normalized from 0 to 1. Pixels above the threshold are classified as edges and set to a value of 1, while those below the threshold represent the background and have a value of 0. ODS (equation 3.2) is a maximum F1-score among all thresholds and AP (equation 3.1) is the average precision among all thresholds.







	Target		Prediction
different edges		ODS	0.08
		AP	0.05
similar edges		ODS	0.21
		AP	0.18
identical edges		ODS	1.0
		AP	1.0
			
			
			

FIGURE 3.3: Edge detection metrics for different target and prediction pairs: top – unrelated images, center – similar, bottom – identical.

The Figure 3.3 shows the image examples and the ODS and AP metrics for 3 cases: unrelated images, similar and identical edges. The metrics improve appropriately as the similarity between the target and predicted edges increases, reaching

optimal values when the edges are identical. The practical optimal values align with the theoretical ones in this context.

The Figure 3.4 represents the same pairs, but when the predicted edges are blurred. The metrics show slight variations in metrics compared to unblurred edges for dissimilar and similar pairs, with higher  $ODS_{blur}$  but lower  $AP_{blur}$  values for blurred edges. In the case of identical images, the optimal achievable values are 0.45 for  $ODS_{blur}$  and 0.25 for  $AP_{blur}$ . But, the optimal values may vary slightly from one image to another.







	Target		Blurred Prediction
different edges		$ODS_{blur}$ 0.1 $AP_{blur}$ 0.04	
similar edges		$ODS_{blur}$ 0.28 $AP_{blur}$ 0.11	
identical edges		$ODS_{blur}$ 0.45 $AP_{blur}$ 0.25	

FIGURE 3.4: **Blurred** edge detection metrics for different target and prediction pairs. Top – unrelated images, center – similar, bottom – identical

**Task-specific (Depth map) Evaluation metrics.** We have tried nine popular metrics to estimate the similarity between the target depth map and the predicted one. In our experiments, we used ZoeDepth (Bhat et al., 2023) to generate depth maps as a control signal, so we applied the metrics used by Bhat et al., 2023 to evaluate the quality of the generated depth maps. All metrics are computed for every pair of target  $y_i$  and predicted  $\hat{y}_i$  pixels, and then these values are averaged across all  $N$  pixels of the depth map.

There are two relative errors: Absolute Relative Error (ARE) (equation 3.3) and Squared relative error (SRE) (equation 3.4). They represent how well the predicted

depth map  $\hat{y}_i$  aligns with the target depth map  $y_i$ .

$$ARE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \quad (3.3)$$

$$SRE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|^2}{y_i} \quad (3.4)$$

The next metrics are Root Mean Squared Error (RMSE) (equation 3.5) and Root Mean Squared Error Log scale (RMSELog) (equation 3.6), which show the variance in the residuals. RMSE exhibits scale variance, while RMSELog demonstrates scale invariance properties.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|^2} \quad (3.5)$$

$$RMSELog = \sqrt{\frac{1}{N} \sum_{i=1}^N |\ln(y_i) - \ln(\hat{y}_i)|^2} \quad (3.6)$$

The Absolute Log10 error ( $ABSLog_{10}$ ), as described in equation 3.7, shares a similar concept with RMSELog but employs slightly different mathematical calculations.

$$ABSLog_{10} = \frac{1}{N} \sum_{i=1}^N |\log_{10}(y_i) - \log_{10}(\hat{y}_i)| \quad (3.7)$$

Another metric is Scale Invariant Log error (SILog) (equation 3.8), that was firstly proposed by Eigen, Puhrsch, and Fergus, 2014. It shows the difference between average of squared errors and squared average of errors, where errors are calculated in the log scale, and is independent of the absolute global scale.

$$SILog = 100 * \sqrt{\frac{1}{N} \sum_{i=1}^N |\ln(\hat{y}_i) - \ln(y_i)|^2 - \left| \frac{1}{N} \sum_{i=1}^N \ln(\hat{y}_i) - \ln(y_i) \right|^2} \quad (3.8)$$

The threshold accuracy ( $\delta_n$ ) (equation 3.9) represents the percentage of pixels where the relative difference between the true  $y_i$  and predicted  $\hat{y}_i$  pixels falls within the scale factor of  $1.25^n$ . The scale factor determines how close the estimated depth needs to be to the ground truth depth for it to be considered accurate.

$$\delta_n = \% \text{ of pixels s.t. } \max\left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}\right) < 1.25^n, \text{ for } n = 1, 2, 3 \quad (3.9)$$

Three of the target and prediction depth maps pairs are shown in Figure 3.5. The upper depth maps are completely unrelated (although bottom part of both images are closer to the camera than the top part), which leads to high errors and low threshold accuracy. In the middle pair prediction repeats some patterns of the target image, so the metrics are better. And the lower images are identical, so the best possible values of all metrics are achieved. They coincide with the best possible theoretical values of each metric.

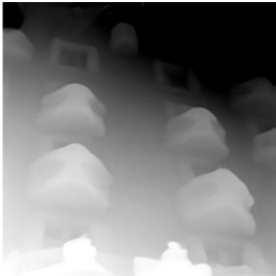
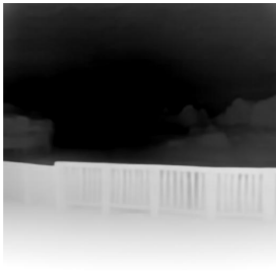
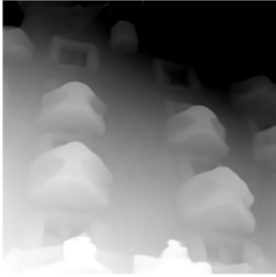
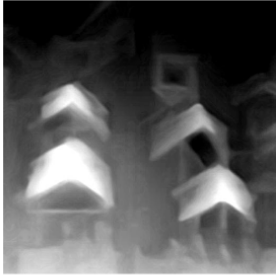

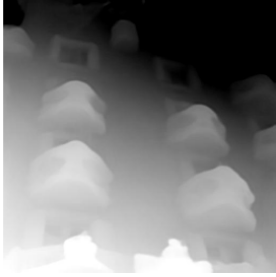
	Target			Prediction
different depth maps		ARE	1.286	
		SRE	4.296	
		RMSE	2.927	
		RMSELog	0.841	
		ABSLog10	0.335	
		SILog	33.302	
		$\delta_1$	0.04	
$\delta_2$	0.176			
$\delta_3$	0.424			
similar depth maps		ARE	0.171	
		SRE	0.152	
		RMSE	0.743	
		RMSELog	0.279	
		ABSLog10	0.085	
		SILog	25.82	
		$\delta_1$	0.675	
$\delta_2$	0.927			
$\delta_3$	0.934			
identical depth maps		ARE	0.0	
		SRE	0.0	
		RMSE	0.0	
		RMSELog	0.0	
		ABSLog10	0.0	
		SILog	0.0	
		$\delta_1$	1.0	
$\delta_2$	1.0			
$\delta_3$	1.0			

FIGURE 3.5: Depth map metrics for various target and prediction pairs: top – unrelated, center – similar, bottom – identical.



## Chapter 4

# Data

### 4.1 Small-scale fill50k dataset

**Training dataset.** The small-scale synthetic dataset is provided by the authors of the ControlNet Zhang, Rao, and Agrawala, 2023. It consists of 50k examples, namely circle lines as control signals, filled circles with colors as target images, and prompts, which describe the colors (see Fig. 4.1).

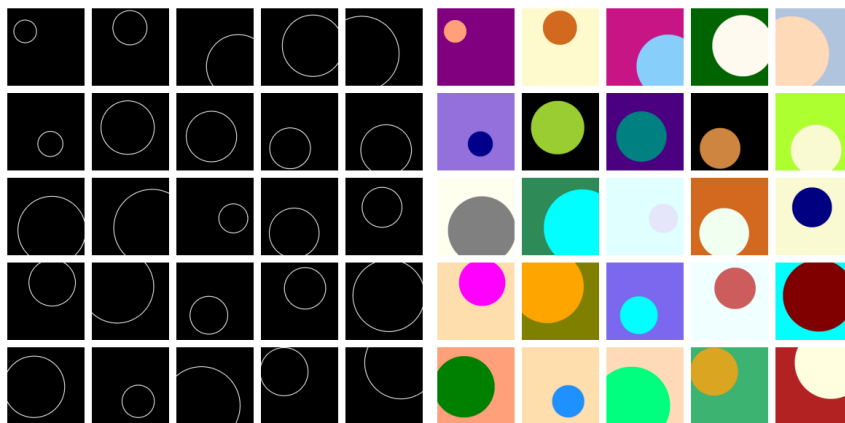


FIGURE 4.1: Training examples of the fill50k dataset

The task of the ControlNet model is to learn to color the circle indicated as a control signal, based on the given prompt (see Fig. 4.2).

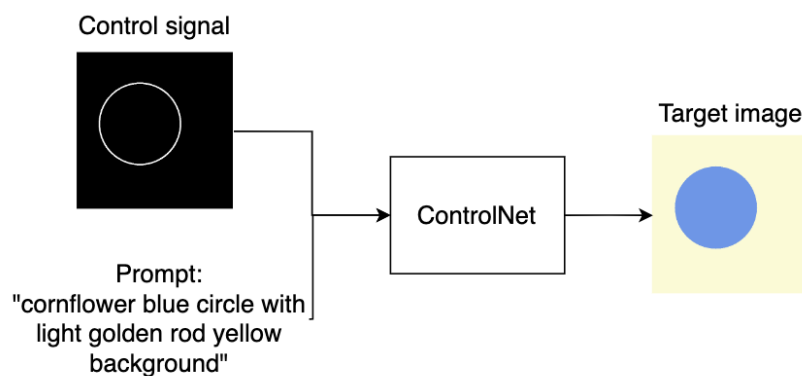


FIGURE 4.2: Training sample from small-scale experiment on fill50k dataset

## 4.2 Large-scale dataset

We start with the LAION5B dataset Schuhmann et al., 2022, a huge collection of 5.85 billion image-text pairs filtered with CLIPRadford et al., 2021, making it one of the largest publicly available datasets at the time of starting doing this work (October 2023). The just meta-data (URLs and captions) takes around 10 terabytes of disk space.

The LAION5B dataset consists of three subsets: Laion2B-en, Laion2B-multi, and Laion1B-nolang. The first subset includes captions only in English, while the other two contain captions in multiple languages. We used only the first subset. Considering our available resources (disk space, GPUs and time), we estimated the number of images we could download and use for training purposes to be around **160 million**. This subset was randomly selected from Laion2B-en.

The dataset itself does not contain images themselves, but the URLs, from where the images could be downloaded. Some URLs are obsolete, but the most of them are still valid. The dataset was utilized for Stable Diffusion Rombach et al., 2022 training.

**Stanford CSAM report.** The dataset online access has been revoked after December 2023, when David Thiel published an report (Thiel, 2023) revealing that the original dataset urls contained images of child sexual abuse material (CSAM). For the safety reasons, the report does not tell, which urls are those are.

After considering different options and a consultation with a lawyer we decide to perform an automatic filtering of the dataset, and that the using a filtered version of the dataset for the scaling laws estimation purpose does not generate a risk of abuse. To be on the safe side, we decided to *remove all the images, depicting people, altogether*.

**Two-iteration Filtering.** Given the dataset issue mentioned earlier, we decided to remove all images containing at least one person to ensure that our dataset remains free from potentially hazardous content. It's important to note that images containing people do not contribute significantly to the ControlNet's ability to learn the control signals which we use in our project, namely Canny edges and depth maps. And we do not require high-quality images of people in our results. Before applying the filter, we needed to download images using the dataset containing image URLs that we possess. The process of downloading images was enabled by the *img2dataset* library Beaumont, 2021, designed specifically for efficiently converting large-scale image URL datasets into image datasets within a reasonable timeframe. To identify objects within the images, we applied a popular and near the state-of-the-art object detection model known as the DETECTION TRANSFORMER (DETR)Carion et al., 2020 trained on MS COCO dataset.

The first iteration of the pipeline proceeded as follows:

1. Downloading images and resizing them all to a resolution of 256x256 before saving. This choice of a smaller resolution aimed to speed up the object detection process;
2. Assigning object-detection labels with images using DETR and saving files that match image URLs to their respective labels;
3. Filtering out URLs of images containing people, and deleting these images.
4. Downloading images with a resolution of 512 from the filtered URLs for subsequent use in the ControlNet training.

After the final step was done, we have discovered a bug in our preprocessing for the DETR model. The issue arose from resizing all images to a resolution of 256x256 with added borders. This consequently caused us to deviate from the default resizing method utilized by the DETR processor in Hugging Face/Hugging Face Team, 2024, resulting in degradation of the model accuracy. Given that we care more about deleting all the images containing people, and less about false positives, we decided to repeat the procedure on the higher resolution images, which survived the first stage. These images were then used for the second iteration of filtering. The default DETR processor was applied for resizing them before giving to the DETR model.

The label distribution of the initial 160M dataset after two-iteration filtering is shown in Figure 4.3. 33 million images containing people were filtered out during the first iteration, followed by an additional 2 million during the second iteration. Additionally, 53 million images were not downloaded due to errors related to urls being obsolete. 52 million images are left, where no objects were detected and 20 million images with labels other than 'person'. So, in total, we have a dataset comprising **72 million images** in total.

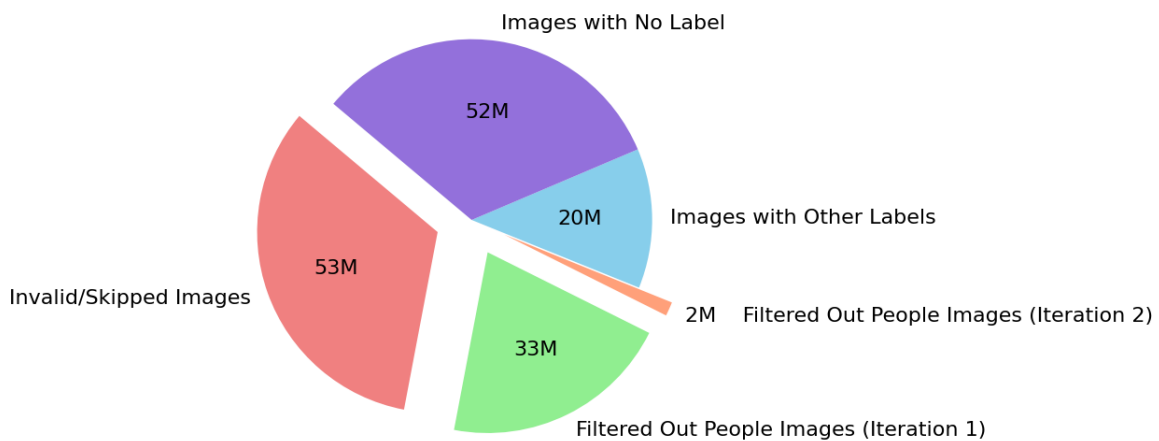


FIGURE 4.3: Distribution of initial 160M dataset after images downloading and two-iteration filtering by DETR labels

**Training dataset.** The distribution of the sizes of the images in the 72M dataset is presented in the Figure 4.4. Approximately 90% of the images have widths and heights ranging from 128 to 512 pixels. There is a noticeable shortage of high-resolution images sized 1024 pixels and above, as we resized images to 512 pixels while maintaining their aspect ratio.

We used the ZoeDepth model Bhat et al., 2023 to generate depth maps, which were cached to ensure the efficiency of the training process. The pre-processing of images for depth prediction consists of resizing images to 512 while maintaining the aspect ratio, then center cropping them to 512x512, and resizing them to 256x256. This resizing strategy ensures consistency between depth maps and images throughout ControlNet training. Before training, preprocessing also involves resizing and center cropping images to 512x512 pixels. The depth maps are stored in a 256x256 size to save disk space. The depth maps are upsampled back to 512x512 at the training time. The Canny edge method Canny, 1986 as implemented in OpenCV/Bradski,

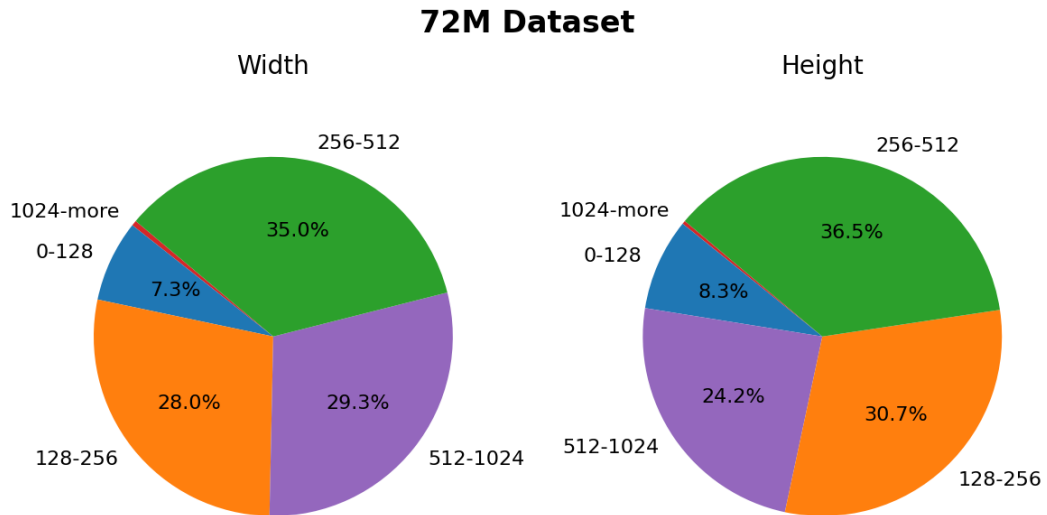


FIGURE 4.4: The size distribution of images in the 72M dataset

2000 works fast, so we opted to calculating them on-the-fly during the training. Examples of images from the training data with the appropriate depth maps, Canny edges and captions presented on the Figure 4.5, Figure 4.6 and Table 4.1 respectively.



FIGURE 4.5: Training images examples with respective depth maps

Finally, we have created a series of the subsets of the 72 million image dataset: 1k, 5k, 10k, 25k, 50k, 100k, 250k, 500k, 1M, 2.5M, 5M, 10M, 25M, 50M and 72M. Each smaller subset is fully contained in the bigger sets.

The training on them is performed consecutively, starting from the smallest, due to resource constraints. Specifically, training on A100 for 15k steps - (corresponds to model seeing 7680000 samples) takes around 3 days. At the time of submission, we have got results up until the 1M subset, which corresponds to 27 GPU-days in total, and the bigger ones are in process. We plan to continue training the models

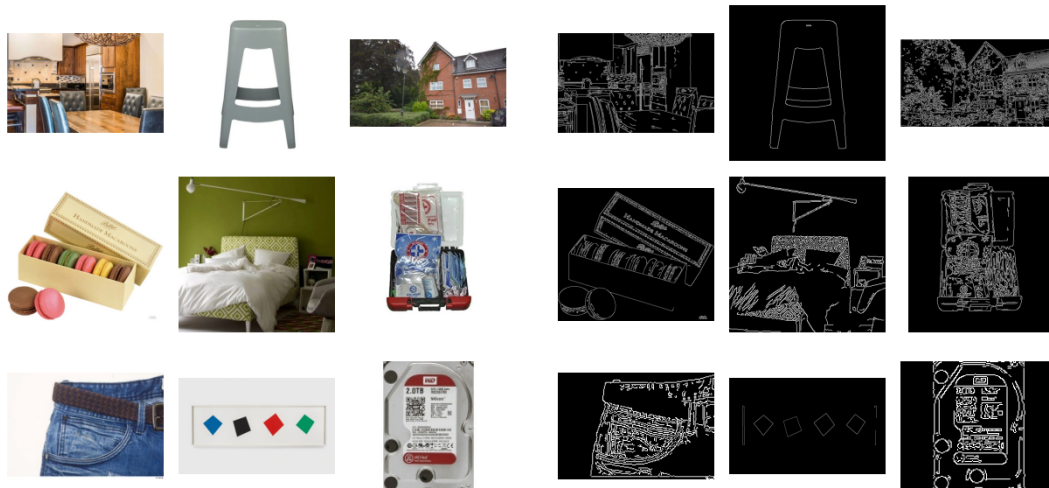


FIGURE 4.6: Training images examples with respective Canny edges

and publish updated results on arXiv.

The distribution of image sizes in the so-far-processed 1 million-image dataset presented in this work closely matches that of the 72 million-image dataset. It is shown in the Figure 4.7.

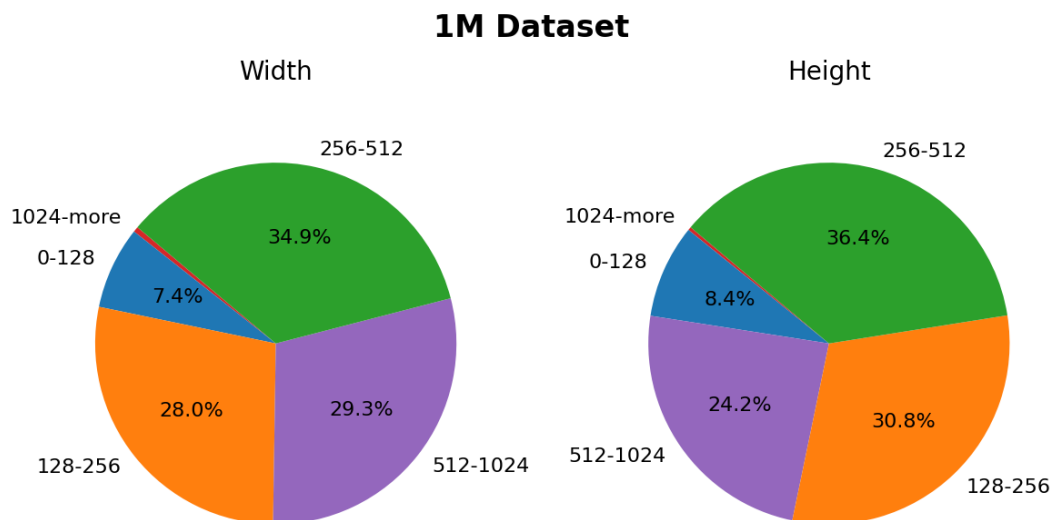


FIGURE 4.7: The size distribution of images in the 1M dataset - the largest subset, we have finished training on, at the time of submission

**Test dataset** We created a test dataset comprising 1000 images. Among them, 89 are our personal images that have never been on the internet, at least, before December 2023, ensuring they are entirely absent from the Stable Diffusion training data, or LAION5B. The remaining 911 images are from Laion2B-en, filtered using the same approach as the training data. All images were resized to 512x512 pixels. Captions were produced using the BLIP-2 model, developed by Li et al., 2023a, which takes an image as input and generates a textual description for it. The examples of the images and their caption from the test set are presented in the Table 4.2. The Figure 4.8 contains all 1000 images from the test set.










Image	Caption	Image	Caption	Image	Caption
	Suite Ascend Resort Collection Bluegreen Vacations Big Bear Village		Maxwell Bar Stool (Set of 4) Finish: Gray		3 Bedrooms End Of Terrace House for sale in Eaton Avenue, Slough, Berkshire
	1000 best Macarons images on Pinterest Boxes, Desserts and Food		Unleash cool bedroom ideas #bedroom #paint #color		Medique 40061 First Aid Kit, 61-Piece by Medique
	Clothes 240 blue jeans trousers 0003.jpg		Blue Black Red Green by Ellsworth Kelly		Western Digital Red 3.5 SATA III 2To (WD20EFRX)

TABLE 4.1: Training images examples with captions







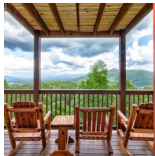


Image	Caption	Image	Caption	Image	Caption
	a wooden deck with a patio and a wooden porch		a building with many windows and balconies		shark in the aquarium
	a lizard is sitting on a cement wall next to a snow covered street		a balcony with a fire hydrant and a plant pot		a statue of a lizard laying on the ground
	a porch with rocking chairs overlooking the mountains		variety greeting card pack - greeting cards		the kawasaki motorcycle is parked on a white background

TABLE 4.2: Test images examples with BLIP2 (Li et al., 2023a) generated captions



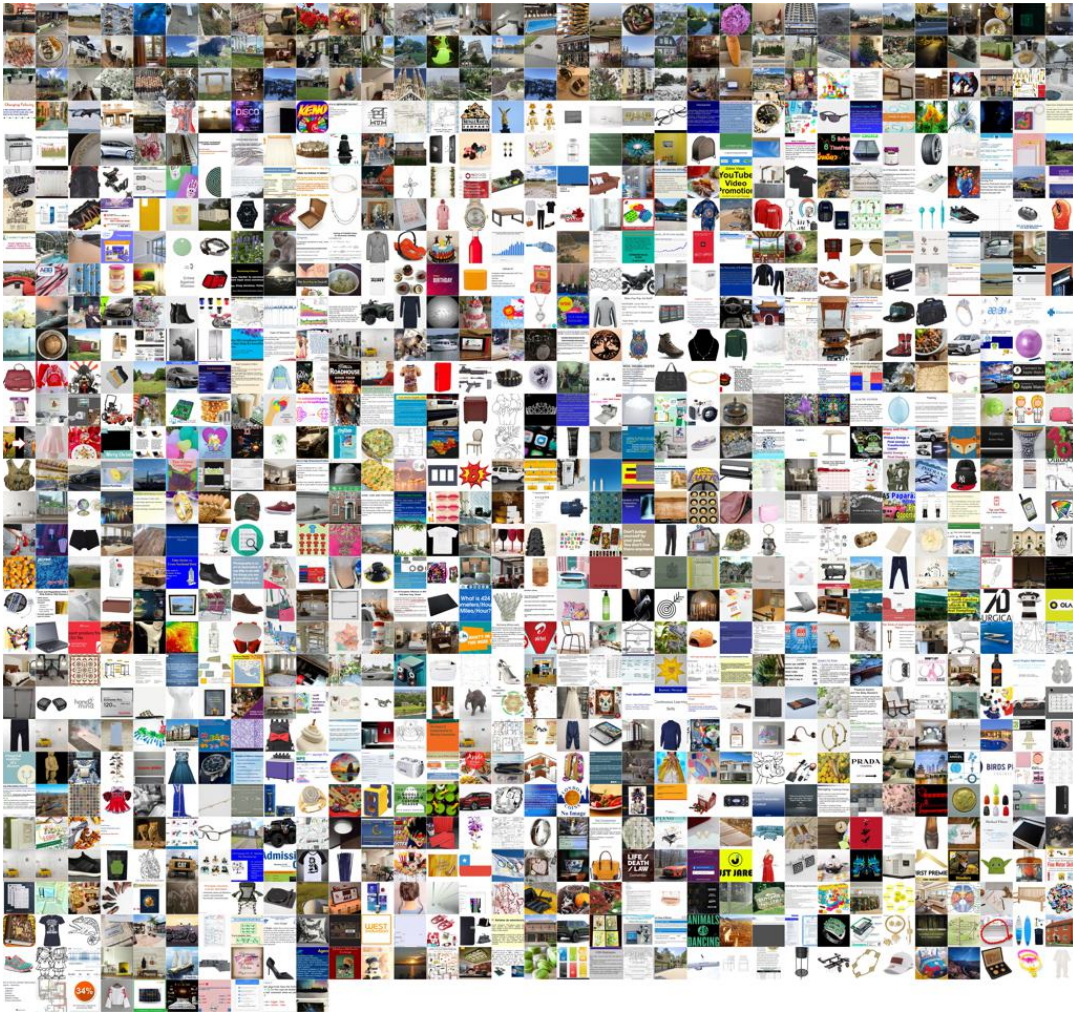


FIGURE 4.8: Entire test set of 1000 images

## Chapter 5

# Experiments

### 5.1 Small-scale dataset

**Task-specific metrics validation.** Research hypothesis 1 is confirmed on a small-scale dataset with the AP metric. The AP metric represent correspondence between the predicted image and target image edges. Also, the metrics display a sudden convergence phenomenon, described in the ControlNet paper Zhang, Rao, and Agrawala, 2023, when the model suddenly starts to follow the control signal after a certain step (see Fig. 5.1).

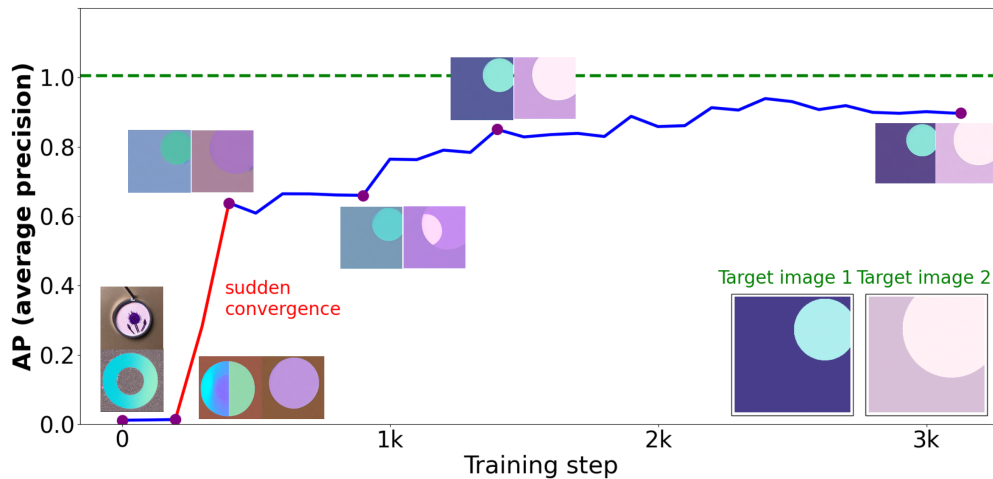


FIGURE 5.1: Correlation between AP metric and generated images correspondence to the target image. The metric also shows a so-called sudden convergence phenomenon.

**Dataset Sizes Experiment.** We conducted experiments varying the sizes of the training datasets to validate our second hypothesis. We employed subsets of the original dataset, ensuring that the smaller subset precisely constituted a portion of the larger subset, all while maintaining consistent training times. We compared the AP metric and training loss at the 3150 training step, which corresponds to 4 epochs on the full 50k dataset, for different dataset sizes, such as 500, 2.5k, 5k, 12.5k, 25k, and 50k. The obtained results confirm the second hypothesis, namely the larger the dataset size - the higher AP metric. The training loss doesn't represent this dependence. The results are shown in the Figure 5.2.

The two control signals from the validation set with predictions for them are shown in the Figure 5.3. The shown predicted images were generated by ControlNets, trained with different dataset sizes, where 100% corresponds to the full dataset consisting of 50k images, 50% to 25k images, etc. Additionally, the trained models



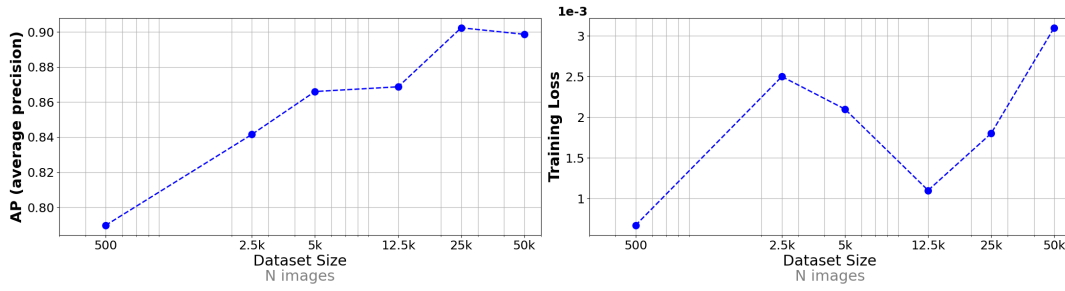


FIGURE 5.2: Correlation between metrics, such as AP and training loss, and different dataset sizes of the fill50k dataset.

were evaluated in a ‘no prompt’ mode to confirm whether the model truly learned to follow the control signal. The examples in the Figure 5.3 show that it does.

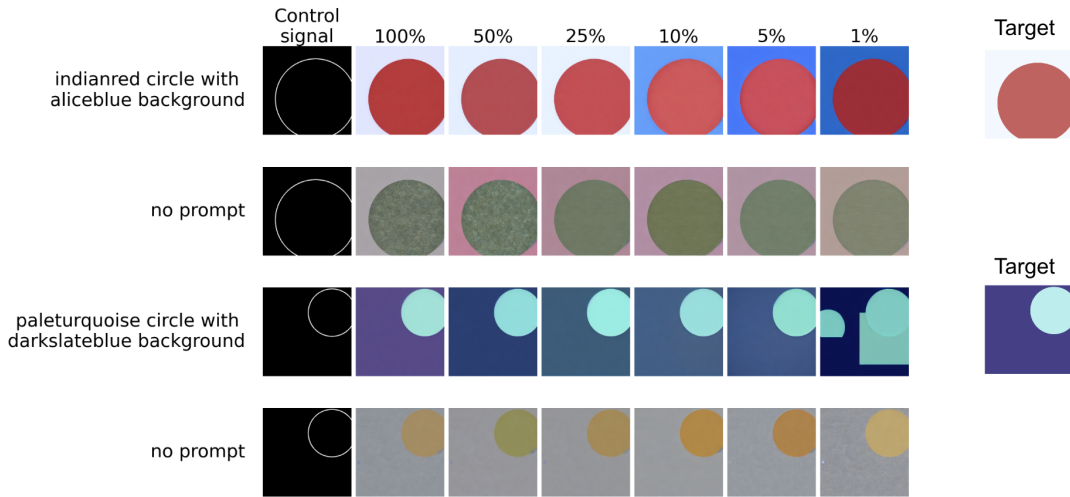


FIGURE 5.3: Validation-generated images from experiments with varying dataset sizes, predicted with prompt and in a ‘no prompt’ mode.

## 5.2 Large-scale dataset

**Task-specific metrics validation.** We confirmed the validity of our Hypothesis 1 using two types of control signals that the ControlNets were trained on, specifically Canny edges and depth maps. All task-specific metrics and validation losses were estimated on the test dataset contains of 1k images, which described in the Section 4.2.

The Figure 5.4 illustrates the correspondence of edge detection metrics, including AP, ODS, AP blur, and ODS blur, to the target image, throughout the training process using 1M images. All metrics clearly display a sudden convergence phenomenon, described in the ControlNet paper Zhang, Rao, and Agrawala, 2023, when the model suddenly starts to follow the control signal after a certain step. In our experiments with ControlNets, which utilizing Canny edges as condition, this occurred around steps 2k-3k, resulting in significantly improved similarity between the predicted and target images. Furthermore, the AP and ODS metrics demonstrate a noticeable upward trend following the sudden convergence, indicating an improvement in the quality of the predicted images. In contrast, the trends in AP blur and ODS blur metrics are less distinct. Additionally, metrics for the blurred versions of the conditioned images exhibit lower values for a good alignment between targets and predictions than non blurred metrics, especially AP blur. Thus we believe that using a shift-robust metrics is not important in practice.

The AP and ODS metrics plots are highly similar and moreover has a similar range, so they are both good to use.

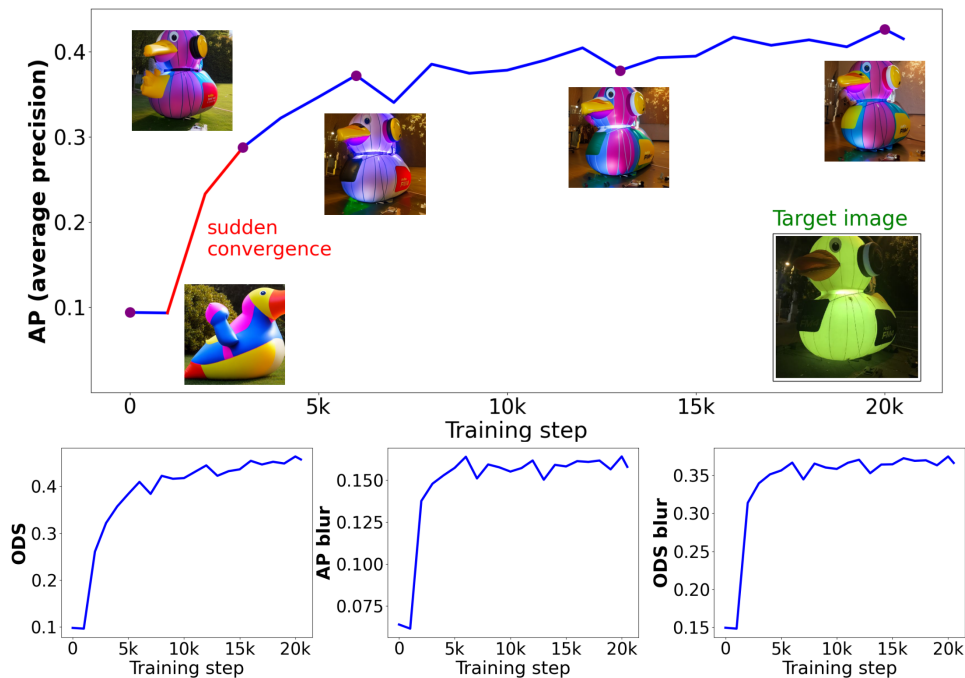


FIGURE 5.4: Correlation between edge detection metrics and generated images correspondence to the target image during the training process using 1M images with Canny edges as control signals. The metrics clearly show a "sudden convergence: phenomenon.

The experiments conducted with ControlNet, trained on depth maps of 1M images and corresponding depth metrics akin to Canny edge experiments, demonstrate a distinct sudden convergence phenomenon, as illustrated in Figure 5.5. But, the subsequent decreasing trend of the RMSE Log metric post-convergence is not as evident.

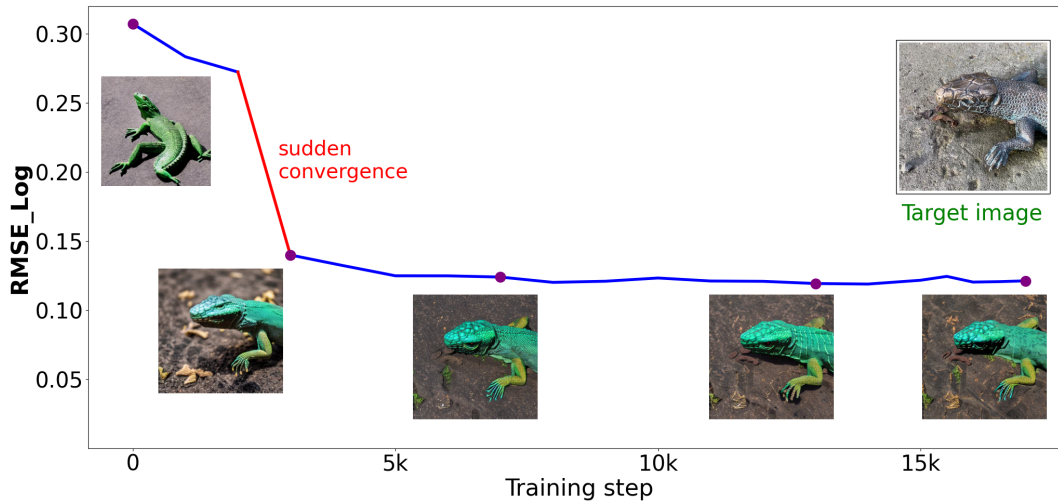


FIGURE 5.5: Correlation between RMSE Log metric and generated images correspondence to the target image during the training process using 1M images with depth maps as control signals. The metrics clearly represent a sudden convergence phenomenon.

Moreover, the plots of all metrics used for depth maps correspondence estimation exhibit a high degree of similarity to each other. Specifically, the plots where lower values are preferable are similar, as are those where higher values are preferable. They are shown in the Figure 5.6. The difference is only in the scale of the metrics.

Additionally, we examined the validation losses at corresponding steps to those used for task-specific metrics. The outcomes are shown in Figure 5.7. Overall, there is a decreasing tendency for the loss; however, loss appears to be more unstable compared to the task-specific metrics. Furthermore, while the large spikes in the task-specific metric plots indicate sudden convergence and substantial differences in correspondence of predicted image to target, the spikes in losses appear more random. For instance, the loss for the depth condition exhibits a sharp decrease followed by a sharp increase between 5k and 10k steps, potentially indicating overfitting and suggesting an early stopping in the training process at this point. But, examining the RMSE Log for the same steps reveals a relatively stable pattern. The visual inspection of generated results is in agreement with RMSE than a loss.

The training loss can be valuable for detecting overfitting in the ControlNet. Both the training losses and their corresponding validation losses are illustrated in Figure 5.8. The pink plots (upper) represent models trained on 1M images, while the green plots (lower) represent those trained on 1k images for the same amount of steps. When the training losses are rapidly decreasing, as shown in the lower-left chart, it is a clear indicator of overfitting. This can be confirmed firstly by examining the predicted image at the 12k step, which exhibit low quality and pixelation. Secondly, the validation loss demonstrates a consistent increase. Conversely, non-overfitted training losses, depicted in the upper-left chart, remain stable or exhibit slow decreases, indicating the model's learning process without overfitting.

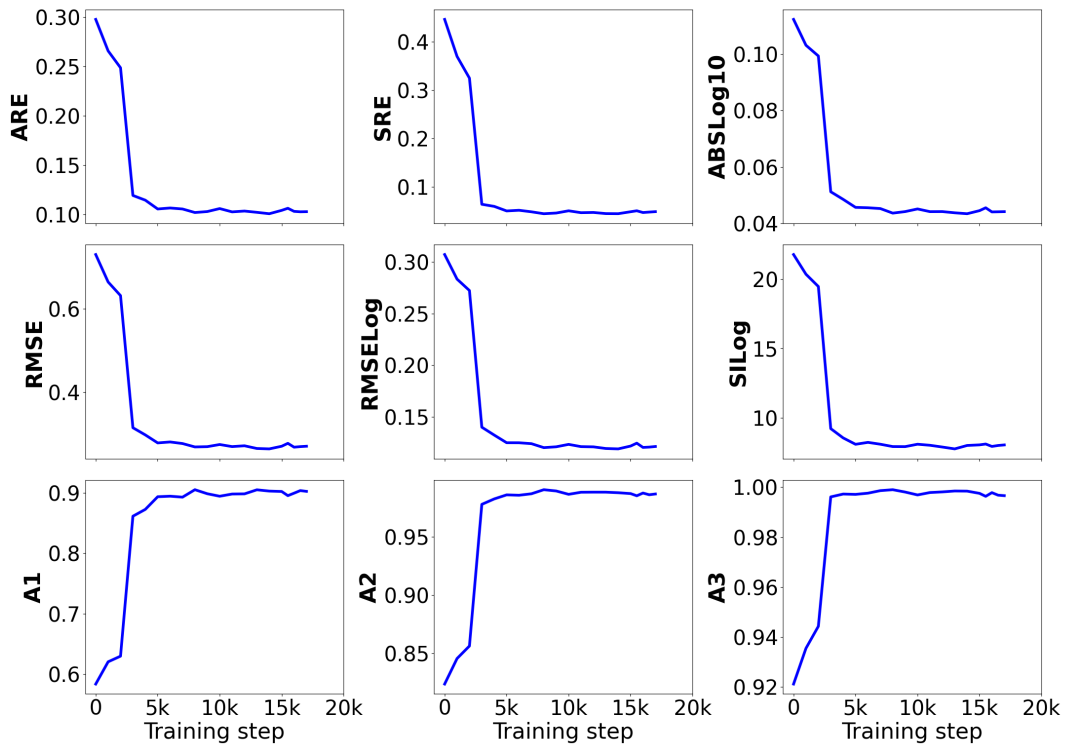


FIGURE 5.6: Full range of depth metrics utilized for establishing correspondence in depth maps during training on a 1M dataset. Their plots are highly similar, while the scale of the metrics is different.

Although the validation loss may show some spikes, it also remains relatively stable.

**Dataset Sizes Experiments.** In order to confirm Hypothesis 2 and accomplish our primary research objective of establishing scaling laws for ControlNet, we performed experiments using various dataset sizes. These experimental datasets included: 1k, 5k, 10k, 25k, 50k, 100k, 250k, 500k, and 1M images, with each smaller subset completely included within the larger sets.

The Figure 5.9 displays the AP metric for ControlNet trained with Canny conditions across various dataset sizes. Similarly, the Figure 5.10 showcases the RMSE Log metric for ControlNet trained with depth map conditions across different dataset sizes. The plots indicate significant variations in metrics across different dataset sizes throughout the training process. These differences are more noticeable for smaller datasets and reduce for larger ones.

When we possess metrics for the ControlNet trained across varying dataset sizes, we can graph the scaling laws line using these metrics. Our methodology involves selecting the best task-specific metric achieved by the model throughout training. Additionally, we visualize the relationship between validation losses at the selected checkpoints, chosen by best task-specific metric, for comparison purposes.

The AP metric for the ControlNet with Canny condition depending on the dataset sizes are depicted in Figure 5.11. The AP metric is monotonically increasing across all dataset sizes, which simplifies the formulation of scaling laws. Whereas the validation loss shows instability, posing challenges for formulating scaling laws based on the loss.

We selected two metrics for assessing the scaling laws for Controlnet with the

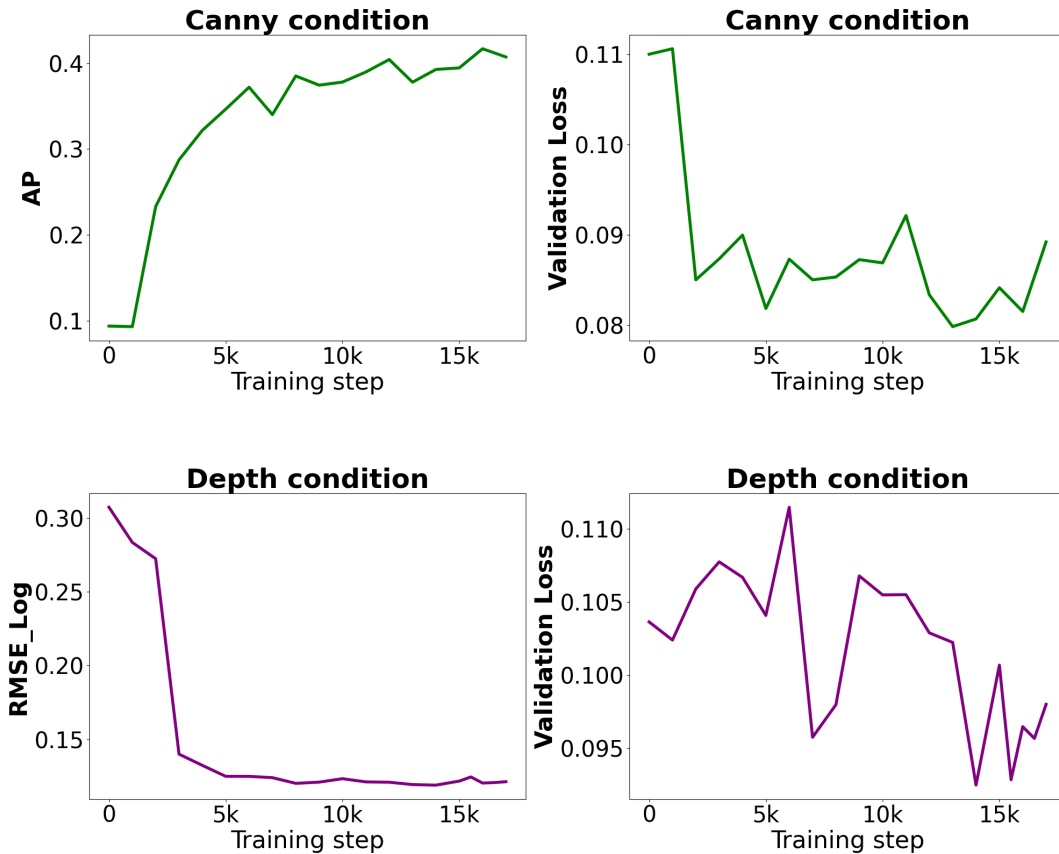


FIGURE 5.7: Comparison of task-specific metrics and validation losses. Task-specific metrics look more stable, while the validation loss exhibits random spikes.

depth condition: RMSE Log, where lower values indicate better performance, and threshold accuracy A1, where higher values indicate better performance. These metrics are shown in Figure 5.12. The situation with validation loss resembles ControlNet with Canny conditions. But, the task-specific metrics do not exhibit a fully monotonically increasing trend; notably, the values for the 500k dataset are worse than those for the 250k dataset.

**Scaling Laws.** Based on the metrics specific to the task, we can fit a parametric formula and establish scaling laws. We developed equations for both logarithmic and linear scales of dataset sizes. The given results are presented in the Figure 5.13 for ControlNet with Canny condition, in the Figure 5.14 for ControlNet with Depth condition.

In the case of the Canny condition, the empirical and fitted values alignment appear quite close, whereas for depth, they are not as closely aligned. One possible explanation for this disparity could be the unexpected values observed with the 500k dataset size. Additionally, the experimental and regression lines for RMSE Log and A1 metrics resembles each other, if one of them will be flipped. So, the fitted parametric estimates support the Hypothesis 3. However, it might be that different parameterization of the scaling law, compared to the commonly used ones would be more suitable.

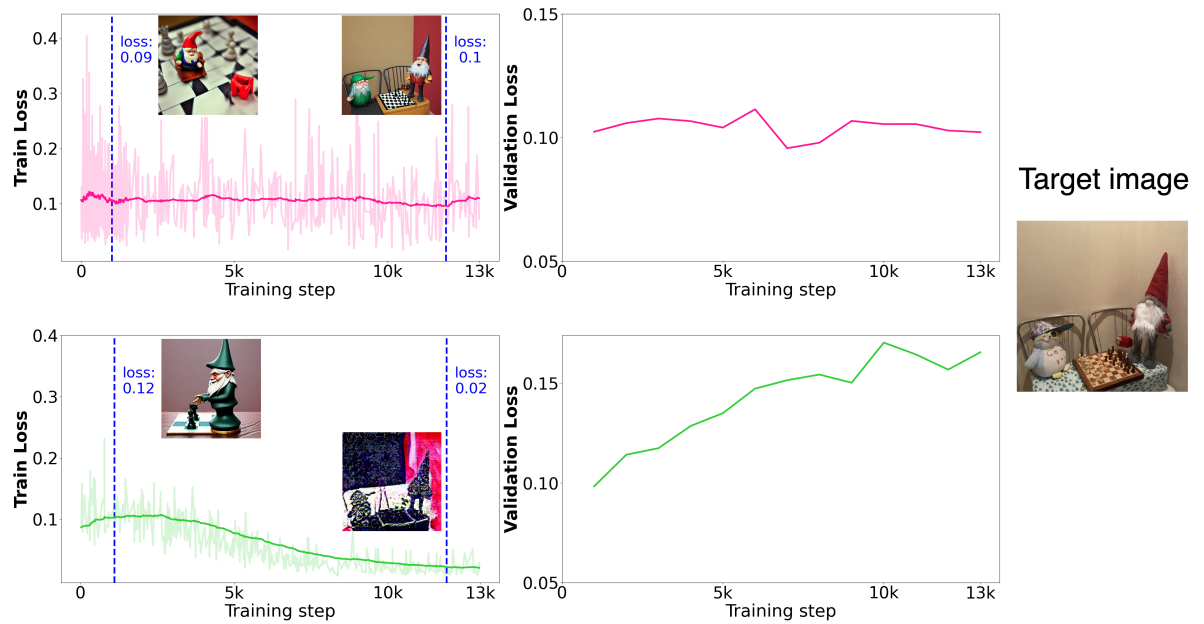


FIGURE 5.8: Significant decrease in ControlNet training loss might be an indication of the overfitting. This both can be observed in validation loss graph (going up), and the visual inspection of the generated images. The model is trained on 1k dataset

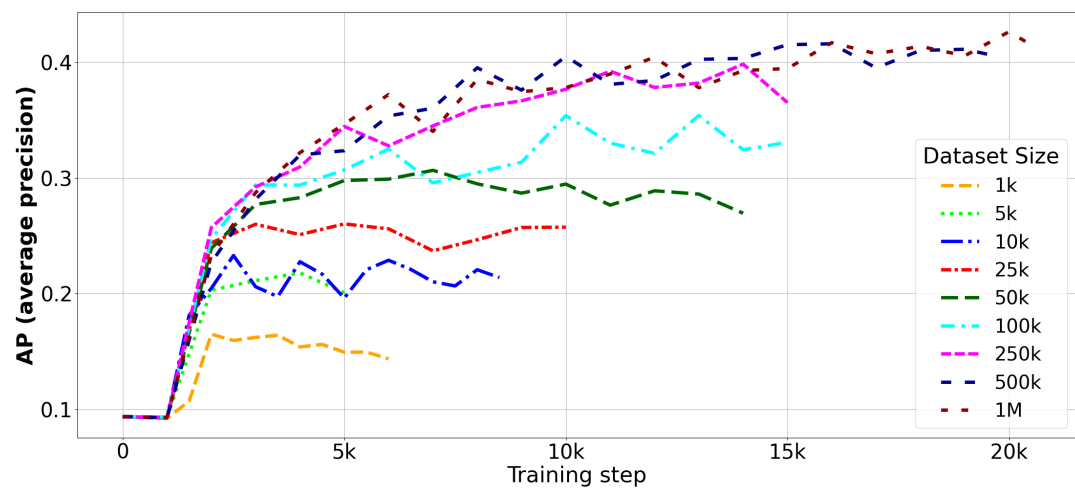


FIGURE 5.9: Canny edge condition: AP metric on the test set for ControlNet trained on various dataset sizes throughout the training process

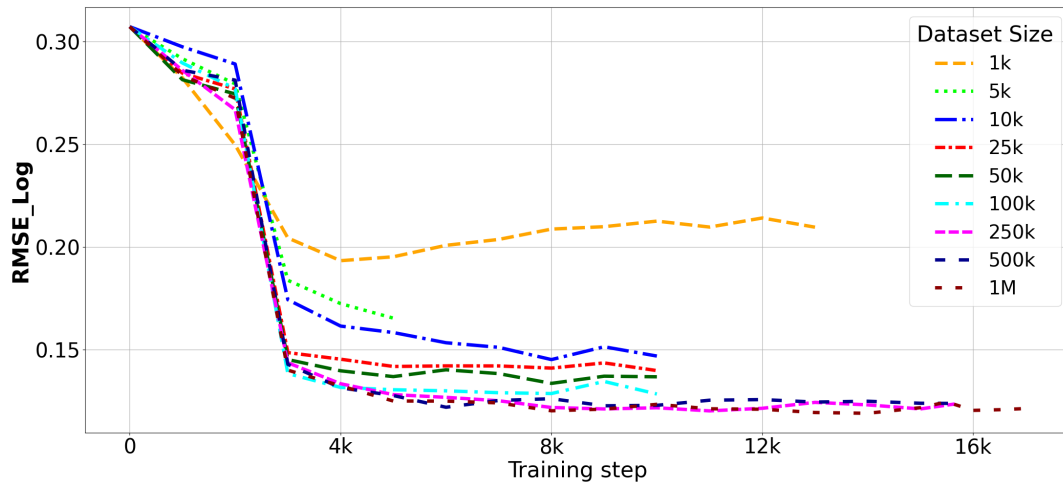


FIGURE 5.10: Depth condition: RMSE Log metric for ControlNet trained on various dataset sizes throughout the training process.

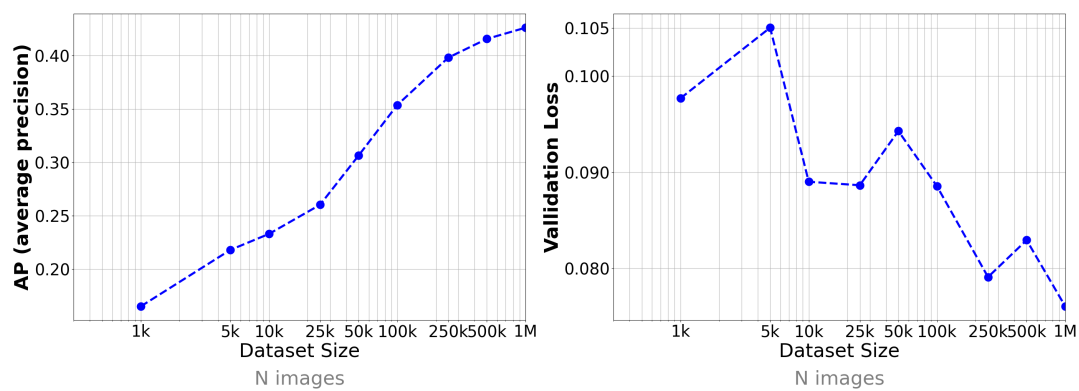


FIGURE 5.11: Canny edge condition: AP metric and validation loss depending on the dataset sizes.

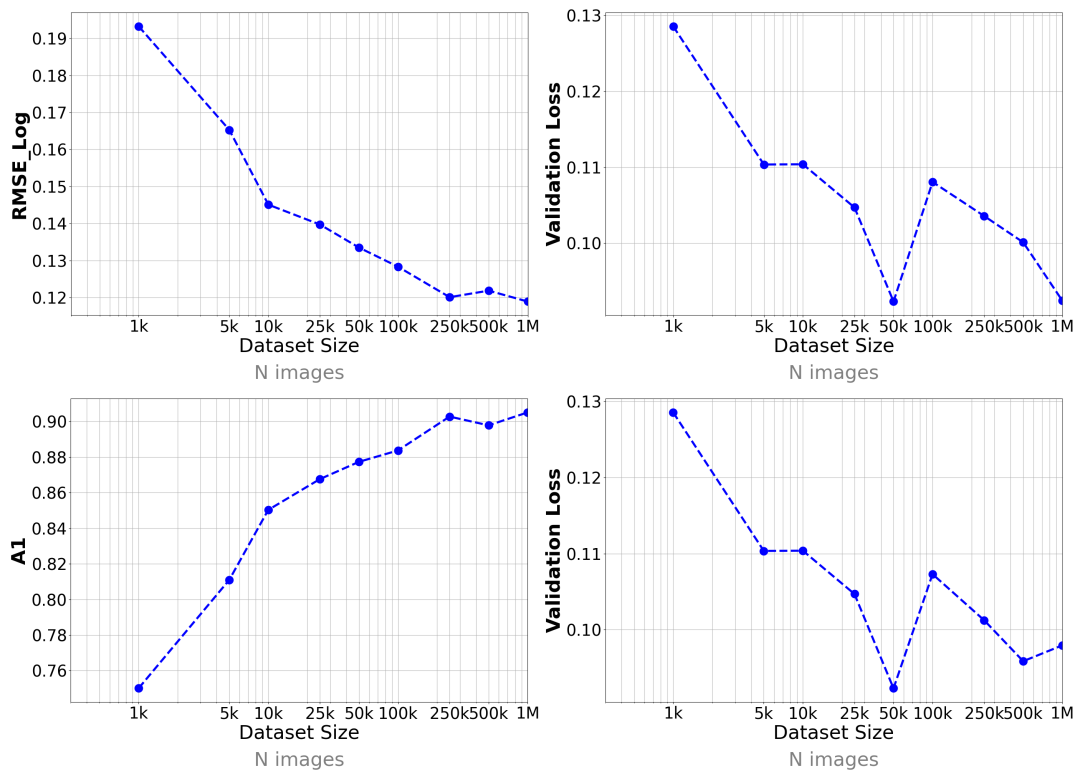


FIGURE 5.12: Depth condition: metrics depending on the dataset sizes for the ControlNet. Top: RMSE Log and validation loss, bottom: a1 (threshold accuracy).

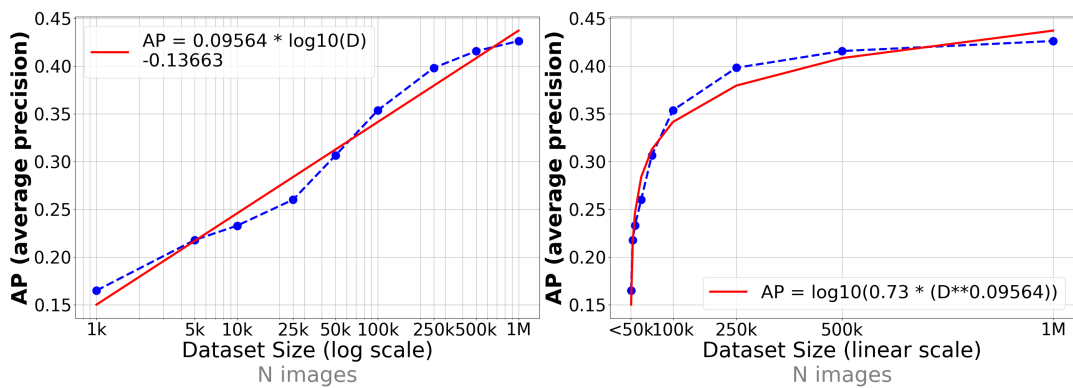


FIGURE 5.13: Canny edge condition: scaling laws for ControlNet based on AP metric



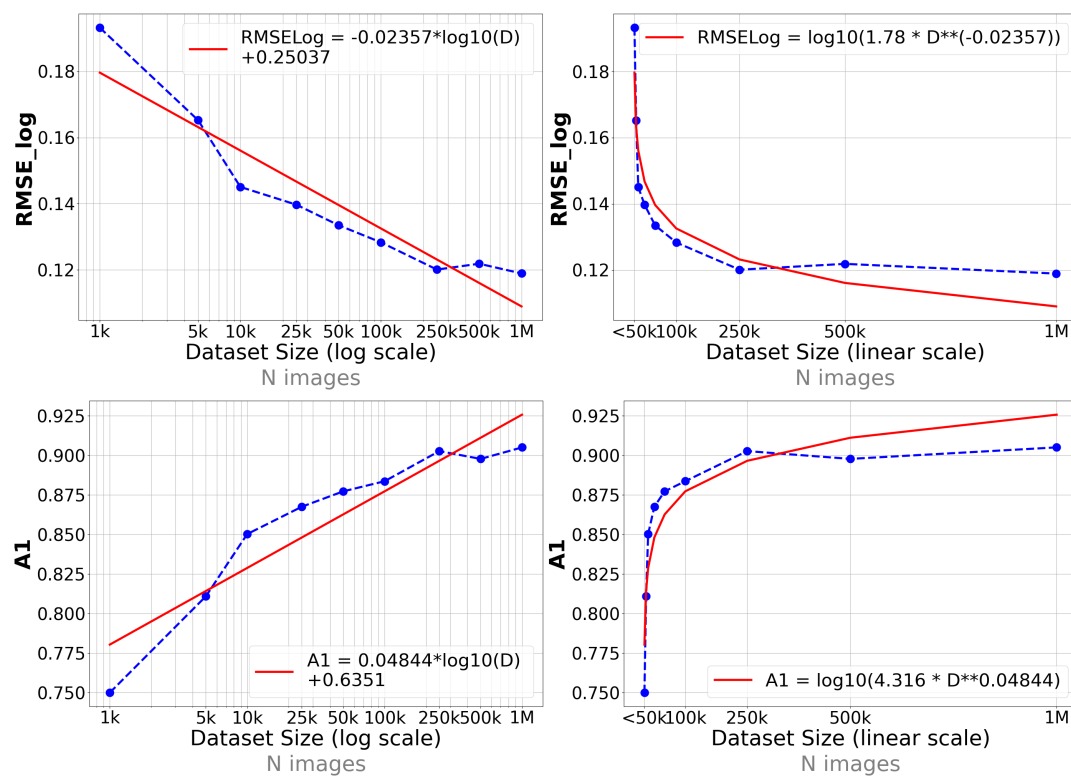


FIGURE 5.14: Depth condition: scaling laws for ControlNet, with RMSE Log metric (top) and a1 (bottom) metrics

## Chapter 6

# Conclusions

### 6.1 Discussion

The four research hypotheses were created and explored for this project. The first hypothesis posited that using task-specific metrics is reasonable in estimating alignment between target and predicted control signals. The second hypothesis focuses on the notion that larger dataset sizes lead to better generated images and metrics. The third hypothesis proposes a functional relationship between dataset size and performance metrics. Lastly, the fourth hypothesis examines the consistency of scaling laws across various control signals. Initially, the first two hypotheses (Hypothesis 1 and Hypothesis 2) were validated using a small-scale (synthetic) dataset which led to gaining valuable insights even before working with a larger dataset. Our findings indicate that task-specific metrics are reasonable indicators of the image quality and how the generated image follow the conditions, and models trained on larger datasets yield improved outcomes.

We have created a dataset for ControlNet training, comprising 72 million images containing target images, depth maps, and captions. Due to constraints in computational resources and time, the hypotheses were validated using only 1M images so far.

The **Hypothesis 1** was confirmed with the 1M dataset, similarly as it was for the small-scale data. Both metrics used for edge detection tasks and depth map estimation tasks demonstrate validity. It indicates, that the task-specific metrics are useful for estimating quality of the ControlNet predictions. Additionally, we conducted an exploration of the ControlNet loss, evaluating its utility and alignment with task-specific metrics. The ControlNet loss can be used for identifying overfitting and monitoring the training process, but it is not suitable for estimating scaling laws. **Hypothesis 2** was validated across nine diverse datasets of varying sizes, with larger datasets contain smaller ones entirely. The results demonstrate a clear correlation: as dataset size increases, there is a noticeable enhancement in the quality of generated images and task-specific metrics. The confirmation of **Hypothesis 3** occurred through fitting formulas using task-specific metrics, which were got during experiments of varying dataset sizes.

We estimated scaling laws for the AP metric in the ControlNet trained under the Canny condition, as well as for the RMSE Log and A1 metrics in the ControlNet trained under the Depth condition. From the practical point of view, **we recommend 250k as a dataset size** sweet spot between resulting generated image quality, and the effort of gathering the dataset.

Validation of the fourth **Hypothesis 4** turns out challenging due to the utilization of diverse metrics with distinct value ranges across different condition types. On the one hand, the estimated numerical scaling laws are numerically different for the

different control signals and metrics. On the other hand, they all start to saturate at dataset size 250k, showing the agreement in that regard.

## 6.2 Future Work

We plan to further train the ControlNet using larger datasets, specifically the 72M dataset that has already prepared for training. Also, as we have obtained meta-data for the entire LAION5B dataset and established the filtering pipeline for it, we can potentially increase the dataset size to the extent permitted by available computational resources. Another interesting area for exploration is valuating how data augmentation (commonly used in many computer vision tasks, but not image generation) influence the dependence of the quality on the dataset size.

Additionally, ControlNet can be trained using a wide range of diverse control signals, such as scribbles, human poses, segmentation masks, which can also be explored for further investigation. Moreover, since ControlNet can utilize various neural network blocks beyond just Stable Diffusion, it would be interesting to explore whether scaling laws remain consistent across different diffusion model architectures.

The findings from the proposed research project can be valuable for predicting the performance of the ControlNet for efficient utilization of resources. Considering the common application of scaling laws for LLMs and the increasing popularity of diffusion models, the outcomes are likely to benefit both the research community and the industry. Exploring the identified research gap in performance metrics for estimating ControlNet results opens the possibility for further investigation in other condition-based image generation models.

# Bibliography

- Ardalani, Newsha et al. (2022). “Understanding Scaling Laws for Recommendation Models”. In: *arXiv preprint arXiv:2208.08489*.
- Banko, Michele and Eric Brill (July 2001). “Scaling to Very Very Large Corpora for Natural Language Disambiguation”. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. Toulouse, France: Association for Computational Linguistics, pp. 26–33. DOI: [10.3115/1073012.1073017](https://doi.org/10.3115/1073012.1073017).
- Bansal, Yamini et al. (2022). “Data scaling laws in NMT: The effect of noise and architecture”. In: *ICML*. PMLR, pp. 1466–1482.
- Beaumont, Romain (2021). *img2dataset: Easily turn large sets of image urls to an image dataset*. <https://github.com/rom1504/img2dataset>.
- Bhat, S. et al. (2023). “ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth”. In: *ArXiv abs/2302.12288*. URL: <https://api.semanticscholar.org/CorpusID:257205739>.
- Bradski, G. (2000). “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools*.
- Brock, Andrew, Jeff Donahue, and Karen Simonyan (2018). “Large scale GAN training for high fidelity natural image synthesis”. In: *arXiv preprint arXiv:1809.11096*.
- Brown, Tom B. and et al. (2020). “Language Models are Few-Shot Learners”. In: *ArXiv abs/2005.14165*.
- Canny, John F. (1986). “A Computational Approach to Edge Detection”. In: *IEEE TPAMI PAMI-8*, pp. 679–698.
- Carion, Nicolas et al. (2020). “End-to-End Object Detection with Transformers”. In: *ArXiv abs/2005.12872*. URL: <https://api.semanticscholar.org/CorpusID:218889832>.
- Cherti, Mehdi et al. (2023). “Reproducible scaling laws for contrastive language-image learning”. In: *CVPR*, pp. 2818–2829.
- Chitwan, Saharia and et al (2022). “Photorealistic text-to-image diffusion models with deep language understanding”. In: *NeurIPS 35*, pp. 36479–36494.
- cloneofsimon (2022). *Low-rank Adaptation for Fast Text-to-Image Diffusion Fine-tuning*. <https://github.com/cloneofsimon/lora>. Accessed: 2023-12-11.
- Dhariwal, Prafulla and Alex Nichol (2021). “Diffusion Models Beat GANs on Image Synthesis”. In: *ArXiv abs/2105.05233*.
- Dosovitskiy, Alexey et al. (2020). “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929*.
- Droppo, Jasha and Oguz Elibol (2021). “Scaling laws for acoustic models”. In: *arXiv preprint arXiv:2106.09488*.
- Eigen, David, Christian Puhrsch, and Rob Fergus (2014). “Depth map prediction from a single image using a multi-scale deep network”. In: *Advances in neural information processing systems 27*.
- Gal, Rinon et al. (2022). “An image is worth one word: Personalizing text-to-image generation using textual inversion”. In: *arXiv preprint arXiv:2208.01618*.
- Gao, Leo, John Schulman, and Jacob Hilton (2023). “Scaling laws for reward model overoptimization”. In: *ICML*. PMLR, pp. 10835–10866.

- getimg.ai* (2023). <https://getimg.ai/use-cases/anime-ai-art-generator>. Accessed: 2023-12-15.
- Gordon, Mitchell A, Kevin Duh, and Jared Kaplan (2021). “Data and parameter scaling laws for neural machine translation”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5915–5922.
- Graham, Mark S et al. (2023). “Denoising diffusion models for out-of-distribution detection”. In: *CVPR*, pp. 2947–2956.
- He, Kaiming et al. (2015). “Deep Residual Learning for Image Recognition”. In: *CVPR*, pp. 770–778.
- Henighan, Tom et al. (2020). “Scaling laws for autoregressive generative modeling”. In: *arXiv preprint arXiv:2010.14701*.
- Hernandez, Danny et al. (2021). “Scaling laws for transfer”. In: *arXiv:2102.01293*.
- Hestness, Joel and et al (2017). “Deep learning scaling is predictable, empirically”. In: *arXiv preprint arXiv:1712.00409*.
- Hestness, Joel, Newsha Ardalani, and Gregory Diamos (2019). “Beyond human-level accuracy: Computational challenges in deep learning”. In: *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*, pp. 1–14.
- Heusel, Martin et al. (2017). “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *NeurIPS*.
- Ho, Jonathan (2022). “Classifier-Free Diffusion Guidance”. In: *ArXiv abs/2207.12598*.
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). “Denoising diffusion probabilistic models”. In: *NeurIPS*. Vol. 33, pp. 6840–6851.
- Hoffmann, Jordan et al. (2022). “Training compute-optimal large language models”. In: *arXiv preprint arXiv:2203.15556*.
- Hotpot.ai* (2023). <https://hotpot.ai/>. Accessed: 2023-12-15.
- HOVER Inc.* (2023). <https://hover.to/designer>. Accessed: 2023-12-15.
- Hu, Edward J et al. (2021). “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685*.
- Hugging Face Team (2024). *DETR*. [https://huggingface.co/docs/transformers/main/en/model\\_doc/detr](https://huggingface.co/docs/transformers/main/en/model_doc/detr). Accessed: 2024-05-14.
- Jiaming Song Chenlin Meng, Arash Vahdat (2023). *Denoising Diffusion Models: A Generative Learning Big Bang*.
- Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). “Perceptual losses for real-time style transfer and super-resolution”. In: *ECCV*. Springer, pp. 694–711.
- Kaplan, Jared et al. (2020). “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361*.
- Kolesnikov, Alexander et al. (2020). “Big transfer (bit): General visual representation learning”. In: *ECCV*, pp. 491–507.
- Krizhevsky, Alex (2009). “Learning Multiple Layers of Features from Tiny Images”. In: In.
- Kumari, Nupur et al. (2023). “Multi-concept customization of text-to-image diffusion”. In: *CVPR*, pp. 1931–1941.
- Lee, Kenny (2023). *Estimating Image Annotation Pricing for AI Projects*. <https://kili-technology.com/data-labeling/estimating-image-annotation-pricing-for-ai-projects>. Accessed: 2024-05-15.
- Li, Junnan et al. (2023a). “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *International Conference on Machine Learning*. URL: <https://api.semanticscholar.org/CorpusID:256390509>.
- Li, Yuheng et al. (2023b). “Gligen: Open-set grounded text-to-image generation”. In: *CVPR*, pp. 22511–22521.

- Midjourney (2022). <https://www.midjourney.com/>. Accessed: 2023-12-11.
- Mostaque, Emad (2022). *We actually used 256 A100s for this per the model card, 150k hours in total so at market price 600k*. <https://twitter.com/EMostaque/status/1563870674111832066>. Accessed: 2024-05-17.
- Mou, Chong et al. (2023). “T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models”. In: *arXiv preprint arXiv:2302.08453*.
- Netzer, Yuval et al. (2011). “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: *NeurIPS*.
- Neumann, Oren and Claudius Gros (2022). “Scaling laws for a multi-agent reinforcement learning model”. In: *arXiv preprint arXiv:2210.00849*.
- Nichol, Alexander Quinn and Prafulla Dhariwal (2021). “Improved denoising diffusion probabilistic models”. In: *ICML*. PMLR, pp. 8162–8171.
- Nie, Weili, Arash Vahdat, and Anima Anandkumar (2021). “Controllable and Compositional Generation with Latent-Space Energy-Based Models”. In: *NeurIPS*.
- OpenAI (2022). *DALL·E 2*. <https://openai.com/dall-e-2>. Accessed: 2023-12-11.
- (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].
- Peebles, William and Saining Xie (2023). “Scalable diffusion models with transformers”. In: *ICCV*, pp. 4195–4205.
- Radford, Alec et al. (2021). “Learning transferable visual models from natural language supervision”. In: *ICML*. PMLR, pp. 8748–8763.
- Rae, Jack W et al. (2021). “Scaling language models: Methods, analysis & insights from training gopher”. In: *arXiv preprint arXiv:2112.11446*.
- Ramesh, Aditya et al. (2022). “Hierarchical Text-Conditional Image Generation with CLIP Latents”. In: *ArXiv abs/2204.06125*.
- Rombach, Robin et al. (2022). “High-resolution image synthesis with latent diffusion models”. In: *CVPR*, pp. 10684–10695.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *MICCAI*. Springer, pp. 234–241.
- Rosenfeld, Jonathan S et al. (2019). “A constructive prediction of the generalization error across scales”. In: *arXiv preprint arXiv:1909.12673*.
- Ruiz, Nataniel et al. (2023). “Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation”. In: *CVPR*, pp. 22500–22510.
- Russakovsky, Olga et al. (2014). “ImageNet Large Scale Visual Recognition Challenge”. In: *IJCV* 115, pp. 211–252.
- Saharia, Chitwan et al. (2021). “Image Super-Resolution via Iterative Refinement”. In: *IEEE TPAMI* 45, pp. 4713–4726.
- Schuhmann, Christoph et al. (2022). “LAION-5B: An open large-scale dataset for training next generation image-text models”. In: *ArXiv abs/2210.08402*.
- Shen, Li et al. (2023). “On Efficient Training of Large-Scale Deep Learning Models: A Literature Review”. In: *ArXiv abs/2304.03589*.
- Sohl-Dickstein, Jascha et al. (2015). “Deep unsupervised learning using nonequilibrium thermodynamics”. In: *ICML*. PMLR, pp. 2256–2265.
- Sorscher, Ben et al. (2022). “Beyond neural scaling laws: beating power law scaling via data pruning”. In: *NeurIPS* 35, pp. 19523–19536.
- Sun, Chen et al. (2017). “Revisiting unreasonable effectiveness of data in deep learning era”. In: *ICCV*, pp. 843–852.
- Tay, Yi et al. (2022). “Scaling laws vs model architectures: How does inductive bias influence scaling?” In: *arXiv preprint arXiv:2207.10551*.
- Tewel, Yoad et al. (2023). “Key-locked rank one editing for text-to-image personalization”. In: *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11.

- Thiel, David (2023). “Identifying and Eliminating CSAM in Generative ML Training Data and Models. Stanford Digital Repository”. In: URL: <https://purl.stanford.edu/kh752sm9123>.
- Vasu, Subeesh, Nimisha Thekke Madam, and AN Rajagopalan (2018). “Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network”. In: *ECCVW*, pp. 0–0.
- Vaswani, Ashish et al. (2017). “Attention is All you Need”. In: *NeurIPS*.
- Wang, Zhou et al. (2004). “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13, pp. 600–612.
- Xie, Saining and Zhuowen Tu (2015). *Holistically-Nested Edge Detection*. arXiv: [1504.06375](https://arxiv.org/abs/1504.06375) [cs.CV].
- Zhai, Xiaohua et al. (2022). “Scaling vision transformers”. In: *CVPR*, pp. 12104–12113.
- Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala (2023). “Adding conditional control to text-to-image diffusion models”. In: *ICCV*, pp. 3836–3847.