AutomaTikZ: Text-Guided Synthesis of Scientific Vector Graphics with TikZ

Jonas Belouadi
Natural Language Learning Group
Bielefeld University, Germany
jonas.belouadi@uni-bielefeld.de

Anne Lauscher
Data Science Group
University of Hamburg, Germany
anne.lauscher@uni-hamburg.de

Steffen Eger
Natural Language Learning Group
University of Mannheim, Germany
steffen.eger@uni-mannheim.de

Abstract

Generating bitmap graphics from text has gained considerable attention, yet for scientific figures, vector graphics are often preferred. Given that vector graphics are typically encoded using low-level graphics primitives, generating them directly is difficult. To address this, we propose the use of Ti*k*Z, a well-known abstract graphics language that can be compiled to vector graphics, as an intermediate representation of scientific figures. Ti*k*Z offers human-oriented, high-level commands, thereby facilitating conditional language modeling with any large language model. To this end, we introduce DaTikZ, the first large-scale Ti*k*Z dataset consisting of 120k Ti*k*Z drawings aligned with captions. We fine-tune LLaMA on DaTikZ, as well as our new model CLiMA, which augments LLaMA with multimodal CLIP embeddings. In both human and automatic evaluation, CLiMA and LLaMA outperform commercial GPT-4 and Claude 2 in terms of similarity to human-created figures, with CLiMA additionally improving text-image alignment. Our detailed analysis shows that all models generalize well and are not susceptible to memorization. GPT-4 and Claude 2, however, tend to generate more simplistic figures compared to both humans and our models. We make our framework, AutomaTikZ, along with model weights and datasets, publicly available.[1]

1 Introduction

Recent advancements in text-to-image generation have facilitated the generation of detailed images from simple natural language descriptions (Esser et al., 2021; Ramesh et al., 2021; 2022; Saharia et al., 2022; Rombach et al., 2022; Zhang et al., 2023a). Models like Stable Diffusion (Rombach et al., 2022) and DALL-E (Ramesh et al., 2021; 2022) often yield results comparable to real photographs or human-created artworks. However, these models primarily generate *raster graphics*, typically at low resolutions, which are not ideal for *scientific figures*. Researchers use scientific figures to convey complex ideas or present critical findings, making them central to scientific research (Tufte, 1992; Hsu et al., 2021). Consequently, they demand a high degree of geometric precision and legible text, even at small font sizes, areas where raster graphics fall short. As a result, many research conferences advocate the use of *vector graphics*,[2] which decompose information into geometric shapes, allow searchable text, and usually have smaller file sizes.

Automated vector graphics generation is a growing research area as well (Lopes et al., 2019; Carlier et al., 2020; Aoki & Aizawa, 2022; Ma et al., 2022; Frans et al., 2022; Jain et al., 2023; Wu et al., 2023), but current methods have their own share of limitations. Specifically, they mainly generate low-level path elements of the Scalable Vector Graphics (SVG) format, either failing to maintain

[1] https://github.com/potamides/AutomaTikZ
[2] https://acl-org.github.io/ACLPUB/formatting.html

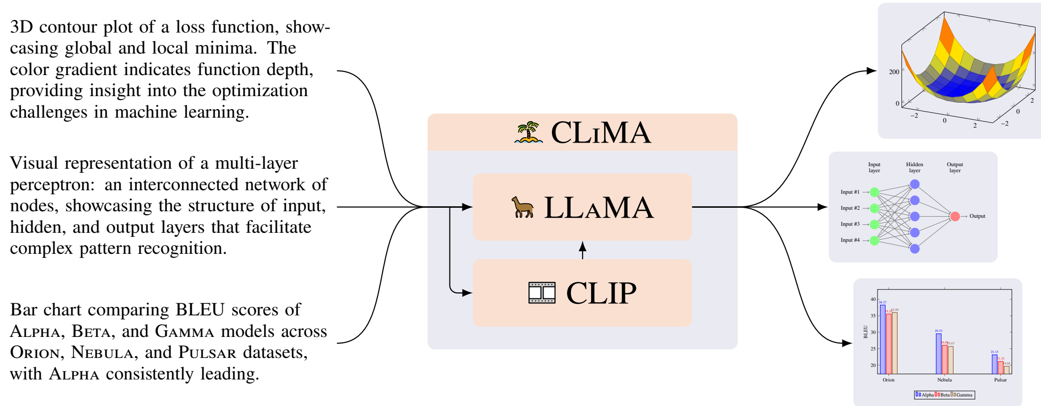


Figure 1: Exemplary scientific figures generated with CLIMA. CLIMA takes the captions as input, processes them with CLIP and LLAMA, and generates TikZ drawings that compile to vector graphics.

accurate geometric relations (Ma et al., 2022; Frans et al., 2022; Jain et al., 2023) or only generating outputs of limited complexity such as single icons or font characters (Lopes et al., 2019; Carlier et al., 2020; Aoki & Aizawa, 2022; Wu et al., 2023).

To address these limitations, we explore the use of *graphic languages*, which abstract from lower-level vector graphics formats by providing high-level constructs that can be compiled to such formats (Van Zandt, 2007; Hobby, 2014; Tantau, 2023). Language models show potential in learning these languages to solve simple tasks (Bubeck et al., 2023; Zhang et al., 2023b), but the depth of this capability, i.e., whether it can produce scientific figures, remains unexplored. Due to its expressiveness and emphasis on science, which enables the creation of complex figures with only a few commands, we focus on the graphics language *TikZ* in this work (Tantau, 2023). We aim to understand whether language models can capture the nuances of TikZ and automatically generate scientific figures based on image captions, analogous to text-to-image generation. This could not only enhance productivity and foster inclusiveness (aiding researchers less versed in programming-like languages, such as social scientists), but also aid education by creating tailored TikZ examples. The use case for this is demonstrated by the TEX Stack Exchange[3], where nearly 10% of the asked questions pertain to TikZ, making it the most frequently discussed topic on the platform. Our key contributions are as follows:

(i) As part of our AUTOMATIKZ project, we create DATIKZ, the first large-scale TikZ dataset to our knowledge, featuring approximately 120k paired TikZ drawings and captions.

(ii) We fine-tune the large language model (LLM) LLAMA (Touvron et al., 2023a) on DATIKZ and compare its performance to general-purpose LLMs, specifically GPT-4 (OpenAI, 2023) and CLAUDE 2 (Anthropic, 2023). Both automatic and human evaluation agree that scientific figures generated by fine-tuned LLAMA resemble human-created figures more closely.

(iii) We further develop CLIMA, a variant of LLAMA augmented with multimodal CLIP embeddings (cf. Figure 1; Radford et al., 2021). This enhancement allows CLIMA to visually interpret input captions, thereby improving text-image alignment. It also enables the use of images as supplementary inputs, leading to a further boost in performance.

(iv) In addition, we demonstrate that all models exhibit few memorization problems and generate novel outputs. However, GPT-4 and CLAUDE 2 tend to generate simpler outputs than LLAMA and CLIMA, sometimes producing degenerate solutions that maximize text-image similarity by visibly copying the input caption into the output image.

2 RELATED WORK

Our work connects to several distinct but interrelated fields, namely text-to-image generation, vector graphics generation, scientific figure understanding, and code generation. For each field, we provide a comprehensive review of the most relevant prior work.

[3] https://tex.stackexchange.com

Text-to-Image & Vector Graphics Generation The evolution of text-to-image generation can be characterized by three development stages: Generative Adversarial Networks (Reed et al., 2016; Zhang et al., 2017; Brock et al., 2019; Kang et al., 2023), auto-regressive models (Ramesh et al., 2021; Ding et al., 2021; 2022; Chang et al., 2023), and diffusion models (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Zhang et al., 2023a). Although Rodriguez et al. (2023a;b) explore their use for scientific figures, these approaches are inherently limited to generating raster graphics.

Vector graphics generation has evolved as a parallel field. Building upon innovative work in sketch generation (Ha & Eck, 2018), Lopes et al. (2019) generate single SVG font characters made up of straight lines and Bézier curves. Carlier et al. (2020) extend this approach to include SVG *icons*. However, none of these models support text conditioning. Another branch of research focuses on vectorizing text-to-image models (Ma et al., 2022; Frans et al., 2022; Jain et al., 2023). Although these methods enable text conditioning, text-to-image models typically have difficulties producing flat-colored SVG-style images, and the vectorized results tend to have imprecise geometric relations and jagged paths (Wu et al., 2023). Addressing these challenges, Cai et al. (2023) and Wu et al. (2023) investigate auto-regressive language modeling directly on SVG representations. Even though these approaches better capture the aesthetics of vector graphics, they are limited to editing existing graphics or generating monochrome icons of limited complexity.

Scientific Figure Understanding Despite the limited number of approaches dedicated to generating scientific figures, scientific figure *understanding* is a subject of extensive research. Arguably, the task that is inverse to ours is the captioning of scientific figures. Expanding on prior work in scientific visual question-answering (Siegel et al., 2016; Kahou et al., 2018; Kafle et al., 2018), Chen et al. (2019a;b; 2020) train a captioning model using a corpus of synthesized figure-caption pairs. Hsu et al. (2021) extend this approach to real scientific figures, noting substantial challenges. To improve performance, Yang et al. (2023) reformulate the task, augmenting captions with OCR tokens and paragraphs referencing the figures. Singh et al. (2023) take a different approach and utilize reinforcement learning to consider the quality of the captions during training. In addition to such task-oriented models, recent advancements in multimodal large language modeling (MLLM; Liu et al., 2023; Dai et al., 2023; Yin et al., 2023) allow for generalized visual reasoning about scientific figures (Ye et al., 2023; Zhang et al., 2023c; Horawalavithana et al., 2023).

Code Generation As graphics languages are a subset of programming languages, our work is closely related to code generation (Xu et al., 2022). At its core is the ongoing research on pre-training or fine-tuning LLMs on code (Rozière et al., 2023; Li et al., 2023; Fried et al., 2023; Li et al., 2022b; Chen et al., 2021), commonly with a multitask objective (Fried et al., 2023) consisting of causal language modeling and infilling prediction (Bavarian et al., 2022). Despite the significant amount of recent progress, the primary focus of code generation remains on high-resource programming languages such as Python, Java, or JavaScript (Zan et al., 2023). Ti*k*Z commands, in comparison, are invoked as TEX macros, and the TEX programming language is considered low-resource and typically overlooked in model evaluations. Yet, TEX may still exist in training corpora, as evidenced by GPT-4's ability to comprehend TEX and Ti*k*Z (Bubeck et al., 2023; Zhang et al., 2023b). As far as we know, there has been no comprehensive evaluation of this capability, which we also address in this work.

3 The DaTi*k*Z Dataset

DaTi*k*Z is, to our best knowledge, the first large-scale dataset of Ti*k*Z drawings with corresponding captions. Ti*k*Z is well-known within the TEX community, but its resources are scattered across the internet, making the creation of a large dataset a fundamental challenge of our work. Consequently, we gather Ti*k*Z drawings and their captions from a variety of online resources, as detailed below.

3.1 Data Acquisition

We collect the data from dedicated websites, online repositories, the TEX Stack Exchange, arXiv papers, and also artificial examples. A comprehensive overview of our data collection is provided in Table 1. We gather Ti*k*Z drawings created between November 2006 and June 2023 that successfully compile with TEX Live 2023.[4] Ablation studies and examples can be found in Appendices C and E.

[4] https://tug.org/texlive

Source	Size	Augmented
Curated Examples	981	63.2%
TEX Stack Exchange	29 238	51.31%
ArXiv Papers	85 656	67.75%
Artificial Examples	3 914	50%
All	119 789	62.71%

Table 1: Detailed breakdown of DaTikZ showing size and percentage of augmented data for the whole dataset and each source individually.

Curated Examples Several websites and online repositories[5] focus on collecting and sharing Ti*k*Z drawings for educational purposes and to showcase high-quality examples. Through web scraping, we retrieve any Ti*k*Z drawings from these sites that have associated captions.

TEX Stack Exchange We also source Ti*k*Z drawings from TEX Stack Exchange (cf. §1). We examine the quarterly data dump and extract questions tagged with Ti*k*Z and relevant answers with a minimum score of 1. To convert textual questions into image captions, we utilize WizardLM (Xu et al., 2023), an LLM trained to follow arbitrary instructions. Using the title and body of a question as context, we task WizardLM with creating a descriptive caption for the figure provided in the answer. More details on the caption generation procedure can be found in Appendix B.

ArXiv Papers ArXiv[6] is a widely-used open-access archive for scientific articles. As arXiv encourages researchers to upload their papers alongside their source files, it serves as a valuable resource for obtaining Ti*k*Z drawings. Initially, we retrieve all papers with TEX source files and retain those that use the Ti*k*Z package. Subsequently, we expand any include directives and extract all Ti*k*Z environments using regular expressions. To ensure compilability, we additionally preserve all required preamble elements. For that, we first establish a set of rules by analyzing documents obtained from other sources that determine which package imports and configuration options should be retained. We then parse all macro definitions and keep for each Ti*k*Z drawing the macros it uses. Finally, we exclude any Ti*k*Z drawings that fail to compile after this extraction process (around 120k).

Artificial Examples GPT-4 has demonstrated the emergent ability to generate simple, tangible objects (e.g., unicorns) in Ti*k*Z (Bubeck et al., 2023). While not the primary focus of this work, we seek to transfer this ability to our models through knowledge distillation (Bucila et al., 2006). To this end, we compile a diverse set of object descriptions derived from the object categories in the MS COCO, LVIS, and VISOR datasets (Lin et al., 2014; Gupta et al., 2019; Gokhale et al., 2023). Moreover, we sample emoji descriptions from the OpenMoji database.[7] Following this, we instruct GPT-4 to generate a Ti*k*Z drawing for each description, using a chain-of-thought prompt (Wei et al., 2022) we adopt from Zhang et al. (2023b), as detailed in Appendix B.

3.2 Data Augmentation

Prior research indicates a correlation between caption length and caption quality (Gelman et al., 2002; Hartley, 2003; Huang et al., 2023). Notably, Huang et al. (2023) propose a heuristic to judge scientific captions with less than 30 tokens as being of poor quality. Given the recent advancements in MLLM, and notably in MLLM with a focus on science (cf. §2), we propose the automatic augmentation of such captions (Belouadi & Eger, 2023a;b). For an in-depth analysis of implications, refer to Appendix C.

Specifically, we leverage LLaVAR (Zhang et al., 2023c), instructing it to generate short descriptions for Ti*k*Z drawings with captions containing fewer than 30 tokens (cf. Appendix B).[8] Inspired by the CapFilt method (Li et al., 2022a), we generate five candidate descriptions and rank them based on their text-image similarity using CLIPScore (Hessel et al., 2021). The final augmented caption is then formed by concatenating the original caption with the top-ranked description. For GPT-4, we cannot rely on the heuristic, as the captions used are not scientific. Instead, we augment all captions to increase diversity in our dataset while retaining the original captions as well. Table 1 displays the percentage of augmented captions in our dataset. On average, this method increases the CLIPScore of captions with originally fewer than 30 tokens from 24.76 to 29.12, a substantial improvement in text-image similarity, especially considering that CLIPScore typically ranges between zero and 40 (Hessel et al., 2021). The CLIPScore for original captions exceeding 30 tokens is 27.06.

[5] https://texample.net, https://tikz.net, https://pgfplots.net & https://github.com projects
[6] https://arxiv.org
[7] https://openmoji.org
[8] In this work, we use the Moses tokenizer (Koehn et al., 2007) to count tokens.

4 Methods

We leverage LLaMA (Touvron et al., 2023a) as the base model in most experiments, using captions from DaTikZ as model input and TikZ code as ground truths. Since TeX source files from arXiv were included in LLaMA's pre-training data, it may have prior knowledge beneficial for this task. We choose the original LLaMA release over its updated revisions, LLaMA 2 (Touvron et al., 2023b) and CodeLLaMA (Rozière et al., 2023), as their training data is not as clearly specified. This uncertainty and their more recent release would make it difficult to create a test set without training-to-test data leakage. We also experiment with GPT-4 and Claude 2, as earlier research hints at inherent potential for our task (cf. §3.1 and §2), and employ the same chain-of-thought approach outlined in §3.1. However, as they are proprietary, we can only address data leakage for LLaMA (Aiyappa et al., 2023).

4.1 CLiMA

A potential drawback of vanilla LLaMA is that it may not understand visual concepts, given that it was not designed to process image data. However, this ability could significantly enhance the creation of scientific figures. Therefore, we modify LLaMA by combining it with a CLIP ViT-H/14 model (Cherti et al., 2023). CLIP is frequently employed to establish a bridge between vision and natural language, thereby facilitating the creation of MLLMs (Yin et al., 2023).

However, unlike most MLLM methods, to our knowledge we are the first to use CLIP's *multimodal* projection layer, which allows us to extract visual information from both text and images in a common embedding space (cf. Figure 1). This approach is akin to text-to-image models like DALL-E and CLIP-GEN (Wang et al., 2022b), that make use of this duality to generate raster graphics. In our case, our primary objective is to provide LLaMA with a visual interpretation of the input caption, anticipating that this adjustment will boost the alignment with generated TikZ drawings. In addition, it also enables us to experiment with supplying rasterized scientific figures as an additional input (cf. §5). As this new model can be described as using **CL**IP **i**nside LLa**MA**, we refer to it as *CLiMA*.

We accomplish this integration by connecting CLIP's output with LLaMA's input via soft prompting (Lester et al., 2021); i.e., we prepend CLIP's embedding vector to LLaMA's input embeddings of the caption. This requires adding a feed-forward layer with dimensions $\delta_{\text{ViT-H/14}} \times \delta_{\text{LLaMA}}$ to connect image features of dimension $\delta_{\text{ViT-H/14}}$ with LLaMA's word embedding space of dimension δ_{LLaMA}. Following insights from Liu et al. (2023), we pre-train this adaption layer on a dataset of 595K generic text-image pairs for one epoch while keeping both CLIP and LLaMA frozen during the process.

4.2 Error Handling & Correction

A general issue with our language modeling approach to TikZ generation is that outputs may violate the syntactic and semantic rules of TeX, potentially leading to errors and uncompilable documents. While there are constrained decoding algorithms that can force models to form valid programs (Poesia et al., 2022; Scholak et al., 2021), they depend on parse trees and are only useful for languages with a context-free grammar. TeX, however, has a flat, unstructured syntax that is generally impossible to parse (Erdweg & Ostermann, 2010), rendering these methods unsuitable for our approach.

As an alternative, we propose an *iterative resampling* method, leveraging the diagnostic data produced during compilation. If an error arises during compilation, we analyze the logfile to identify its source. Rather than resampling from the start, we then reverse the generation process to just before the error line and continue sampling from there. If the error persists, we infer that the origin of the problem lies earlier in the code and reverse further back, specifically $4^{(i-1)}$ lines above the error, with i denoting the current iteration. While this method does not guarantee error-free results, it provides a more efficient and targeted strategy than simply reinitiating sampling from the beginning.

5 Experiments

Before fine-tuning our models on DaTikZ, we extract a sample of 1k *human-created* items to serve as our test set. As LLaMA's training started in December 2022, we only sample items introduced after this date to avoid data leakage. We conduct both automatic (§5.1) and human evaluations (§5.2). Additional results and instances of generated TikZ drawings are available in Appendices C and E.

Model Sizes In terms of model size, we fine-tune LLaMA$_{7B}$ and CLiMA$_{7B}$, each with 7 billion parameters (7b), as well as LLaMA$_{13B}$ and CLiMA$_{13B}$ with 13 billion parameters (13b), respectively. During inference, we additionally evaluate CLiMA$_{13B}$ with CLIP receiving compiled human-created Ti*k*Z drawings as input instead of captions, which we refer to as CLiMA$_{IMG}$ for clarity (cf. §4.1).

Training Given the size of these models, we introduce trainable low-rank adaption weights (LoRA; Hu et al., 2022) while keeping the base model weights frozen and in half precision (Micikevicius et al., 2018). Following Dettmers et al. (2023), we apply LoRA to all linear layers. In addition, we find that training the embedding layer and language modeling heads is crucial for successful fine-tuning. Since we are not aware of any studies applying LoRA to these layers, we make them fully trainable and leave this investigation to future work. In line with Liu et al. (2023), we train for 12 epochs with AdamW (Loshchilov & Hutter, 2019) and a batch size of 128, but increase the learning rate to 5e−4 as this leads to faster convergence. As a form of data augmentation only possible for CLiMA, we randomly replace the captions forwarded to CLIP with the reference image in 50% of the cases.

5.1 Automatic Evaluation

We use a variety of automatic evaluation metrics to evaluate the performance of our models on our test set in terms of code, image, and caption-image similarity. In particular, we use the following metrics:

CLIPScore calculates the similarity between image and text, as outlined in §3.2. We utilize it to evaluate the correlation between a rasterized Ti*k*Z drawing and its corresponding caption.

CLIPScore$_{IMG}$ is technically the same metric as CLIPScore, but with human-made Ti*k*Z drawings as a reference input. Therefore, it assesses the similarity of two images rather than an image and a caption. To our best knowledge, we are the first to use CLIPScore in this configuration.

Kernel Inception Distance (KID) assesses the quality of generated Ti*k*Z drawings by comparing their distribution with the distribution of real images in the test set (Binkowski et al., 2018). This comparison helps determine how realistic the generated images appear in general. We extract image features using the same CLIP model utilized in CLIPScore.

CrystalBLEU is an n-gram-based metric designed to measure textual similarity (Eghbali & Pradel, 2023). As a variant of BLEU (Papineni et al., 2002), optimized for evaluating code, we employ it to assess the similarity between human-created and machine-produced Ti*k*Z code.

Extended Edit Distance (EED) is a metric dedicated to assessing string similarity (Stanchev et al., 2019), much like CrystalBLEU. We utilize it to determine the minimum number of operations needed to convert the machine-generated code into the reference code.

Compilation Sampling Rate (CSR) measures how frequently we need to sample from a model to yield compilable Ti*k*Z code that outputs an image. This is crucial as some metrics depend on images. With LLaMA and CLiMA, we use iterative resampling (cf. §4.2) and account for partial samples. This is not feasible with GPT-4 and Claude 2 due to their chain-of-thought prompt, which generates code across multiple steps. We take a relaxed stance, counting a sample as successful if it results in an image, even if there are errors.

Results We compute the above metrics for each model and present the system-level scores in Figure 2. The radar chart on the left illustrates that there are small but noticeable score differences between the LLaMA and CLiMA models, revealing some intriguing trends. Aligning with the intuitive expectation that larger models yield better performance (Kaplan et al., 2020), the 13b models clearly outperform the 7b models on all string-similarity and image-based metrics by 0.2–0.5pp (percentage points). A notable exception is CSR, where all models perform comparably. This shows that all models require approximately 1.2 samples per caption to generate a compilable Ti*k*Z drawing.

Within model sizes, CLiMA$_{7B}$ outperforms LLaMA$_{7B}$ on CrystalBLEU, EED, CLIPScore, and CLIPScore$_{IMG}$ by up to 0.4pp, suggesting that, even when only text inputs are involved, integrating CLIP into the model has a predominantly positive effect. CLiMA$_{13B}$ continues this trend, showing a 0.1pp higher CLIPScore than LLaMA$_{13B}$. However, we also see that this does not necessarily have to increase the similarity with a reference image as well, as LLaMA$_{13B}$ has a 0.1pp higher CLIPScore$_{IMG}$. On CrystalBLEU and EED, CLiMA$_{13B}$ again fares better, although, with 0.1pp, the gap is not as pronounced as for the 7b models, possibly due to diminishing returns (Hong et al., 2023).

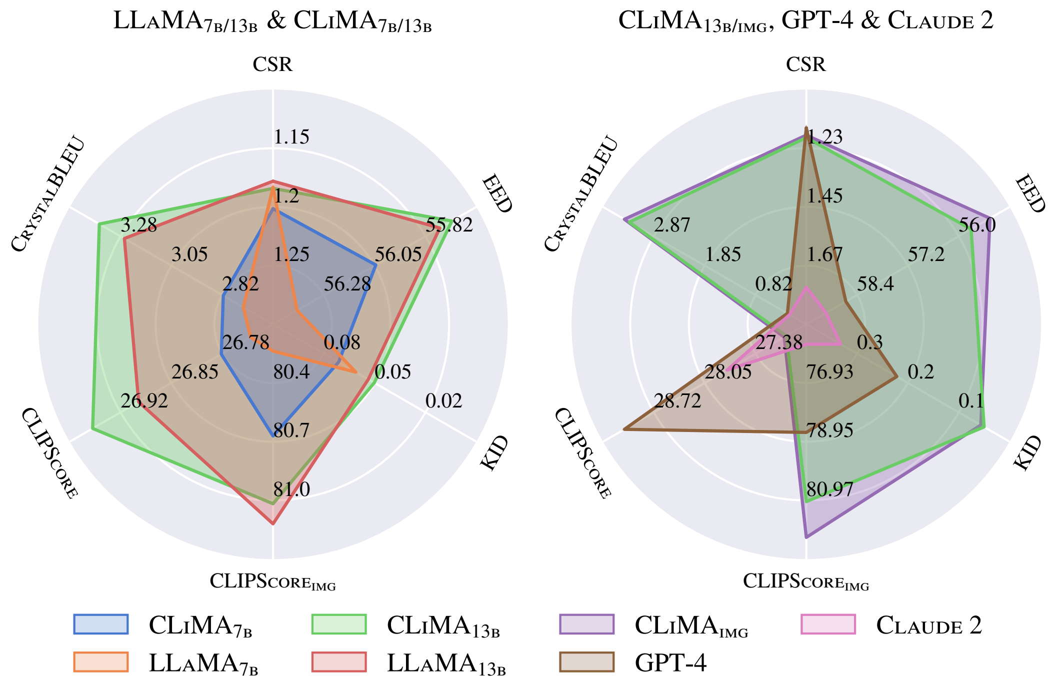


Figure 2: Automatic evaluation results for LLaMA$_{7B/13B}$, CLiMA$_{7B/13B/IMG}$, GPT-4, and Claude 2. Axes representing metrics where lower values are better (CSR, EED, and KID) have been inverted. Detailed scores are provided in Appendix C for further reference.

The right radar chart compares our best text-only model, CLiMA$_{13B}$, with CLiMA$_{IMG}$, GPT-4, and Claude 2. As before, all models perform roughly the same on CSR, except for Claude 2, which needs noticeably more samples. As expected, CLiMA$_{IMG}$, having access to reference images, improves upon CLiMA$_{13B}$ in CLIPScore$_{IMG}$ by 1.2pp. However, this does not lead to an improvement in CLIPScore, echoing our earlier observation that image and caption-image similarity do not always correlate. It also does not improve KID, demonstrating that the overall quality of the images remains constant. Nevertheless, the string-based metrics are 0.1–0.4pp higher, indicating that conditioning on reference images positively impacts code similarity.

We also observe that Claude 2 performs much worse than GPT-4, and both perform noticeably worse than both CLiMA$_{13B}$ and CLiMA$_{IMG}$ on most metrics. The drastically lower CrystalBLEU and EED (up to 3.9pp) suggest that GPT-4 and Claude 2 generate fundamentally different code (in Appendix A we show that it exhibits a lower level of complexity). The up to 6.6pp lower CLIPScore$_{IMG}$ and over six times as large KID indicate that not only do the generated images look different from human ones, but also that the general quality of images is much different from the human distribution. However, most strikingly, both models achieve an up to 2.1pp higher CLIPScore. Upon investigation, we find that both models tend to produce degenerate images, which visibly copy the input caption into the output image. Since the outputs of CLIP (and by extension CLIPScore) can be controlled with *images of text* (Ramesh et al., 2022), Claude 2, and in particular GPT-4, essentially employ such typographic attacks to achieve exceptional caption-image similarities. We further explore this phenomenon in §6.

Overall, CLiMA$_{7B}$ and CLiMA$_{13B}$ outperform their respective LLaMA models in five out of seven metrics each, with Claude 2 and GPT-4 substantially underperforming all of them. While CLiMA$_{IMG}$ unsurprisingly improves upon CLiMA$_{13B}$, CLiMA$_{13B}$ is the best model with only textual inputs.

5.2 Human Evaluation

To further evaluate the effectiveness of our models, we conduct a human annotation campaign using *best-worst scaling* (BWS; Louviere et al., 2015). As a form of comparative annotation, BWS yields high-quality results even when the number of annotations is low (Kiritchenko & Mohammad, 2016;

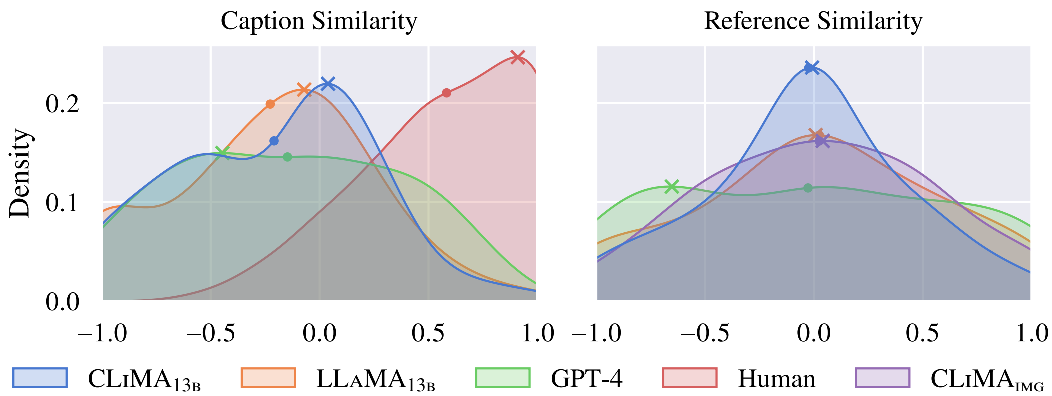


Figure 3: Distributions of BWS scores per model for caption and reference similarity. Scores span from -1 (poor) to 1 (excellent). The "•" markers denote expected values, and "×" signifies the mode.

2017). Within this method, annotators are tasked to compare tuples of $n = 4$ items, identifying the best and the worst item based on a given property. Real-valued scores, ranging from -1 (poor) to 1 (excellent), are then computed by subtracting the fraction of times an item is chosen as the worst from the fraction of times it is chosen as the best (Orme, 2009).

In this work, we focus on *caption similarity* (CS) and *reference similarity* (RS). In CS, annotators evaluate image tuples based on text-image similarity with captions (similar to CLIPSCORE). We construct 4-tuples consisting of our two leading text-only models from automatic evaluation (CLIMA$_{13B}$ and LLAMA$_{13B}$), GPT-4, and human reference images. In RS, the human reference images are used as the standard of comparison instead (similar to CLIPSCORE$_{IMG}$), so we replace them in the tuples with CLIMA$_{IMG}$, while leaving the other models unchanged. Each property is then annotated by four unique expert annotators with relevant research experience (cf. Appendix D).[9] To ensure a manageable workload for the annotators, we create our tuples from a subset of 100 items sampled from our test set. We assess the consistency of the annotators using *split-half reliability* (SHR; Kiritchenko & Mohammad, 2017). This method involves randomly splitting all annotations into two sets, calculating scores for each set, and then determining the correlation between them using Spearman's ρ.

Results For CS, the SHR is $\rho = 0.6$, indicating a moderate but adequate consensus among annotators. Figure 3 (left) exhibits kernel density estimates for the computed scores, with marked modes and expected values. Unsurprisingly, humans perform best with a mode near 1. CLIMA$_{13B}$ is the only other model with a mode above 0, followed by LLAMA$_{13B}$, while GPT-4 lags behind. This indicates that when sampling once with a given caption, CLIMA$_{13B}$ is most likely to generate the best image. Since CLIMA$_{13B}$ and LLAMA$_{13B}$ retain their earlier CLIPSCORE ranking, but GPT-4 drops substantially, we hypothesize that human annotators are not as prone to typographic attacks as metrics. However, we still observe a slight bias towards images of text. In 75% of cases where GPT-4 is selected as the best model, it copies more n-grams from the caption into the image than the worst-ranked image, potentially leading to outliers and thus a slightly higher expected value than CLIMA$_{13B}$ or LLAMA$_{13B}$.

Regarding RS, we record a similar SHR, with $\rho = 0.58$. For LLAMA$_{13B}$, CLIMA$_{13B}$, and CLIMA$_{IMG}$, the distributions in Figure 3 (right) are almost normally distributed, with the mode and expected value being nearly identical. As with CLIPSCORE$_{IMG}$, LLAMA$_{13B}$ is ranked marginally higher than CLIMA$_{13B}$, indicating CLIPSCORE$_{IMG}$ correlates well with human rankings (Appendix C details exact correlations). On a similar scale, CLIMA$_{IMG}$ outperforms LLAMA$_{13B}$. In contrast, GPT-4 follows a nearly uniform distribution, with a slight downward trend for better scores. Therefore, its mode is noticeably lower than for the other models. The expected value, albeit only slightly, is also the lowest.

In summary, our human evaluation aligns well with our automatic metrics, with the added benefit of lower susceptibility to typographic attacks. CLIMA$_{13B}$ outperforms LLAMA$_{13B}$ on CS, while CLIMA$_{IMG}$ surpasses LLAMA$_{13B}$ on RS. GPT-4 shows peculiar distributions, with the mode (and also the expected value for RS) lagging behind, highlighting the effectiveness of our models.

[9] We tried crowdsourcing as well, but due to low agreement with experts, we concluded that crowdworkers lack the necessary expertise for our tasks (cf. Appendix D).

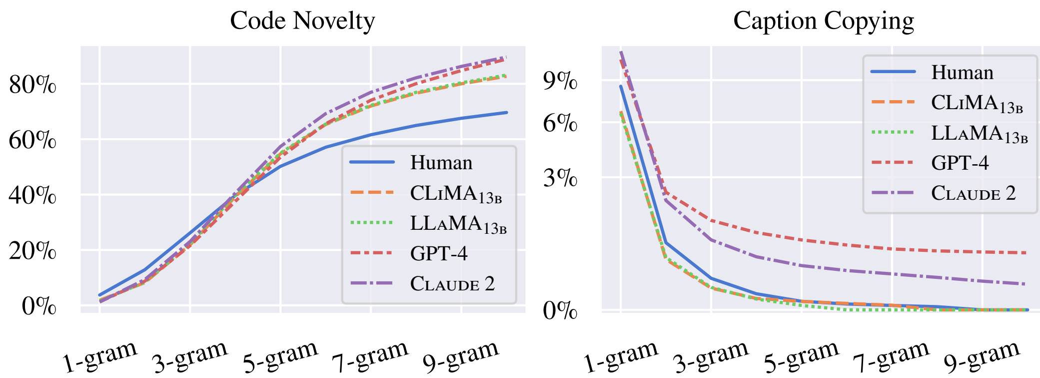


Figure 4: Proportion of unique code n-grams ($n \in [1, 10]$) that do not appear in the training data (left), and proportion of caption n-grams that were copied into the output image (right).

6 Analysis

The issue of language models memorizing and copying training data is a prevalent concern (McCoy et al., 2023; Carlini et al., 2023; Raunak & Menezes, 2022; Meehan et al., 2020). Similarly, we discovered in §5.1 that GPT-4 and Claude 2 tend to perform typographic attacks by memorizing and copying input captions. In this section, we analyze the extent of these issues on our test set using the concept of *n-gram novelty* (McCoy et al., 2023). Specifically, to measure *code novelty*, we determine the proportion of n-grams in the model-generated Ti*k*Z code that are *not* found in the training data. To measure *caption copying*, we calculate the proportion of n-grams from the caption that were copied verbatim into the output code. For comparison, we also calculate both metrics for human references.

Figure 4 displays the results for both metrics ($n \in [1, 10]$) after filtering code comments. In terms of code novelty, models tend to generate less novel code than humans for smaller n-grams ($n < 4$). However, for $n > 6$, models become more novel, with more than 80% of all model n-grams being novel for $n > 8$. McCoy et al. (2023) observe the same phenomenon in all datasets investigated and conclude that this ratio is the normal case when a model is not affected by memorization of the training data. Regarding caption copying, GPT-4 and Claude 2 copy considerably more n-grams from the caption than our models. For 1-grams (i.e., $n = 1$), CLiMA$_{13B}$ and LLaMA$_{13B}$ copy around 6.5% of n-grams, while GPT-4 and Claude 2 copy more than 10%. For $n > 5$, CLiMA$_{13B}$, LLaMA$_{13B}$, and humans practically stop copying, but Claude 2 and especially GPT-4 continue with an almost linear trend, hinting at instances where they might copy the entire caption (cf. Appendix E for examples). This reinforces our hypothesis from §5.1 and points towards CLIPScore$_{IMG}$ as a more robust metric for assessing the visuals of text-rich images since it seems less susceptible to typographic attacks.

7 Conclusion & Future Work

In this work, we present AutomaTi*k*Z, a project for automatically generating Ti*k*Z drawings based on natural language descriptions. As part of AutomaTi*k*Z, we release DaTi*k*Z, a pioneering dataset of aligned Ti*k*Z drawings and captions, and CLiMA, a novel model architecture which integrates multimodal CLIP embeddings into LLaMA. By fine-tuning CLiMA on DaTi*k*Z, we demonstrate that it outperforms LLaMA on several metrics and also surpasses proprietary GPT-4 and Claude 2. In addition, CLiMA can also process images, potentially extending its application to vectorization and conditioning on sketches. Important finds are that (i) integrating CLIP can lead to improvements, even when only text inputs are involved, provided the task relates to visual concepts, and that (ii) attention should be paid to typographic attacks when evaluating models that generate text-rich images.

In future research, we aim to incorporate insights from the caption generation community and enrich our input texts with other figure-mentioning sections of the source documents (cf. §2). We also plan to enhance our extraction pipeline, especially since we had to exclude over 120k Ti*k*Z images from arXiv that failed to compile. We hope that these modifications will bring us a step closer to bridging the gap to human performance.

8 Ethics Statement

We ensure that the Ti*k*Z drawings we gather from online sources are licensed in a manner that permits us to copy and redistribute them. Most sources license their content under a Creative Commons attribution license,[10] the GNU Free Documentation License,[11] or the MIT license.[12] ArXiv is an exception in that, even though it allows licensing under a Creative Commons license, the majority of papers are published under a non-exclusive license, which does not grant us permission to redistribute.[13] As a result, we exclude any Ti*k*Z drawings from arXiv that use this license in the public release of DaTi*k*Z. Nevertheless, we do release AutomaTi*k*Z in conjunction with the dataset generation code, enabling anyone to recreate the full version of DaTi*k*Z themselves. As for auto-generated samples, OpenAI prohibits the use of GPT-4 for creating competing services, restricting this part of our dataset to non-commercial applications.[14]

Apart from our dataset, in this work we compare openly developed models with the proprietary GPT-4 and Claude 2, whose full training details and hyperparameters have not been published. While we strive for a fair evaluation of all models, the lack of transparency in proprietary systems inevitably hinders comparative evaluations and reproducibility.

Within the scope of these limiting aspects, however, our models perform best in generating Ti*k*Z across the tested conditions. Yet, our models should not be used as a substitute for human judgment and critical thinking. They may carry any biases, flaws, or gaps that exist in the base models and training data and could potentially misinterpret the input, fabricate non-existent details, or overlook crucial information. Users should be aware of potential differences between the results they expect and the output the models generate.

Furthermore, while our models are designed to aid in the production of legitimate scientific figures, they could potentially be used to generate disinformation and fake science in the hands of malicious actors.

Acknowledgments

We gratefully thank, in no particular order, Timm Dill, Yanran Chen, Daniil Larionov, JiWoo Kim, Vivian Fresen, Martin Kerscher, Christoph Leiter, and Ran Zhang for their help with our human evaluation campaign, proofreading, discussions, and comments on our work. We further thank the BMBF for its support via the grant Metrics4NLG. The second author is funded under the Excellence Strategy of the German Federal Government and the Länder. The last author is supported by DFG grant EG 375/5–1. We also thank Hugging Face for providing a community GPU grant. Any icons used in this work were designed by OpenMoji, the open-source emoji and icon project.

References

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-yeol Ahn. Can we trust the evaluation on ChatGPT? In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pp. 47–54, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.trustnlp-1.5. URL https://aclanthology.org/2023.trustnlp-1.5.

Anthropic. Model card and evaluations for Claude models, 2023. URL https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf.

Haruka Aoki and Kiyoharu Aizawa. SVG vector font generation for chinese characters with transformer. In *2022 IEEE International Conference on Image Processing, ICIP 2022, Bordeaux, France, 16-19 October 2022*, pp. 646–650. IEEE, 2022. doi: 10.1109/ICIP46576.2022.9897633. URL https://doi.org/10.1109/ICIP46576.2022.9897633.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle, 2022.

[10] https://creativecommons.org/licenses
[11] https://www.gnu.org/licenses/fdl-1.3.en.html
[12] https://opensource.org/license/mit
[13] http://arxiv.org/licenses/nonexclusive-distrib/1.0
[14] https://openai.com/policies/terms-of-use

Jonas Belouadi and Steffen Eger. ByGPT5: End-to-end style-conditioned poetry generation with token-free language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7364–7381, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.406. URL https://aclanthology.org/2023.acl-long.406.

Jonas Belouadi and Steffen Eger. UScore: An effective approach to fully unsupervised evaluation metrics for machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 358–374, Dubrovnik, Croatia, May 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.27. URL https://aclanthology.org/2023.eacl-main.27.

Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4, 2023.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 535–541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

Mu Cai, Zeyi Huang, Yuheng Li, Haohan Wang, and Yong Jae Lee. Leveraging large language models for scalable vector graphics-driven image understanding, 2023.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. DeepSVG: A hierarchical generative network for vector graphics animation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/bcf9d6bd14a2095866ce8c950b702341-Abstract.html.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=TatRHT_1cK.

Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4055–4075. PMLR, 2023. URL https://proceedings.mlr.press/v202/chang23b.html.

Charles Chen, Ruiyi Zhang, Sungchul Kim, Scott Cohen, Tong Yu, Ryan A. Rossi, and Razvan C. Bunescu. Neural caption generation over figures. In Robert Harle, Katayoun Farrahi, and Nicholas D. Lane (eds.), *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, UbiComp/ISWC 2019 Adjunct, London, UK, September 9-13, 2019*, pp. 482–485. ACM, 2019a. doi: 10.1145/3341162.3345601. URL https://doi.org/10.1145/3341162.3345601.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. Figure captioning with reasoning and sequence-level training, 2019b.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan A. Rossi. Figure captioning with relation maps for reasoning. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pp. 1526–1534. IEEE, 2020. doi: 10.1109/WACV45572.2020.9093592. URL https://doi.org/10.1109/WACV45572.2020.9093592.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, June 2023.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning, 2023.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized llms, 2023.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering text-to-image generation via transformers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 19822–19835, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a4d92e2cd541fca87e4620aba658316d-Abstract.html.

Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. CogView2: Faster and better text-to-image generation via hierarchical transformers. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6baec7c4ba0a8734ccbd528a8090cb1f-Abstract-Conference.html.

Aryaz Eghbali and Michael Pradel. CrystalBLEU: Precisely and efficiently measuring the similarity of code. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ASE '22, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394758. doi: 10.1145/3551349.3556903. URL https://doi.org/10.1145/3551349.3556903.

Sebastian Thore Erdweg and Klaus Ostermann. Featherweight TeX and parser correctness. In Brian A. Malloy, Steffen Staab, and Mark van den Brand (eds.), *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pp. 397–416. Springer, 2010. doi: 10.1007/978-3-642-19440-5_26. URL https://doi.org/10.1007/978-3-642-19440-5_26.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12873–12883.

Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01268. URL https://openaccess.thecvf.com/content/CVPR2021/html/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.html.

Kevin Frans, Lisa B. Soros, and Olaf Witkowski. CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/21f76686538a5f06dc431efea5f475f5-Abstract-Conference.html.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474, 2021. doi: 10.1162/tacl_a_00437. URL https://aclanthology.org/2021.tacl-1.87.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Scott Yih, Luke Zettlemoyer, and Mike Lewis. InCoder: A generative model for code infilling and synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=hQwb-lbM6EL.

Andrew Gelman, Cristian Pasarica, and Rahul Dodhia. Let's practice what we preach: Turning tables into graphs. *The American Statistician*, 56(2):121–130, 2002. doi: 10.1198/000313002317572790. URL https://doi.org/10.1198/000313002317572790.

Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation, 2023.

Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5356–5364. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00550. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Gupta_LVIS_A_Dataset_for_Large_Vocabulary_Instance_Segmentation_CVPR_2019_paper.html.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hy6GHpkCW.

James Hartley. Single authors are not alone: Colleagues often help. *Journal of Scholarly Publishing*, 34(2):108–113, 2003. doi: 10.3138/jsp.34.2.108. URL https://doi.org/10.3138/jsp.34.2.108.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595.

John D. Hobby. *MetaPost*, 2014. URL https://tug.org/metapost.

Zhi Hong, Aswathy Ajith, James Pauloski, Eamon Duede, Kyle Chard, and Ian Foster. The diminishing returns of masked language models to science. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1270–1283, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.82. URL https://aclanthology.org/2023.findings-acl.82.

Sameera Horawalavithana, Sai Munikoti, Ian Stewart, and Henry Kvinge. SCITUNE: Aligning large language models with scientific multimodal instructions, 2023.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3258–3264, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.277. URL https://aclanthology.org/2021.findings-emnlp.277.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chieh-Yang Huang, Ting-Yao Hsu, Ryan A. Rossi, Ani Nenkova, Sungchul Kim, Gromit Yeuk-Yin Chan, Eunyee Koh, C. Lee Giles, and Ting-Hao (Kenneth) Huang. Summaries as captions: Generating figure captions for scientific documents with automated text summarization. In C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß (eds.), *Proceedings of the 16th International Natural Language Generation Conference, INLG 2023, Prague, Czechia, September 11 - 15, 2023*, pp. 80–92. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.inlg-main.6.

Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-SVG by abstracting pixel-based diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1911–1920. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00190. URL https://doi.org/10.1109/CVPR52729.2023.00190.

Kushal Kafle, Brian L. Price, Scott Cohen, and Christopher Kanan. DVQA: Understanding data visualizations via question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 5648–5656. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00592. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Kafle_DVQA_Understanding_Data_CVPR_2018_paper.html.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. FigureQA: An annotated figure dataset for visual reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1mz0OyDz.

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 10124–10134. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00976. URL https://doi.org/10.1109/CVPR52729.2023.00976.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

Svetlana Kiritchenko and Saif Mohammad. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 465–470, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2074. URL https://aclanthology.org/P17-2074.

Svetlana Kiritchenko and Saif M. Mohammad. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 811–817, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1095. URL https://aclanthology.org/N16-1095.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-2045.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.243. URL https://doi.org/10.18653/v1/2021.emnlp-main.243.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 11336–11344. AAAI Press, 2020. doi: 10.1609/AAAI.V34I07.6795. URL https://doi.org/10.1609/aaai.v34i07.6795.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 2022a. URL https://proceedings.mlr.press/v162/li22n.html.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. StarCoder: may the source be with you!, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Ré mi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, dec 2022b. doi: 10.1126/science.abq1158. URL https://doi.org/10.1126%2Fscience.abq1158.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=w0H2xGHlkw.

Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 7929–7938. IEEE, 2019. doi: 10.1109/ICCV.2019.00802. URL https://doi.org/10.1109/ICCV.2019.00802.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015. doi: 10.1017/CBO9781107337855.

Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 16293–16302. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01583. URL https://doi.org/10.1109/CVPR52688.2022.01583.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023. doi: 10.1162/tacl_a_00567. URL https://aclanthology.org/2023.tacl-1.38.

Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A three sample hypothesis test for evaluating generative models. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3546–3556. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/meehan20a.html.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

OpenAI. GPT-4 technical report, 2023.

Bryan K. Orme. MaxDiff analysis: Simple counting, individual-level logit, and HB. *Sawtooth Software Research Paper Series*, 2009. URL https://content.sawtoothsoftware.com/assets/8e69929d-a089-4b93-a9f3-d4c64d156642.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pp. 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=KmtVD97J43e.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 2021. URL http://proceedings.mlr.press/v139/ramesh21a.html.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents, 2022.

Vikas Raunak and Arul Menezes. Finding memo: Extractive memorization in constrained sequence generation tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5153–5162, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.378.

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1060–1069. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/reed16.html.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. Searching for COMETINHO: The little metric that could. In Helena Moniz, Lieve Macken, Andrew Rufener, Loïc Barrault, Marta R. Costa-jussà, Christophe Declercq, Maarit Koponen, Ellie Kemp, Spyridon Pilos, Mikel L. Forcada, Carolina Scarton, Joachim Van den Bogaert, Joke Daems, Arda Tezcan, Bram Vanroy, and Margot Fonteyne (eds.), *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pp. 61–70, Ghent, Belgium, June 2022. European Association for Machine Translation. URL https://aclanthology.org/2022.eamt-1.9.

Juan A. Rodriguez, David Vázquez, Issam H. Laradji, Marco Pedersoli, and Pau Rodríguez. FigGen: Text to scientific figure generation. In Krystal Maughan, Rosanne Liu, and Thomas F. Burns (eds.), *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/pdf?id=Hx_iTXnCR5.

Juan A. Rodriguez, David Vázquez, Issam H. Laradji, Marco Pedersoli, and Pau Rodríguez. OCR-VQGAN: Taming text-within-image generation. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pp. 3678–3687. IEEE, 2023b. doi: 10.1109/WACV56688.2023.00368. URL https://doi.org/10.1109/WACV56688.2023.00368.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code LLaMA: Open foundation models for code, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9895–9901, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.779. URL https://aclanthology.org/2021.emnlp-main.779.

Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Kumar Divvala, and Ali Farhadi. FigureSeer: Parsing result-figures in research papers. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, pp. 664–680. Springer, 2016. doi: 10.1007/978-3-319-46478-7_41. URL https://doi.org/10.1007/978-3-319-46478-7_41.

Ashish Singh, Prateek Agarwal, Zixuan Huang, Arpita Singh, Tong Yu, Sungchul Kim, Victor Bursztyn, Nikos Vlassis, and Ryan A. Rossi. FigCaps-HF: A figure-to-caption generative framework and benchmark with human feedback, 2023.

Peter Stanchev, Weiyue Wang, and Hermann Ney. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 514–520, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5359. URL https://aclanthology.org/W19-5359.

Till Tantau. *The TikZ and PGF Packages*, 2023. URL https://github.com/pgf-tikz/pgf.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. LLaMA 2: Open foundation and fine-tuned chat models, 2023b.

Edward Rolf Tufte. *The visual display of quantitative information*. Graphics Press, 1992. ISBN 978-0-9613921-0-9.

Timothy van Zandt. *PSTricks: PostScript macros for Generic TeX*, 2007. URL https://tug.org/PSTricks.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23318–23340. PMLR, 2022a. URL https://proceedings.mlr.press/v162/wang22al.html.

Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. CLIP-GEN: Language-free training of a text-to-image generator with CLIP, 2022b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. IconShop: Text-guided vector icon synthesis with autoregressive transformers, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. WizardLM: Empowering large language models to follow complex instructions, 2023.

Frank F. Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. A systematic evaluation of large language models of code. In Swarat Chaudhuri and Charles Sutton (eds.), *MAPS@PLDI 2022: 6th ACM SIGPLAN International Symposium on Machine Programming, San Diego, CA, USA, 13 June 2022*, pp. 1–10. ACM, 2022. doi: 10.1145/3520312.3534862. URL https://doi.org/10.1145/3520312.3534862.

Zhishen Yang, Raj Dabre, Hideki Tanaka, and Naoaki Okazaki. SciCap+: Knowledge augmented dataset to study the challenges of scientific figure captioning, 2023.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mPLUG-DocOwl: Modularized multimodal large language model for document understanding, 2023.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models, 2023.

Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Wang Yongji, and Jian-Guang Lou. Large language models meet NL2Code: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7443–7464, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.411. URL https://aclanthology.org/2023.acl-long.411.

Han Zhang, Tao Xu, and Hongsheng Li. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 5908–5916. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.629. URL https://doi.org/10.1109/ICCV.2017.629.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023a.

Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with GPT-4, 2023b.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. LLaVAR: Enhanced visual instruction tuning for text-rich image understanding, 2023c.

A Code & Image Complexity

An interesting phenomenon we observe in our experiments in §5 is noticeable differences in Ti*k*Z code size. As shown in Table 2, human-created Ti*k*Z drawings, after filtering comments, contain twice the number of tokens as CLiMA$_{13B}$ and LLaMA$_{13B}$, which in turn have twice as many tokens as GPT-4 and Claude 2. We investigate how this discrepancy is reflected in the compiled images, in particular, whether images created from code with fewer tokens are less complex and whether this also affects image quality. To this matter, we gather another round of BWS scores. Analogous to §5.2, we form tuples with CLiMA$_{13B}$, LLaMA$_{13B}$, GPT-4, and humans. We assign a single annotator to evaluate image tuples in terms of complexity and, due to the potentially subjective nature of the task, four annotators to evaluate image quality. Since we obtain an SHR of $\rho = 0.68$ for image quality, we conclude that all annotators have a similar idea of what constitutes it.

Source	**Tokens**	**Cplx**	**Qlty**
Human	916.56	0.76	0.83
CLiMA$_{13B}$	422.56	0.46	0.4
LLaMA$_{13B}$	420.28	0.46	0.39
GPT-4	186.73	0.31	0.38
Claude 2	179.42	—	—

Table 2: Average number of tokens in Ti*k*Z documents, along with averaged min-max normalized BWS scores for image complexity (Cmplx) and quality (Qlty).

We display averaged and min-max normalized BWS scores in Table 2. Unsurprisingly, there is a clear correlation between code size and image complexity. Humans score over twice as high as GPT-4, with CLiMA$_{13B}$ and LLaMA$_{13B}$ falling in between. A link to quality is also visible, although less pronounced. Humans excel distinctly, but CLiMA$_{13B}$, LLaMA$_{13B}$, and GPT-4 are not far apart. Nonetheless, in absolute terms, CLiMA$_{13B}$ still ranks higher than LLaMA$_{13B}$, and GPT-4 comes in last. Overall, this confirms our observations in §5.1 that GPT-4 and Claude 2 produce code most different from humans, yielding figures that are simpler than other models. Consequently, we believe that our tool can assist humans better in creating more detailed Ti*k*Z drawings.

B Prompt Engineering

Our work involves developing a series of prompts to guide general-purpose language models in accomplishing specific tasks. In this section, we discuss each model and the corresponding prompts we designed. Within a prompt, terms enclosed in curly braces symbolize placeholder variables that are substituted during inference.

WizardLM We employ WizardLM to create captions for TeX Stack Exchange content based on a question and its title (cf. §3.1). We force the generated tokens to start with "`Desired outcome:`" which we subsequently strip from the output, along with any text generated after the first newline:

```
Create a clear and specific caption for an image that depicts the
desired outcome of the following question. Utilize all relevant
details provided in the question, particularly focusing on the
visual aspects.  Avoid referencing any example images or code
snippets. Ensure that your caption is comprehensive and accurate:

{TITLE}

{QUESTION}
```

LLaVAR We use LLaVAR to generate short descriptions for scientific figures (cf. §3.2). We initially tried to provide LLaVAR with the original caption as context to improve it directly, but in many cases this seemed to confuse the model. Therefore, we decided to create short descriptions based solely on the figure and later append them to the original caption. We found that short, concise prompts work best for generating these descriptions:

```
{IMAGE}
Write a short description for the image.
```

Model	CER$_\downarrow$	CSR$_\downarrow$	EED$_\downarrow$	KID$_\downarrow$	CLIP$_{\text{IMG}\uparrow}$	CLIP$_\uparrow$	BLEU$_\uparrow$
LLaMA$_{7B}$	0.644	1.183	56.393	0.059	80.237	26.733	2.732
LLaMA$_{13B}$	0.552	1.178	55.762	0.053	81.12	26.898	3.258
CLiMA$_{7B}$	0.545	1.202	56.045	0.068	80.671	26.776	2.82
CLiMA$_{13B}$	0.656	1.184	55.708	**0.05**	81.017	26.966	3.369
CLiMA$_{\text{IMG}}$	0.658	1.175	**55.264**	0.057	**82.252**	26.994	**3.469**
Claude 2	1.115	1.758	59.135	0.333	75.589	27.753	0.137
GPT-4	**0.384**	**1.147**	58.667	0.222	78.63	**29.115**	0.181

Table 3: Precise system-level scores for automatic evaluation metrics. CLIPScore is abbreviated with CLIP and CrystalBLEU with BLEU. CER shows the average number of compile time errors. Arrows indicate whether larger or smaller scores are better, and the best scores are visually highlighted.

GPT-4 & Claude 2 To generate Ti*k*Z drawings (cf. §3.1, §4, and §5), we use the same prompt for GPT-4 and Claude 2, which was originally designed by Zhang et al. (2023b) and slightly modified for our task. After a comparison with our own prompts and prompts from other works (Bubeck et al., 2023), we found it to work best in our initial experiments. As a chain-of-thought prompt it breaks down the task of code generation into a series of logical steps. By default, the models tend to copy the input caption into the output image. To counter this, we additionally instruct the models not to include the caption in generated images. However, instances of caption copying still occur, as discussed in §6:

```
Draw a TikZ picture for the following caption: {CAPTION}. First,
you need to provide an abstract step-by-step drawing guide. Then,
you need to generate the full code (beginning with "\documentclass"
and ending with "\end{document}") following the guide. Avoid any
direct inclusion of the caption or any lengthy text sequences within
the code. Finally, summarize the drawing.
```

C Additional Details on Experiments

In this section, we conduct additional ablation studies and provide more specific aspects of our experiments in §5. This will provide a clearer understanding of our methodologies and their respective considerations and limitations.

C.1 Training Challenges

During the training phase, we limit the context size of our models to 1200 tokens due to constraints of our existing GPU resources, and filter out all examples that exceed this limit. Although most examples were within these limits, they were generally very close to the maximum allowable length. This prevented us from training our models to perform functions such as multi-turn conversations like chatbots, which could have allowed compiler logs to be fed back into the model to obtain improved outputs. Although we did consider this approach in preliminary experiments with vanilla LLaMA and its instruction-tuned derivatives in a zero-shot setting, the baseline performance of these models was too poor to produce usable Ti*k*Z output. Instead, by drawing parallels to text-to-image generation models that condition the output solely on the input caption and using our iterative resampling technique to improve the outputs, we provide more resource-efficient alternatives with good performance.

C.2 In-depth Evaluation Metric Scores

Table 3 presents the precise system-level scores from our automatic evaluation in §5.1. As observed in that section, CLiMA$_{\text{IMG}}$ and CLiMA$_{13B}$ generally outperform the other models, with CLIPScore being a notable exception due to GPT-4 and Claude 2 being prone to caption copying. The table also contains the Code Error Rate (CER), which indicates the average number of compile time errors of our models. As evidenced, it largely reflects the trends of CSR. Most models generate less than

Training Configuration	CER$_{\downarrow}$	CSR$_{\downarrow}$	EED$_{\downarrow}$	KID$_{\downarrow}$	CLIP$_{\text{IMG}\uparrow}$	CLIP$_{\uparrow}$	BLEU$_{\uparrow}$
Full Training	0.545	1.202	56.045	0.068	80.671	26.776	2.82
−Caption Augmentation	0.603	1.169	56.384	0.056	78.312	25.703	2.714
−Image Training	0.681	1.178	56.336	0.08	80.129	26.604	2.736
−ArXiv Papers (86k)	2.145	1.546	60.552	0.229	75.15	25.328	0.886
−TEX Stack Exchange (29k)	0.624	1.207	56.274	0.071	80.461	26.958	2.936
−Artificial Examples (4k)	0.503	1.194	56.51	0.079	79.912	26.68	2.537
−Curated Examples (1k)	0.656	1.22	56.246	0.072	80.346	26.815	2.711

Table 4: Ablation study results for CLIMA$_{7\text{B}}$ demonstrating how omitting either caption augmentation or randomly forwarding images to CLIP instead of captions during training impacts performance on the test set. The study also evaluates the impact of each data source by training CLIMA$_{7\text{B}}$ on subsets of DATIKZ, omitting one data source each time. Scores lower or higher than those from full training (taken from Table 3) are shaded in red and green, respectively. Color intensity mirrors each score's proportion to the minimum and maximum values of its metric.

one error on average, with CLAUDE 2 being the only exception. Overall, GPT-4 achieves the best score, possibly because it has fewer opportunities to make mistakes by generating shorter and less complex code than other models (cf. §A). Since CLIPSCORE, at a fundamental level, evaluates similar aspects to CS from our human evaluation in §5.2 (caption-image similarity), and CLIPSCORE$_{\text{IMG}}$ is the intuitive counterpart to RS (image similarity), we also calculate their respective Spearman correlations. At the segment-level, the correlation coefficients are 0.23 for CLIPSCORE and CS, and 0.25 for CLIPSCORE$_{\text{IMG}}$ and RS (the degrees of correlation aligning with values commonly found for machine translation metrics; Freitag et al., 2021; Rei et al., 2022). The system-level correlations are 0.5 and 0.8, respectively. The higher correlations witnessed for image over caption-image similarity lends credibility to our hypothesis regarding CLIPSCORE's biases favoring images of text.

C.3 ABLATION STUDIES

In this subsection we delve into ablation studies performed to understand the degree to which data augmentation and the different data sources of DATIKZ contribute to the test set performance of our models.

Impact of Data Augmentation In our work, we incorporate two key data augmentation techniques: we (i) automatically augment image captions in DATIKZ (cf. §3.2), and we (ii) randomly replace image captions forwarded to CLIP with reference images during the training of CLIMA (cf. §5). Even though we show that caption augmentation improves caption-image alignment, we have yet to conclusively quantify its effect on model performance. Similarly, although mixing text-only and text-image modalities is common when training models accepting both unimodal and multimodal input (Li et al., 2020; Wang et al., 2022a), the implications of our particular training strategy remain to be explicitly measured. To investigate these matters, we conduct an ablation study using CLIMA$_{7\text{B}}$ as a representative model for efficiency reasons. In particular, we construct a version of CLIMA$_{7\text{B}}$ that uses DATIKZ without caption augmentation and another version where we train without sampling images. The performance on our evaluation metrics compared to full training results of CLIMA$_{7\text{B}}$ can be seen in Table 4.

Although eliminating caption augmentation reveals subtle improvements for CSR and KID, the model noticeably underperforms for all other metrics, particularly CLIPSCORE (1.1pp lower) and CLIPSCORE$_{\text{IMG}}$ (2.4pp lower). Interestingly, the metrics that show an improvement do not compare model outputs with individual references but either do not use references at all (CSR) or use them only to evaluate general characteristics (KID). On the other hand, metrics with worse results are mostly reference-based. We attribute this observation to the reduced amount of information carried by non-augmented captions, most likely leading to a decline in CLIMA's ability to capture the connection between captions and images and, therefore, delivering lower performance in reference-based metrics. Conversely, with less information that might otherwise constrain the output space, the model is likely to gain more flexibility in producing figures. We conjecture that this allows the model to concentrate more on aspects that contribute positively to metrics without direct references, such as CSR and KID. This hypothesis provides a plausible explanation for our observed results.

Upon examining the impacts of not sampling images during the training process, we find that most metrics exhibit worse performance, with only CSR yielding comparable results. These findings support our choice of training strategy and show that integrating images throughout the training process can bolster performance even when only textual inputs are considered at the evaluation stage.

In summary, we observe that both data augmentation techniques contribute positively to model performance. Caption augmentation enhances the caption-image alignment, benefiting CLiMA$_{7B}$, while leveraging images during training results in improvements as well.

Importance of each Data Source Given the diverse data sources making up DaTikZ (arXiv, TeX Stack Exchange, curated examples, and artificial examples; cf. §3.1), we aim to assess the contribution of each source to the test set performance. This exploration serves multiple purposes: firstly, it enables us to justify the integration of each source; secondly, it can inform future data collection initiatives. To realize this objective, we develop four additional variants of CLiMA$_{7B}$, where each variant is trained by omitting one data source and utilizing only the remaining three. As before, we plot the difference in performance compared to the full training results in Table 4, which should expose the importance of each data source.

The greatest performance decline occurs with the removal of arXiv, a consistent outcome across all metrics. As the primary contributor of scientific figures totaling 86k, its absence instigates a decline of 4.5pp for EED, 5.5pp for CLIPScore$_{IMG}$, 1.4pp for CLIPScore, and 1.9pp for CrystalBLEU. Similarly, CER inflates nearly fourfold, and CSR experiences a noticeable increase, as well. Strikingly, however, results diverge for DaTikZ's second largest source, the TeX Stack Exchange, which provides 29k examples. Contrary to expectations, the general downward trend is comparatively small, with EED and CLIPScore$_{IMG}$ only suffering a 0.2pp decrease and CLIPScore and CrystalBLEU even improving by 0.2pp and 0.1pp, respectively. We attribute this outcome to TeX Stack Exchange's unique focus on providing problem-specific minimum working examples rather than scientific figures like the rest of DaTikZ. Although this could still provide valuable training signals, we did not design our test set (only ~100 of its 1k random samples originate from this source) to evaluate the ability of our models to follow technical advice given in the figure captions. Consequently, we plan to improve our testing framework in future work. Next, although we exclude artificial examples from our test set due to their non-human origin and their removal from the training data improves CER and CSR, their absence still has a negative impact on all other metrics. Without artificial examples, EED drops by 0.5pp, CLIPScore$_{IMG}$ by 0.8pp, CLIPScore by 0.1pp and CrystalBLEU by 0.3pp, showing that distillation of GPT-4 is a useful component in the training process. Lastly, even though curated examples constitute the smallest data source (less than 1k examples), their omission from the training data induces greater negative performance impacts than the removal of TeX Stack Exchange (0.2pp for EED, 0.3pp for CLIPScore$_{IMG}$ and 0.1pp for CrystalBLEU). We attribute this disproportionate value per size to the likely higher quality of code and captions inherent to curated examples.

Overall, if the results are examined independently, we observe that each source has primarily positive effects. However, when evaluated in the context of the source sizes, we find that curated examples create a particularly noticeably positive impact, whereas TeX Stack Exchange exhibits smaller effects. These insights will guide us to focus more on collecting such curated examples in future work.

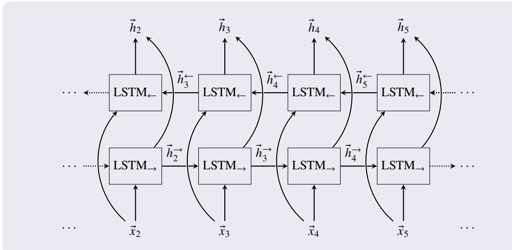
D Annotator Demographics

Our annotators are proficient in English, possessing a C1 level or above as per the Common European Framework of Reference for Languages.[15] They all come from a science and technology background with research experience. They specifically consist of one female and one male faculty member, one female PhD student, three male PhD students, and two female and two male assistants from other institutions.

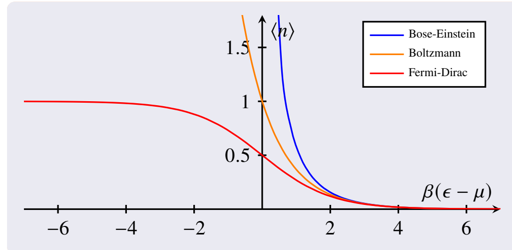
In addition to these expert annotators, we also conducted preliminary tests with crowdworkers via Amazon Mechanical Turk,[16] gathering ten crowd-annotations per tuple for each task. However, the

[15] https://www.coe.int/lang-cefr

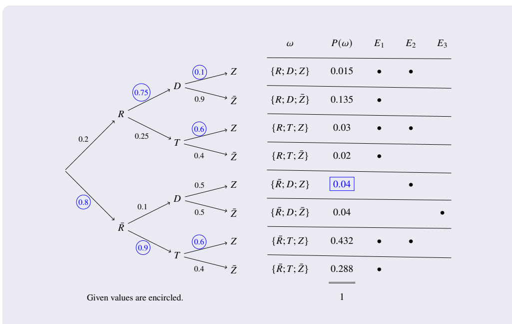
[16] https://www.mturk.com



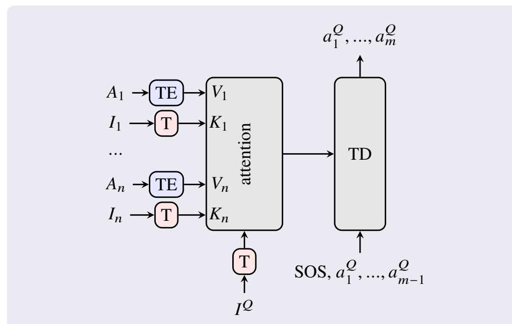
(a) A diagram representing a recurrent neural network consisting of several LSTM blocks, processing the input sequence simultaneously forwards and backwards (to exploit both directions of temporal dependence). Contains some rather tight manoeuvering.



(b) A plot comparing the distribution functions of Bose-Einstein, Boltzmann, and Fermi-Dirac statistics as a function of the reduced chemical potential $\beta(\epsilon - \mu)$. This visualiation highlights the differences between the three types of distribution functions, which are used to describe the behavior of particles in different statistical systems.



(c) Tree with aligned matrix. A probability tree with an aligned matrix listing the possible outcomes, their probabilities and three columns for events described in later tasks. It uses the grahdrawing library and requires LuaLaTeX.



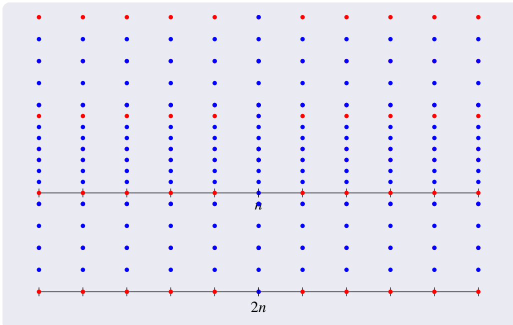
(d) Our approach is a modified version of **meta-seq2seq**. A transformer decoder (TD) is trained to produce a sequence of actions $a_1^Q, \dots, a_m^Q$ given a query instruction I^Q. The context are demonstrations (I_k, A_k) produced by our generative model. We use a transformer encoder-decoder (T) to encode instructions and state S and a transformer encoder (TE) to encode actions. The transformers that process instructions (pink blocks) receive state S as the input of the encoder.

Figure 5: Illustrative examples from the DaTikZ, licensed for free distribution. The examples are from https://github.com/PetarV-/TikZ, https://github.com/janosh/tikz, https://tikz.net, and https://arxiv.org, respectively.

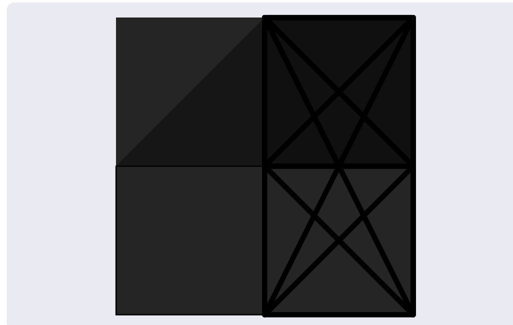
correlation between the results of the crowdworkers and the experts was strikingly low ($\rho < 0.1$). Since the crowdworkers also showed low agreement among themselves, we decided to discontinue further experiments with crowdsourcing.

E Examples

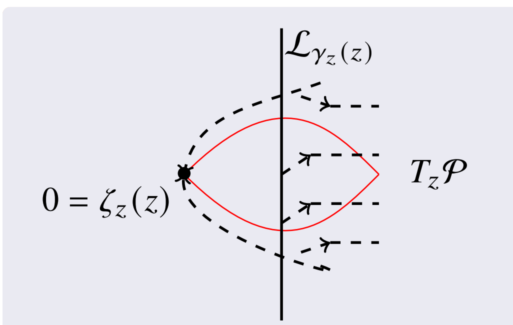
In Figure 5, we provide human-created examples from DaTikZ. Further, Figure 6 contains examples of scientific figures generated by CLiMA$_{13B}$, LLaMA$_{13B}$, and GPT-4 for our human evaluation (cf. §5.2). Each model is represented by one high-scoring and one low-scoring instance, as assessed by our expert annotators in terms of caption similarity. It is worth noting that the low-scoring example of GPT-4 suffers from caption copying. We provide additional examples of caption copying in Figure 7. Instances of generated code can be found in Figure 8.



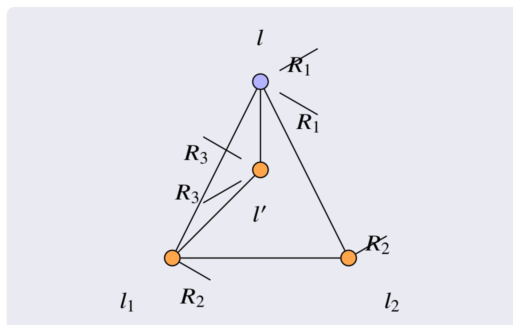
CLiMA$_{13B}$ (good): The ansatz (5.17) and (5.18) for $\alpha = 2$. The red points are evenly spaced, and the blue points scale quadratically with n. *The image is a white background with blue and red dots scattered across it. The dots are placed in various arrangements, creating a visually interesting pattern. Some dots are clustered together, while others are spaced further apart, covering the entire background.*



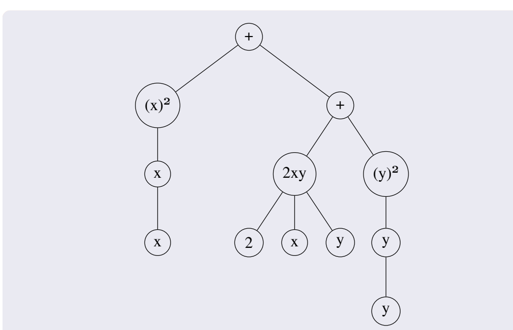
CLiMA$_{13B}$ (bad): For the $P = 2$ scheme, the regular cell footprint is a standard five-point Laplacian, and if any point in the footprint is a cut cell, it is then "irregular." Cut cells are shown with dark shading, irregular cells with light shading, and the remaining white cells are "regular."



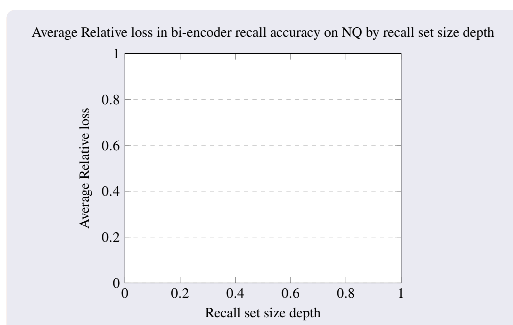
LLaMA$_{13B}$ (good): Local configuration of the path and the foliation (in red) around the point $0 = \Psi_z(z)$. *The image displays a complex mathematical formula with various symbols and notation, including a black dot, waves, and arrows. The formula seems to be related to physics or engineering, and it is written on a white background for easy readability.*



LLaMA$_{13B}$ (bad): Illustration of the proof of Theorem 2.7. *The image displays a tree with blue and orange labels on a white background. The tree has a unique structure, with branches that don't follow a typical tree layout. The labels seem to be representing a formula or a set of instructions, and the tree is accompanied by several equations in the surrounding area.*

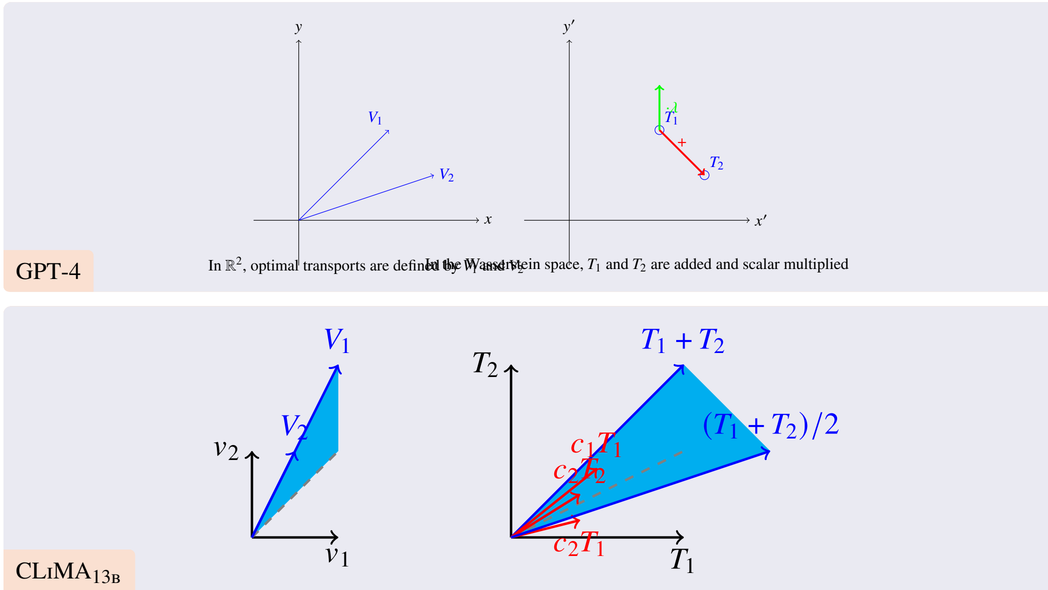


GPT-4 (good): Expression graph for algebraic expression $x^2+2xy+y^2$. *The image shows a tree with a symbol at its root, representing a mathematical concept. The tree has a series of logical connections, and there are variables and mathematical symbols throughout the structure. The image conveys a sense of order and organization in the presentation of the mathematical concept.*

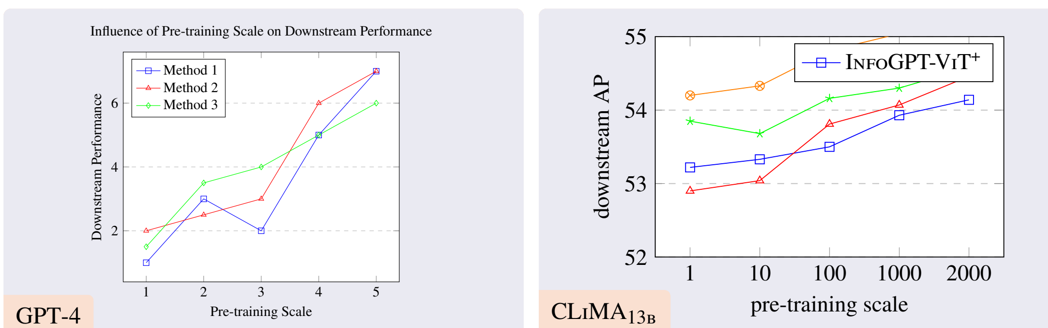


GPT-4 (bad): Average Relative loss in bi-encoder recall accuracy on NQ by recall set size depth on the baseline, Pretrained Alignment (PT), Data Augmentation (DA), and Contrastive Alignment Post Training (CAPOT) on noisy queries.

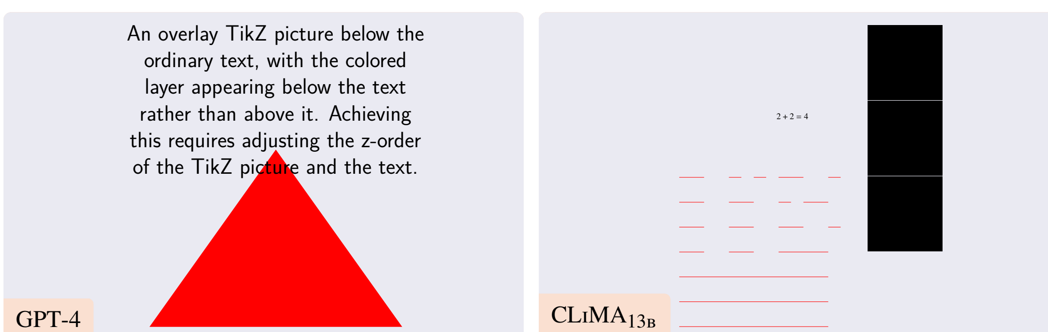
Figure 6: Examples of model-generated scientific figures that received high ratings (good) and low ratings (bad) according to the perception of our expert annotators for caption similarity. The sections of the captions that have been augmented (cf. §3.2) are *emphasized*.



(a) Motivating the definition of the addition ⊕ and scalar multiplication ⊙ operations for $0 < \alpha < 1$ in the Wasserstein optimal transport space for transports T_1, T_2 (right), while in $\mathbb{R}^2$ optimal transports are defined by vectors V_1, V_2 (left).



(b) The influence of pre-training scale on the downstream performance. The value of each method is the average metric values across all datasets in the privacy protection setting with LIGHTGBM.



(c) An overlay Ti*k*Z picture below the ordinary text, with the colored layer appearing below the text rather than above it. Achieving this requires adjusting the z-order of the Ti*k*Z picture and the text.

Figure 7: Additional examples of GPT-4 that suffer from caption copying. For reference, we also provide the output of CLIMA$_{13B}$ for the same captions. The underlying Ti*k*Z code of Figure 7c can be found in Figure 8.

```
\documentclass{article}
\usepackage{tikz}
\pagestyle{empty}

\begin{document}
\sffamily\Huge
\begin{tikzpicture}[overlay, remember picture]
    \node at (current page.center)
    {
        \begin{tikzpicture}
            \fill[red] (0,0) -- (5,7) -- (10,0) -- cycle;
        \end{tikzpicture}
    };
\end{tikzpicture}

\begin{center}
An overlay TikZ picture below the ordinary text, with the colored layer appearing below
the text rather than above it. Achieving this requires adjusting the z-order of the
TikZ picture and the text.
\end{center}
\end{document}
```

GPT-4: An overlay Ti*k*Z picture below the ordinary text, with the colored layer appearing below the text rather than above it. Achieving this requires adjusting the z-order of the Ti*k*Z picture and the text.

```
\documentclass{article}
\usepackage{tikz}
\usepackage{amsmath}
\usetikzlibrary{tikzmark}
\begin{document}
\newcommand{\sep}[1]{\hfill#1\hfill\relax}
\begin{minipage}[t]{.5\linewidth}
\begin{align*}
\sep{2+2=4}
\end{align*}
\end{minipage}%
\begin{minipage}[t]{.5\linewidth}
\tikzmarknode{2}{\rule{3cm}{3cm}}

\tikzmarknode{+}{\rule{3cm}{3cm}}

\tikzmarknode{3}{\rule{3cm}{3cm}}
\end{minipage}
\begin{tikzpicture}[overlay,remember picture]
\draw[red] (-1,-3) -- (5,-3) (-1,-2) -- (5,-2) (-1,-1) -- (5,-1)
(-1,0) -- (0,0) (1,0) -- (2,0) (3,0) -- (5,0)
(-1,1) -- (0,1) (1,1) -- (2,1) (3,1) -- (4,1) (5,1) -- (5.5,1)
(-1,2) -- (0,2) (1,2) -- (2,2) (3,2) -- (3.5,2) (4,2) -- (5,2)
(-1,3) -- (0,3) (1,3) -- (1.5,3) (2,3) -- (2.5,3) (3,3) -- (4,3)
(5,3) -- (5.5,3);
\end{tikzpicture}
\end{document}
```

CLIMA$_{13B}$: An overlay Ti*k*Z picture below the ordinary text, with the colored layer appearing below the text rather than above it. Achieving this requires adjusting the z-order of the Ti*k*Z picture and the text.

Figure 8: The generated code of GPT-4 (top) and CLIMA$_{13B}$ (bottom) from Figure 7c. Both models correctly use the `overlay` option, but they are not devoid of issues. In particular, GPT-4 utilizes nested `tikzpicture` environments which is generally discouraged and the output of CLIMA$_{13B}$ seems particularly verbose.