# TikZero: Zero-Shot Text-Guided Graphics Program Synthesis

Jonas Belouadi* Eddy Ilg† Margret Keuper*,‡ Hideki Tanaka§ Masao Utiyama§
Raj Dabre§ Steffen Eger† Simone Ponzetto*

University of Mannheim, Germany* University of Technology Nuremberg, Germany†
Max Planck Institute for Informatics, Saarland Informatics Campus, Germany‡
National Institute of Information and Communications Technology, Japan§
jonas.belouadi@uni-mannheim.de

## Abstract

*Automatically synthesizing figures from text captions is a compelling capability. However, achieving high geometric precision and editability requires representing figures as graphics programs in languages like TikZ, and aligned training data (i.e., graphics programs with captions) remains scarce. Meanwhile, large amounts of unaligned graphics programs and captioned raster images are more readily available. We reconcile these disparate data sources by presenting TikZero, which decouples graphics program generation from text understanding by using image representations as an intermediary bridge. It enables independent training on graphics programs and captioned images and allows for zero-shot text-guided graphics program synthesis during inference. We show that our method substantially outperforms baselines that can only operate with caption-aligned graphics programs. Furthermore, when leveraging caption-aligned graphics programs as a complementary training signal, TikZero matches or exceeds the performance of much larger models, including commercial systems like GPT-4o. Our code, datasets, and select models are publicly available.*[1]

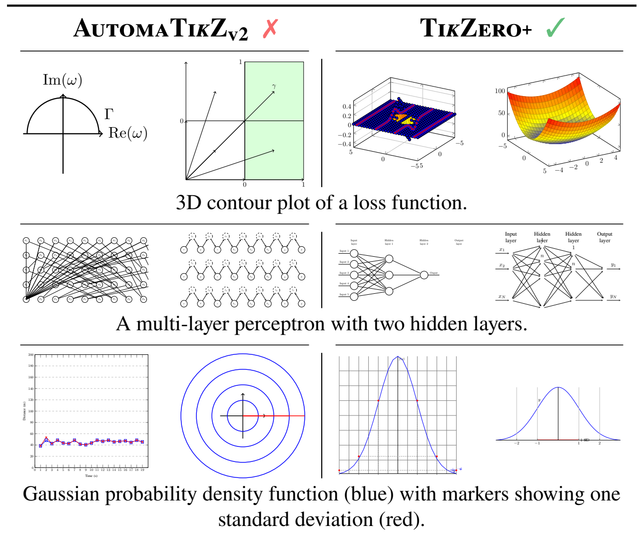


Figure 1. Qualitative comparison of our TikZero+ model (last two columns) and the end-to-end trained baseline AutomaTikZ$_{v2}$ (LLM; first two columns) on text-guided graphics program synthesis with TikZ. Our method generates outputs that more closely follow the given captions. Example program listings are in Supp. Mat. F.

## 1. Introduction

Graphics programming languages offer distinct advantages over low-level vector formats (PDF, SVG) or raster image formats by representing visual concepts as high-level programs that preserve semantics, remain human-interpretable, and allow manual editing. These properties are particularly valuable in academia, where specialized graphics programming languages like TikZ [1] are popular for creating complex figures with high expressivity. However, this comes with a steep learning curve, as seen on the TeX Stack Exchange[2] (TeX.SE), where nearly 10% of questions concern TikZ and make it the most frequently discussed topic on the platform [2, 3].

With recent advances in generative AI, simplifying the creation of graphics programs has become increasingly feasible. Belouadi et al. [2] introduce DeTikZify, an inverse graphics model that generates TikZ programs from images and hand-drawn sketches. However, creating these visual inputs stays cumbersome, motivating alternative input modalities such as natural language. While Belouadi et al. [3] propose AutomaTikZ, a text-guided synthesis model for TikZ programs trained end-to-end on an aligned caption-program corpus, its performance remains limited (cf. Fig. 1) [4, 5].

We identify insufficient training data as the primary limitation. Unlike inverse graphics models such as DeTikZify, which are inherently self-supervised (trained by being conditioned on compiled representations of their output programs) and can access sufficient training data (cf. Fig. 2), end-to-

[1] https://github.com/potamides/DeTikZify
[2] https://tex.stackexchange.com

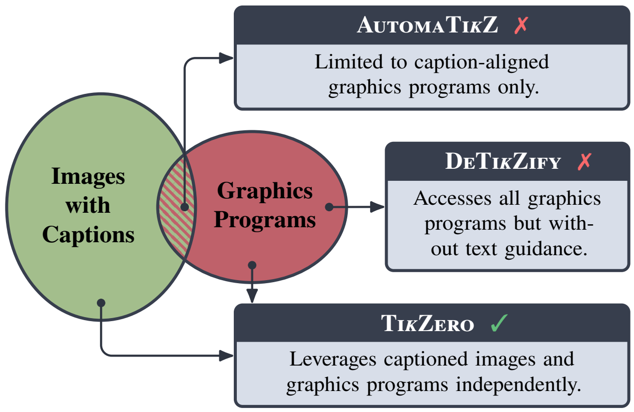


Figure 2. Illustration of training data availability for graphics program synthesis. DeTikZify can leverage all graphics programs for training but lacks text guidance, while AutomaTikZ is constrained to the small intersection of captioned graphics programs, resulting in limited performance. Our approach, TikZero, trains independently on both graphics programs and captioned images, enabling more effective use of available data and yielding superior results.

end text-guided models like AutomaTikZ require graphics programs *paired* with captions, substantially reducing the available data pool (cf. Fig. 2).

To address this challenge, we decouple the graphics program generation component from text understanding, enabling independent training on graphics programs and captioned images *without* requiring paired data (cf. Fig. 2). Our approach first trains an inverse graphics model conditioned on image patch embeddings from a vision encoder [6]. We then train an adapter network that generates synthetic image patch embeddings from captions. This adapter training relies solely on captioned images, effectively circumventing resource limitations and enabling zero-shot (in the sense that no aligned caption-program examples are involved in the training process) text-guided graphics program synthesis [7]. We demonstrate that this approach, to which we refer as TikZero, outperforms previous state-of-the-art methods (cf. Fig. 1). Our key contributions are:

(i) A novel two-stage architecture, TikZero, which addresses the low-resource challenge in text-guided graphics program synthesis by aligning representation spaces rather than relying on aligned data.
(ii) The DaTikZ$_{\text{v3}}$ dataset, comprising over 450k Ti*k*Z graphics programs with roughly 170k captioned samples. Using this dataset, we train both TikZero and AutomaTikZ$_{\text{v2}}$ (an updated version of AutomaTikZ) on the same source data and show that TikZero outperforms AutomaTikZ, AutomaTikZ$_{\text{v2}}$, and other end-to-end trained baselines.
(iii) An enhanced model, TikZero+, combining TikZero with the end-to-end fine-tuning of AutomaTikZ$_{\text{v2}}$, which surpasses larger baselines and matches the performance of commercial models like GPT-4o [8] on key metrics.

## 2. Related Work

**Inverse Graphics Program Synthesis** Inverse graphics, i.e., synthesizing a graphics program to reproduce a visual target, represents a specialized instance of neural program synthesis [9–11]. Deep learning models have shown remarkable success in this domain [12–14], with Vision-Language Models (VLMs) increasingly gaining prominence [2, 15–17]. While controlled experimental studies often rely on synthetic datasets [13, 15, 17–20], real-world applications typically leverage more complex and diverse human-created data [2, 21–25], highlighting the importance of data availability. In scientific contexts, Ti*k*Z has emerged as a popular choice due to its versatility, expressiveness, and widespread adoption in academic circles [2, 21–23, 25]. Although these approaches are not tailored to text-guided generation, we incorporate key elements from them into our approach.

**Text-Guided Graphics Program Synthesis** Current text-guided approaches to graphics program synthesis remain limited, mainly because of the scarcity of captioned graphics programs outlined in Sec. 1, but also because of the difficulty of generating synthetic data with human-like captions [26, 27]. Researchers interested in this capability currently rely on the emerging capabilities of large commercial models such as GPT-4o [8, 28–30], which raises concerns about accessibility, reproducibility, and computational cost [31]. In contrast, related domains like vector graphics generation [24, 32–35] and NL2Vis [36–40] have shown more progress. Similar to inverse graphics, these fields increasingly incorporate large language models (LLMs) [24, 36, 40]. However, vector graphics approaches typically generate only low-level Bézier curves, limiting output complexity [2, 32, 33], and NL2Vis focuses exclusively on data visualization with a restricted set of visualization types [40]. More complex applications, such as generating arbitrary scientific figures from captions with Ti*k*Z, remain underexplored—a gap we address in this work.

**Text-to-Image Generation** TikZero shares conceptual and architectural similarities with several text-to-image generation methods [41–45]. Rodriguez et al. [46, 47] explore generative adversarial networks [48, 49] and diffusion models [50, 51] for scientific figure generation, but these approaches are tied to raster images, which are not ideal for representing scientific figures. Ramesh et al. [43] propose a two-stage model with independently trained prior and decoder components to generate raster images from text. Although prior networks resemble our adapters and have been used with inverse graphics models [52], they target *global* image embeddings containing only abstract information, which degrades performance when used with inverse graphics architectures that work best with *patch-level* details [53]. In contrast, our adapters specifically operate on patch-level embeddings, and we demonstrate that this *improves* performance compared to end-to-end trained baselines.

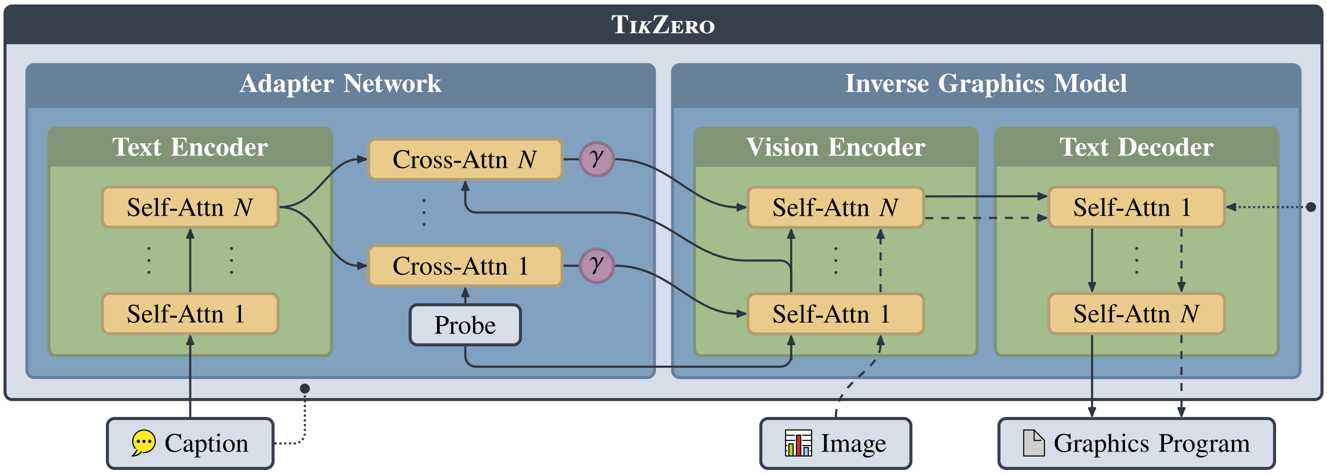


Figure 3. Architecture overview of TikZero during inference. Solid lines represent the standard caption-conditioned path, which flows through the text encoder into the adapter network of TikZero before connecting to the vision encoder. In certain configurations (cf. Sec. 5.2), the caption also feeds into the text decoder (depicted by dotted lines and "•" markers representing shortcuts). The self- and cross-attention layers (yellow) are simplified representations, omitting internal feed-forward layers and residual connections [54]. An exception is the explicit residual connection between the cross-attention and self-attention layers of the vision encoder, visualizing the gating mechanism $\gamma$ (purple). Additionally, the dashed path illustrates how the inverse graphics model generates graphics programs when conditioned on images.

## 3. The TikZero Model & Architecture

As the foundation of TikZero, we first develop a state-of-the-art inverse graphics model for graphics program synthesis. We then incorporate a cross-attention adapter network [55] for text guidance. Fig. 3 provides an overview of our method.

**The Inverse Graphics Model** Due to their demonstrated effectiveness (cf. Sec. 2), we adopt a VLM architecture for the inverse graphics model of TikZero. Figure 3 illustrates its inner workings (dashed lines): the model processes rasterized images and autoregressively generates their corresponding programs without involving captions at this stage.

**The Adapter Network** VLMs consist of two primary components: a vision encoder that produces image embeddings and a text decoder that, in our case, generates graphics programs conditioned on these embeddings. The unidirectional and localized flow of information between these components allows us to inject additional information solely into the vision encoder, thereby influencing the output of the text decoder. We exploit this property by introducing a trainable, text-conditioned adapter network that mimics the outputs of the original vision encoder. This effectively enables zero-shot generation of graphics programs conditioned on text when its outputs are fed into the decoder. In addition to circumventing the resource limitations discussed in Sec. 1, this architecture has other welcome implications: During adapter training, the text decoder, usually the largest component of the model, does not need to be loaded, resulting in efficient and fast training even with large datasets. Our adapter incorporates a lightweight text encoder for embedding captions and introduces newly initialized gated cross-attention layers [56, 57] before each vision encoder layer (cf., Fig. 3). The keys and values derive from the final text encoder representations, while the queries originate from a trainable probe used instead of image inputs. The gates $\gamma$ allow the model to learn at which layers and to what extent information from the text encoder should flow into the vision encoder. Contrary to existing literature, which often employs tanh gates [58] that initialize to zero (indicating no information flow), we find that using sigmoid gates (0.5 at initialization) accelerates training convergence since, in our case, only little information originates from the vision encoder inputs (i.e., the probe). We ablate the gates and the probe in Supp. Mat. A.

**Training Objective** Given a caption-image dataset for training, we first embed the patches $p \in \boldsymbol{p}$ of image $i$ using the unmodified vision encoder $\mathrm{M}$ of our VLM. Subsequently, we incorporate the cross-attention adapter to obtain the modified encoder $\widehat{\mathrm{M}}$, which we then distill on these image patch embeddings conditioned solely on the caption $t$ and probe $\hat{\imath}$ [59]. This leads to the following objective:

$$\mathcal{L}_{\text{dist}} = \frac{1}{|\boldsymbol{p}|}\sum_{p\in\boldsymbol{p}} \operatorname{dist}\Big(\mathrm{M}_{\theta}(\,p \mid i\,), \widehat{\mathrm{M}}_{\theta,\hat{\theta}}(\,p \mid \hat{\imath}, t\,)\Big), \tag{1}$$

where $\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y})$ represents a distance metric. Following common practices in model distillation [60, 61], we experiment with cosine distance and mean squared error. Here, $\theta$ denotes the original model parameters that remain fully frozen, while $\hat{\theta}$ represents the adapter parameters of which the cross-attention layers and the image probe are trainable.

| Source | DaTikZ | DaTikZ$_{v2}$ | DaTikZ$_{v3}$ |
|---|---|---|---|
| curated | 981 | 1 566 | 3 646 |
| TeX.SE | 29 238 | 30 609 | 42 654 |
| arXiv | 85 656 | 326 450 | 407 851 |
| artificial | 1 957 | 1 958 | 2 256 |
| all | 117 832 | 360 583 | 456 469 |

Table 1. Breakdown of the number of unique Ti*k*Z graphics in DaTikZ$_{v3}$ compared to its predecessors DaTikZ and DaTikZ$_{v2}$. Qualitative examples can be found in Supp. Mat. F.

## 4. Datasets & Model Training

We introduce DaTikZ$_{v3}$, a novel dataset of Ti*k*Z graphics programs designed to support the training and evaluation of TikZero. Additionally, we train AutomaTikZ$_{v2}$ as a directly comparable baseline operating on the same data source.

**The DaTikZ$_{v3}$ Dataset** DaTikZ$_{v3}$ expands upon its predecessors DaTikZ and DaTikZ$_{v2}$ [2, 3], incorporating programs from curated repositories, TeX.SE, arXiv papers, and artificial samples (cf. Tab. 1). While previous versions focused exclusively on Ti*k*Z graphics with (v1) or without (v2) captions, DaTikZ$_{v3}$ systematically extracts captions alongside Ti*k*Z graphics whenever possible to support our claims. From over 450k instances, fewer than 170k include captions, underscoring the challenges discussed in Sec. 1.

**Training TikZero** TikZero's VLM builds upon DeTikZify [2] by conditioning a LLaMA-based text decoder [62] on patch embeddings from a SigLIP vision encoder [63]. Specifically, we combine LLaMA$_{3.1}$ (8B) [57] with SigLIP SoViT (0.4B). Unlike DeTikZify and inspired by the continued ViT pretraining approach of InternVL 1.5 [64], we initialize the vision encoder with weights from the fine-tuned encoder of PaliGemma [65] and increase the input resolution to $420 \times 420$ pixels. Furthermore, we fully fine-tune the vision encoder alongside the rest of the model instead of freezing it. We train on DaTikZ$_{v3}$ for 5 epochs with a learning rate of 5e−5 and a batch size of 128. TikZero's VLM consistently outperforms DeTikZify, with detailed evaluation results provided in Supp. Mat. B. For the adapter network, we initialize with LLaMA$_{3.2}$ (1B) as the text encoder [57] and leverage ArxivCap [26], a dataset comprising 6.4 million scientific caption-image pairs for training. The adapter accounts for 2 billion of TikZero's 10 billion total parameters, with only 400 million being trainable. We train for 3 epochs with a learning rate of 1e−4 and a batch size of 512. We emphasize that this two-stage training process *does not* access caption-program pairs. However, we demonstrate that incorporating such aligned data in a subsequent fine-tuning step (Sec. 5.2) further enhances performance.

**Training AutomaTikZ$_{v2}$** Similar to its predecessor, AutomaTikZ$_{v2}$ is a *token-conditioned* LLM that uses *tokenized* captions as conditioning information for graphics prediction (rather than patch embeddings). We initialize AutomaTikZ$_{v2}$ in two different ways: (i) AutomaTikZ$_{v2}$ (LLM), which starts from vanilla LLaMA$_{3.1}$ (8B) weights, and (ii) AutomaTikZ$_{v2}$ (VLM), which leverages TikZero's trained VLM (minus the vision encoder) to benefit from transfer learning [66] on its larger training corpus. Both variants employ the same hyperparameters as TikZero's VLM but can only utilize the caption-annotated subset of DaTikZ$_{v3}$ for training. Despite having access to less caption-aligned data than TikZero's adapter network, AutomaTikZ$_{v2}$ requires a longer training period primarily due to fine-tuning the large decoder (8 billion trainable parameters versus the adapter network's 400 million). Training requires more than two days for AutomaTikZ$_{v2}$ and 1.5 days for TikZero's adapter network when using eight Nvidia A100 40GB GPUs.

## 5. Experiments

Before training models on DaTikZ$_{v3}$, we extract 1k samples from its captioned subset to form our test set. To mitigate data leakage from pretraining to testing, we only include instances created after the cut-off date specified by LLaMA$_{3.2}$ and ArxivCap. We also employ an *n*-gram matching algorithm to avoid cross-contamination with our training split [8]. For all models, the temperature is set to 0.8 and top-p to 0.95. Example outputs are provided in Fig. 1 and Supp. Mat. F.

**Evaluation Metrics** The multimodal nature of our task allows for various evaluation metrics in our automatic evaluations. We assess perceptual *image similarity* between generated outputs and references by computing DreamSim (**DSim**) [67, 68], which correlates highly with human judgments for scientific images [2]. We also calculate the Kernel Inception Distance (**KID**) [69] using SigLIP image features, which evaluates the overall quality of generated figures by comparing to the distribution of reference figures. We evaluate *caption similarity* between generated outputs and reference captions using CLIPScore (**CLIP**) [70] with SigLIP features. To measure *code similarity* between generated and reference Ti*k*Z programs, we use CrystalBLEU (**cBLEU**), a BLEU variant optimized for code evaluation [71, 72], and TeX Edit Distance (**TED**) [2], a variant of the Extended Edit Distance [73] utilizing a TeX tokenizer. Since some metrics require that generated programs compile to images, resampling is necessary if the output contains irrecoverable errors. To quantify this, we compute the *Mean Token Efficiency* (**MTE**), defined as the 10% winsorized mean of the ratio between the number of tokens in the final Ti*k*Z program and the total number of tokens generated to produce that program. For a comprehensive view of model performance, we calculate the arithmetic mean (**AVG**) of *all* previous

| Models | Original Text | | | | | | | Redacted Text | |
|---|---|---|---|---|---|---|---|---|---|
| | DSIM↑ | KID↓ | CLIP↑ | cBLEU↑ | TED↓ | MTE↑ | AVG↑ | CLIP↑ | Ratio↑ |
| IDEFICS 3 (8B) | 45.475 | 11.426 | <u>14.327</u> | 0.656 | 63.175 | 69.558 | 66.628 | 4.851 | 33.858 |
| AUTOMATIKZ (13B) | 46.033 | **1.294** | 3.955 | 0.386 | **62.24** | **85.866** | 63.093 | 2.965 | <u>74.975</u> |
| AUTOMATIKZ$_{v2}$ (VLM) | 38.313 | 33.203 | 0.775 | 0.328 | 76.985 | 21.595 | 0.0 | 0.284 | 36.597 |
| AUTOMATIKZ$_{v2}$ (LLM) | 50.548 | <u>3.491</u> | **15.766** | 0.658 | <u>62.307</u> | 81.775 | 82.375 | <u>8.002</u> | 50.753 |
| TIKZERO (MSE) | <u>52.024</u> | 5.664 | 10.583 | **1.723** | 66.07 | 79.318 | <u>85.004</u> | **8.237** | **77.831** |
| TIKZERO (Cos) | **52.829** | 5.103 | 10.051 | <u>1.603</u> | 65.51 | <u>82.291</u> | **85.599** | 7.226 | 71.893 |

Table 2. System-level scores × 100 for TIKZERO and baselines of comparable size and training setup. Bold and underlined values denote the best and second-best scores for each metric column, respectively. Cell shading illustrates relative score magnitudes. Arrows indicate metric directionality. Overall, TIKZERO achieves the strongest average performance across metrics.

metrics. As these metrics operate on different scales, we apply min-max normalization before computing the average. Additionally, some metrics are recomputed with redacted text in the outputs as part of our analysis, cf. Sec. 6.1.

### 5.1. Comparison against End-to-End Fine-Tuning

In our initial experiment, we evaluate the zero-shot performance of TIKZERO, trained as described in Secs. 3 & 4 using either cosine distance (Cos) or mean squared error (MSE), and compare it against end-to-end trained baselines.

**Baselines** Besides AUTOMATIKZ$_{v2}$ (LLM & VLM), which we designed as directly comparable baselines, we assess other token-conditioned models of similar and slightly larger sizes trained on TikZ. Specifically, we evaluate AUTOMATIKZ (13B),[3] the strongest original AUTOMATIKZ baseline [3], and the general-purpose chatbot IDEFICS 3 (8B) [22]. Additional models and details are available in Supp. Mats. A & C.

**Results** We present the system-level metric scores in Tab. 2 (Original Text). On average, TIKZERO, trained with cosine distance, achieves the best performance with an AVG score of 85.599, closely followed by the MSE variant at 85.004. The next best model, AUTOMATIKZ$_{v2}$ (LLM), scores 82.375, which is 3 percentage points (pp) lower. The remaining models exhibit a substantial performance gap, with IDEFICS 3 (8B) and AUTOMATIKZ (13B) falling behind by approximately 20pp and AUTOMATIKZ$_{v2}$ (VLM) showing the weakest performance across all metrics, resulting in an AVG score of 0. The surprisingly poor results of AUTOMATIKZ$_{v2}$ (VLM) are likely due to catastrophic forgetting [74], as the removal of the vision encoder from TIKZERO's VLM necessitates reacquisition of conditioning based solely on text.

As for individual metrics, our adapter-based models perform particularly well in perceptual image similarity, with TIKZERO (Cos) outperforming the best baseline, AUTOMATIKZ$_{v2}$ (LLM), by 3pp on DREAMSIM. Although AUTOMATIKZ$_{v2}$ (LLM) outperforms TIKZERO by 1.5pp on KID, this indicates in this context that such token-conditioned models (compared to those using patch embeddings) capture the general appearance of scientific figures well but fall short in inferring visual specifics from captions. They do, however, have an edge in reproducing text from captions, which we identify as the primary reason for up to 5pp higher CLIPSCORE, as noted in Sec. 6.1. Regarding code similarity, both TIKZERO models considerably outperform others on cBLEU. Interestingly, we observe a mild inverse correlation between cBLEU and TED. Models conditioned solely on tokenized captions tend to generate shorter, often simplified programs [3], potentially resulting in a reduced edit distance to the reference. In terms of efficiency, all models achieve an MTE of 80–85, indicating that only 2 out of 10 inferences require resampling. TIKZERO (Cos) is 3pp more efficient than MSE, while AUTOMATIKZ (13B), likely benefiting from its larger model size, exceeds it by another 3pp.

In summary, training AUTOMATIKZ$_{v2}$ on top of a VLM yields worse performance than training based on vanilla LLAMA$_{3.1}$, indicating that effective end-to-end training can only leverage the small intersection of graphics programs and images with captions, as illustrated in Fig. 2. However, even without access to this intersection, TIKZERO surpasses both AUTOMATIKZ (13B) and AUTOMATIKZ$_{v2}$ (LLM) on average by being able to train on images with captions independently of graphics programs. Moreover, using a loss function based on cosine distance proves more effective than using MSE.

### 5.2. Combining Adapters with Fine-Tuning

In this section, we investigate whether explicitly incorporating the subset of DATIKZ$_{v3}$ that includes captions into the training process of TIKZERO enhances performance. Our approach involves three incremental stages: (i) We perform a light fine-tuning of TIKZERO (Cos) end-to-end on caption-program pairs for one epoch with a low learning rate of 1e−5. Extending the training duration or increasing the learning rate does not yield further performance gains, likely due to the decoder having already reached its saturation point;

[3] Belouadi et al. [3] refer to this model as CLIMA (13B).

| Models | Original Text | | | | | | | Redacted Text | |
|---|---|---|---|---|---|---|---|---|---|
| | DSim$_\uparrow$ | KID$_\downarrow$ | CLIP$_\uparrow$ | cBLEU$_\uparrow$ | TED$_\downarrow$ | MTE$_\uparrow$ | AVG$_\uparrow$ | CLIP$_\uparrow$ | Ratio$_\uparrow$ |
| Qwen$_{2.5}$ Coder (32B) | 54.473 | 5.493 | 24.87 | 0.285 | 59.856 | 97.269 | 48.593 | 12.164 | 48.911 |
| GPT-4o | **56.464** | 2.844 | **31.787** | 0.327 | **58.511** | **97.675** | 79.019 | **13.32** | 41.905 |
| TikZero (Cos) | 52.829 | 5.103 | 10.051 | 1.603 | 65.51 | 82.291 | 14.658 | 7.226 | **71.893** |
| + Fine-tuning (i) | 53.203 | **1.794** | 10.687 | 0.759 | 61.572 | 94.851 | 46.497 | 6.512 | 60.931 |
| + Separate Captions (ii) | 52.983 | 2.905 | 15.72 | 0.804 | 61.32 | 95.722 | 46.326 | 8.741 | 55.608 |
| + Weight Resetting (iii) | 56.295 | 1.831 | 24.177 | **1.988** | 59.008 | 93.058 | **87.043** | 11.479 | 47.478 |

Table 3. System-level scores × 100 for additional baselines and TikZero combined with fine-tuning and token-conditioning. The scores for TikZero (Cos) are replicated from Tab. 2 for convenience. Bold and underlined values denote the best and second-best scores for each metric column, respectively. Cell shading illustrates relative score magnitudes. Arrows indicate metric directionality. Overall, TikZero (Cos) with Weight Resetting (iii) demonstrates the strongest average performance across metrics.

(ii) Alongside feeding captions into the adapter, we provide them separately to the text decoder in tokenized form (cf., Fig. 3); (iii) Prior to fine-tuning, we reset the decoder to its initial weights to overcome saturation, enabling us to fine-tune using the setup described in Sec. 4, which involves 5 epochs and a learning rate of 5e−5.

**Baselines** In addition to the baselines in Tab. 2, which remain comparable, we also evaluate larger and commercial models that serve as stronger baselines (cf. Supp. Mat. C). Specifically, we assess GPT-4o [8], which has demonstrated strong performance in generating Ti*k*Z [3, 28, 30] and Qwen$_{2.5}$ Coder (32B) [75] as an open-weights model.

**Results** In Tab. 3 (Original Text), all fine-tuning setups of TikZero show considerable improvement over the base version. Approaches (i) and (ii) each enhance performance by over 30pp on AVG, while approach (iii) surpasses them with an improvement of over 70pp, positioning it as the best-performing model on average, even when compared to our new baselines, with GPT-4o being 8pp lower and Qwen$_{2.5}$ Coder (32B) approximately 40pp lower. Approach (i) demonstrates that direct fine-tuning yields positive effects across nearly all metrics, notably improving MTE by 12pp, TED by 4pp, and KID by 3.5pp. Approach (ii) shows similar trends but, by also incorporating tokenized captions, further improves CLIPScore by 5pp, closing the gap to AutomaTikZ$_{v2}$ (LLM). Interestingly, both (i) and (ii) slightly decrease performance on cBLEU, potentially due to similar reasons discussed in Sec. 5.1. However, the same cannot be said for (iii), which not only achieves the highest score on cBLEU but also ranks as the second-best on TED, trailing only 0.5pp behind GPT-4o and showcasing that it is possible to perform well on both metrics. Additionally, it increases DreamSim by another 3pp and CLIPScore by 8.5pp, competing with the much stronger baselines Qwen$_{2.5}$ Coder (32B) and GPT-4o. In KID, it even surpasses them by 3.5pp and 1pp, respectively.

In summary, fine-tuning TikZero, especially when combined with a separate caption input and weight resetting, greatly improves performance. This illustrates that the intersection of graphics programs and images with captions, though small, provides a valuable training signal, and best performance can be achieved by making full use of both sets. The best-performing TikZero model even competes with and often surpasses Qwen$_{2.5}$ Coder (32B) and GPT-4o on several key metrics. Notably, the former model is more than three times larger, and the latter is often estimated at around 1.8 trillion parameters [76], making it 180 times larger.

## 5.3. Human Evaluation

To corroborate our findings from automatic evaluation, we conduct a human annotation campaign focusing on two key properties: caption and image similarity. We employ *Best-Worst Scaling* (BWS) [77], a comparative annotation method that yields high-quality results even with few annotators [78, 79]. We sample 100 instances from our test set and present annotators with $n$-tuples of generated figures, asking them to identify the most and least similar figure to either the reference caption or reference image. This data is then transformed into scores from -1 (poor) to 1 (excellent) by subtracting the proportion of times a figure is selected as the best from the proportion of times it is chosen as the worst [80]. For a manageable workload, we focus on $n = 4$ key models: TikZero (Cos), our best-performing model from Sec. 5.1; AutomaTikZ$_{v2}$ (LLM), its direct end-to-end trained competitor; GPT-4o, our strongest baseline; and TikZero (Cos) fine-tuned using approach (iii) from Sec. 5.2, our best model overall, henceforth referred to as TikZero+ for convenience. We engage thirteen annotators and obtain six fully annotated sets per task (cf. Supp. Mat. E for more details). To assess annotator consistency, we calculate the *split-half reliability* (SHR) [79]. This method randomly divides all annotations into two sets, calculates scores independently, and then determines their correlation using Spearman's $\rho$.

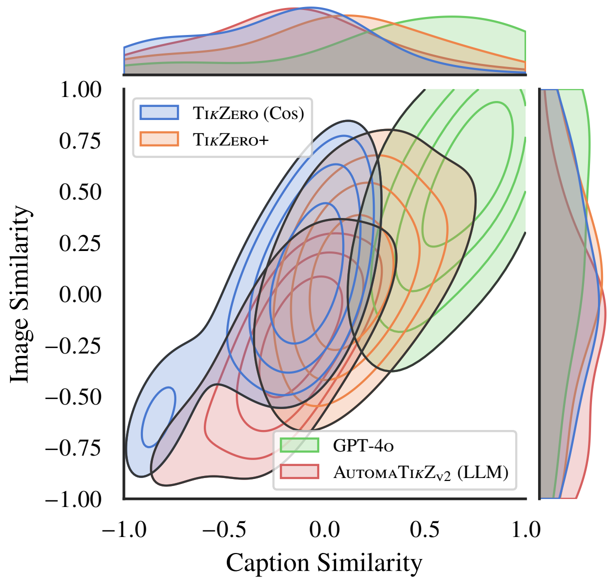


Figure 4. Bivariate distributions of BWS scores (higher is better) using kernel density estimation for caption and image similarity. Along the diagonal, TikZero (Cos) achieves higher scores than AutomaTikZ$_{v2}$ (LLM), while TikZero+ and GPT-4o demonstrate superior performance compared to both.

**Results** Fig. 4 presents kernel density estimates for the BWS scores, showing generally consistent rankings with automatic evaluations but revealing notable differences in the magnitude of gaps. For caption similarity, the ranking aligns with CLIPScore evaluations ($\rho = 1.0$), with TikZero (Cos), AutomaTikZ$_{v2}$ (LLM), TikZero+, and GPT-4o achieving mean scores $\mu$ of -0.25, -0.18, 0.03, and 0.4, respectively. Interestingly, humans perceive a 40% smaller gap between AutomaTikZ$_{v2}$ (LLM) and TikZero (Cos) than suggested by CLIPScore values, where AutomaTikZ$_{v2}$ (LLM) outperforms TikZero (Cos) by 50%. This indicates humans may evaluate caption similarity differently than CLIPScore (cf. Sec. 6.1). For image similarity, the system order remains consistent with our DreamSim metric ($\rho = 1.0$), with AutomaTikZ$_{v2}$ (LLM), TikZero (Cos), TikZero+, and GPT-4o achieving $\mu$ of -0.26, -0.02, 0.01, and 0.27, respectively. However, the relative gaps between models differ: the separation between AutomaTikZ$_{v2}$ (LLM) and TikZero (Cos), as well as between TikZero+ and GPT-4o, appear more pronounced than observed with DreamSim. This discrepancy likely stems from BWS capturing relative preferences rather than absolute performance differences. GPT-4o is selected 20% more often as the best model than TikZero+, and AutomaTikZ$_{v2}$ (LLM) 15% more often as the worst model than TikZero (Cos), creating larger perceived gaps even when qualitative differences may be subtle.

The SHR values of 0.68 for caption similarity and 0.76 for image similarity indicate moderate to strong inter-annotator agreement. We also observe a correlation between these two tasks, with segment-level $\rho = 0.62$ and system-level $\rho = 0.8$, suggesting that both evaluation dimensions capture related aspects of model performance. GPT-4o emerges as the best-performing model, aligning with its superior performance on the corresponding automatic metrics, CLIPScore and DSim. Among open-source models, TikZero+ performs best, while AutomaTikZ$_{v2}$ (LLM) ranks lowest overall.

## 6. Analysis

We present a comprehensive analysis, investigating the influence of typographic attacks on CLIPScore and examining the effectiveness of our architecture in low-resource settings, both in terms of training data and trainable parameters.

### 6.1. CLIPScore Limitations & Typographic Attacks

A known limitation of CLIPScore with text-rich images is its susceptibility to typographic attacks, where scores are disproportionately influenced by string similarity between images and captions [3, 43]. We suspect that token-conditioned models like AutomaTikZ$_{v2}$ (LLM) achieve higher CLIPScore values than models such as TikZero (Cos & MSE) primarily because they tend to visibly copy more substrings from the caption in the output image. To test this hypothesis, we apply the ROT13 substitution cipher [81] to all visible strings in the generated figures and recompute CLIPScore. This basic cipher replaces each letter with the 13th letter after it in the Latin alphabet. While not cryptographically secure, the ratio between the original and recomputed CLIPScore values should indicate the influence of string matching, i.e., higher ratios suggest less copied text and vice versa.

Tabs. 2 & 3 (Redacted Text) present the recomputed CLIPScore values and ratios for all evaluated models. The results reveal that most TikZero models, except for fine-tuning approaches (ii) and (iii), which also condition on tokenized captions, have considerably higher ratios (61%–78%) compared to strictly token-conditioned models (34%–51%), supporting our hypothesis. AutomaTikZ (13B) is an exception, possibly due to its initially low score. Further analysis shows that with redacted text, AutomaTikZ$_{v2}$ (LLM)'s CLIPScore performance drops to the same level as TikZero (Cos & MSE), suggesting that string matching is the primary factor in its superior performance rather than producing better visuals—arguably a more difficult task. Nevertheless, reproducing strings is still somewhat desirable. The human oracle of our test set achieves a ratio of 50.8%, close to the 47.5% of TikZero+, our best-performing model. In contrast, models like GPT-4o, with a lower ratio of 41.9%, may overfit to caption copying, artificially inflating the CLIPScore values.

### 6.2. Low-Resource Training

While our adapters train efficiently on large-scale datasets, we investigate whether such extensive data is necessary for opti-

| | Training Data | | | |
|---|---|---|---|---|
| **Intv.** | 100% | 50% | 25% | 12.5% |
| 1 | **92.411** | 77.478 | 49.967 | 56.055 |
| 2 | <u>87.557</u> | 85.249 | 54.942 | 33.817 |
| 4 | 82.254 | 47.381 | 32.12 | 37.914 |
| 8 | 76.545 | 40.816 | 29.774 | 16.25 |

Table 4. AVG scores for TikZero (Cos) trained on varying fractions of data and intervals of cross-attention layers. Higher scores indicate better performance. Bold and underlined values denote the best and second-best scores for the whole table, respectively. Cell shading illustrates score magnitudes.

mal performance. Along the same vein, we examine the impact of reducing the amount of cross-attention layers inserted into the vision encoder. We retrain TikZero (Cos) using varying fractions of the training data ($d \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$) and insert cross-attention layers at different intervals ($i \in \{1, 2, 4, 8\}$). Table 4 presents the AVG scores from this parameter grid, with detailed scores in Supp. Mat. D. Our findings reveal that utilizing the full dataset and inserting cross-attention at every layer yields the highest average performance, highlighting the benefits of maximizing both variables. Interestingly, the model's performance appears more robust to a reduction in the number of layers compared to a decrease in training data. For instance, training on only $\frac{1}{8}$th of the data leads to a substantial performance drop of 36pp, whereas inserting cross-attention layers every 8 layers (resulting in only 3 cross-attention layers in total) causes a more modest decline of 16pp. Minimizing both variables leads to the most severe drop of over 75pp. These results validate our training setup while suggesting that incorporating additional data might further enhance performance. Given that ArxivCap extracts figures from only 572k papers, whereas some corpora index over 200 million papers [82], there remains a lot of potential for leveraging larger datasets in future work.

## 7. Conclusion

In this work, we demonstrate the potential of TikZero and its variants for generating TikZ graphics programs from captions. Notably, TikZero does not require aligned caption-program pairs in its original formulation but instead aligns representation spaces of unaligned graphics programs and captioned images. This enables our model to leverage substantially more training data compared to end-to-end trained models that operate solely on caption-image pairs (cf. Fig. 2) while maintaining training efficiency. TikZero outperforms strong end-to-end trained baselines, including our independently trained AutomaTikZ$_{v2}$ models, which use the same data pool, excluding instances they cannot process, illustrating the strengths of our approach. When extending the TikZero approach with additional end-to-end training, it also compares favorably to much larger baselines and commercial systems like GPT-4o. While this enhanced approach, TikZero+, is no longer zero-shot by our definition, it remains a TikZero model in the sense that it operates on both sets of graphics programs and captioned images, with the added advantage of explicitly utilizing their intersection (cf. Fig. 2).

These results demonstrate the benefits of designing architectures around available data and validate the approach of decoupling graphics program generation from text understanding (with optional later reconciliation through TikZero+). Although we demonstrate our method specifically on TikZ, we believe its general principles will inspire future work on related graphics program synthesis tasks.

**Future Work** Beyond scaling up our training data to explore convergence limits (cf. Sec. 6.2), we plan to investigate automatic methods for improving the quality and alignment of caption-image or caption-program pairs. This includes rewriting potentially noisy captions with LLMs and enhancing them with the visual understanding capabilities of VLMs [83–85]. We believe our approach to aligning textual and image modalities enables other promising applications for graphics program synthesis, such as editing images in latent space via textual instructions to generate modified graphics programs. Additionally, we intend to explore alternative alignment strategies beyond model distillation, including contrastive learning [86], which has successfully aligned modalities in discriminative models [6, 87, 88].

## Limitations

Our evaluations include proprietary systems that operate as black boxes; their training data is unknown, and they offer no guarantees of consistent performance over time. This (i) makes addressing data leakage and cross-contamination impossible and (ii) limits the fairness and reproducibility of our experiments. Nevertheless, even under these unfavorable conditions, our open models remain competitive. Users should be aware, however, that our models may behave unpredictably, and outputs might differ from expectations. Additionally, our models do not include safeguards against potential misuse, e.g., for generating fake scientific content.

Regarding licensing of our training data, a large portion of the TikZ programs in DaTikZ$_{v3}$ are licensed under permissive terms[4] that allow redistribution. The remaining programs are distributed under the arXiv.org perpetual, non-exclusive license, which prohibits redistribution, which is why we exclude them from the public release of DaTikZ$_{v3}$. However, since we release our dataset creation scripts, we encourage others to reproduce the full version independently.

[4] https://creativecommons.org/licenses; https://opensource.org/license/mit; https://www.gnu.org/licenses/fdl-1.3.en.html; https://openai.com/policies/terms-of-use

## Acknowledgments

We extend our sincere gratitude to the following individuals (in no particular order) for their valuable contributions: Christian Greisinger, Hour Kaing, Ran Zhang, Tejaswini Medi, Yanran Chen, Sotaro Takeshita, Katharina Prasse, JiWoo Kim, Christoph Leiter, Haiyue Song, and Aida Kostikova. Their assistance with our human evaluation campaign, proofreading, insightful discussions, and constructive feedback has been instrumental to our work. The first author conducted part of this research during an internship at the National Institute of Information and Communications Technology (NICT), Japan. The second to last author is supported by the Federal Ministry of Education and Research (BMBF) via the research grant METRICS4NLG and the German Research Foundation (DFG) via the Heisenberg Grant EG 375/5–1. We acknowledge computing resources provided by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597–1 FUGG. Finally, we thank the OpenMoji project for the open-source icons used throughout this work.

## References

[1] Till Tantau. *The TikZ and PGF Packages*, 2023. 1

[2] Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. DeTikZify: Synthesizing graphics programs for scientific figures and sketches with TikZ. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2, 4, 15

[3] Jonas Belouadi, Anne Lauscher, and Steffen Eger. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 4, 5, 6, 7, 16, 18

[4] Leixin Zhang, Yinjie Cheng, Weihe Zhai, Steffen Eger, Jonas Belouadi, Fahimeh Moafian, and Zhixue Zhao. ScImage: How good are multimodal large language models at scientific text-to-image generation? In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 15

[5] Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. DiagrammerGPT: Generating open-domain, open-platform diagrams via LLM planning. In *First Conference on Language Modeling*, 2024. 1

[6] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 8

[7] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009. 2

[8] OpenAI. GPT-4 technical report, 2023. 2, 4, 6

[9] Emilio Parisotto, Abdel rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. Neuro-symbolic program synthesis. In *International Conference on Learning Representations*, 2017. 2

[10] Jacob Devlin, Jonathan Uesato, Surya Bhupatiraju, Rishabh Singh, Abdel rahman Mohamed, and Pushmeet Kohli. RobustFill: Neural program learning under noisy I/O. In *Proceedings of the 34th International Conference on Machine Learning*, pages 990–998. PMLR, 2017.

[11] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, page 835–850, New York, NY, USA, 2021. Association for Computing Machinery. 2

[12] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1666–1675. PMLR, 2018. 2

[13] Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Josh Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *Thirty-second Conference on Neural Information Processing Systems*, pages 6062–6071, 2018. 2

[14] Kevin Ellis, Maxwell Nye, Yewen Pu, Felix Sosa, Josh Tenenbaum, and Armando Solar-Lezama. Write, execute, assess: Program synthesis with a REPL. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[15] Peter Kulits, Haiwen Feng, Weiyang Liu, Victoria Fernandez Abrevaya, and Michael J. Black. Re-thinking inverse graphics with large language models. *Transactions on Machine Learning Research*, 2024. 2

[16] Wen-Ding Li and Kevin Ellis. Is programming by example solved by LLMs? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[17] Shreyas Kapur, Erik Jenner, and Stuart Russell. Diffusion on syntax trees for program synthesis. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[18] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. CSGNet: Neural shape parser for constructive solid geometry. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5515–5523. Computer Vision Foundation / IEEE Computer Society, 2018.

[19] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Learning to infer and execute 3D shape programs. In *International Conference on Learning Representations*, 2019.

[20] Javier Cámara, Javier Troya, Lola Burgueño, and Antonio Vallecillo. On the assessment of generative AI in modeling tasks: an experience report with chatgpt and UML. *Softw. Syst. Model.*, 22(3):781–793, 2023. 2

[21] Hugo Laurençon, Leo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[22] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions, 2024. 5

[23] Shengbang Tong, Ellis L Brown II, Penghao Wu, Sanghyun Woo, Adithya Jairam Iyer, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[24] Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. StarVector: Generating scalable vector graphics code from images and text. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 16175–16186, 2025. 2

[25] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, Sam Dodge, Keen You, Zhen Yang, Aleksei Timofeev, Mingze Xu, Hong-You Chen, Jean-Philippe Fauconnier, Zhengfeng Lai, Haoxuan You, Zirui Wang, Afshin Dehghan, Peter Grasch, and Yinfei Yang. MM1.5: Methods, analysis & insights from multimodal LLM fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[26] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14369–14387, Bangkok, Thailand, 2024. Association for Computational Linguistics. 2, 4

[27] Jaeyoung Kim, Jongho Lee, Hong-Jun Choi, Ting-Yao Hsu, Chieh-Yang Huang, Sungchul Kim, Ryan Rossi, Tong Yu, Clyde Lee Giles, Ting-Hao 'Kenneth' Huang, and Sungchul Choi. Multi-LLM collaborative caption generation in scientific documents. In *AI for Research and Scalable, Efficient Systems*, pages 142–160, Singapore, 2025. Springer Nature Singapore. 2

[28] 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. 2, 6

[29] Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14410–14419, 2024.

[30] Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with GPT-4, 2023. 2, 6

[31] Lingjiao Chen, Matei Zaharia, and James Zou. How Is ChatGPT's Behavior Changing Over Time? *Harvard Data Science Review*, 6(2), 2024. https://hdsr.mitpress.mit.edu/pub/y95zitmz. 2

[32] Sagi Polaczek, Yuval Alaluf, Elad Richardson, Yael Vinker, and Daniel Cohen-Or. NeuralSVG: An implicit representation for text-to-vector generation, 2025. 2

[33] Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. IconShop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Trans. Graph.*, 42(6), 2023. 2

[34] Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-SVG by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1911–1920, 2023.

[35] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders. In *NeurIPS*, 2022. 2

[36] Henrik Voigt, Kai Lawonn, and Sina Zarrieß. Plots made quickly: An efficient approach for generating visualizations from natural language queries. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12787–12793, Torino, Italia, 2024. ELRA and ICCL. 2

[37] Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. Synthesizing natural language to visualization (nl2vis) benchmarks from NL2SQL benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*, page 1235–1247, New York, NY, USA, 2021. Association for Computing Machinery.

[38] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2): 110–141, 1986.

[39] Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 112–117, New York, NY, USA, 1994. Association for Computing Machinery.

[40] Yang Wu, Yao Wan, Hongyu Zhang, Yulei Sui, Wucai Wei, Wei Zhao, Guandong Xu, and Hai Jin. Automated data visualization from natural language via large language models: An exploratory study. *Proc. ACM Manag. Data*, 2(3), 2024. 2

[41] 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*, pages 10674–10685. IEEE, 2022. 2

[42] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents, 2022. 2, 7

[44] 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 *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19822–19835, 2021.

[45] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. CogView2: Faster and better text-to-image generation via hierarchical transformers. In *NeurIPS*, 2022. 2

[46] Juan A. Rodriguez, David Vázquez, Issam H. Laradji, Marco Pedersoli, and Pau Rodríguez. FigGen: Text to scientific figure generation. In *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*. OpenReview.net, 2023. 2

[47] 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*, pages 3678–3687. IEEE, 2023. 2

[48] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020. 2

[49] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, 2021. 2

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, Lille, France, 2015. PMLR. 2

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2

[52] Yiwei Hu, Paul Guerrero, Milos Hasan, Holly Rushmeier, and Valentin Deschaintre. Generating procedural materials from text or image prompts. In *ACM SIGGRAPH 2023 Conference Proceedings*, New York, NY, USA, 2023. Association for Computing Machinery. 2

[53] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024. 2

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3

[56] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022. 3

[57] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney

Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. 3, 4

[58] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. 3

[59] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3

[60] Chuanguang Yang, Zhulin An, Libo Huang, Junyu Bi, Xinqiang Yu, Han Yang, Boyu Diao, and Yongjun Xu. CLIP-KD: An empirical study of CLIP model distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15952–15962, 2024. 3

[61] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020. 3

[62] 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, 2023. 4

[63] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 4

[64] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin

Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024. 4

[65] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3B VLM for transfer, 2024. 4

[66] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. 4

[67] Stephanie Fu, Netanel Yakir Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 4, 18

[68] Shobhita Sundaram, Stephanie Fu, Lukas Muttenthaler, Netanel Yakir Tamir, Lucy Chai, Simon Kornblith, Trevor Darrell, and Phillip Isola. When does perceptual alignment benefit vision representations? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4, 18

[69] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 4

[70] 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*, pages 7514–7528, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 4

[71] 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*, New York, NY, USA, 2023. Association for Computing Machinery. 4

[72] 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*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. 4

[73] 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)*, pages 514–520, Florence, Italy, 2019. Association for Computational Linguistics. 4

[74] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 5

[75] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-Coder technical report, 2024. 6, 15

[76] Michael Saxon, Fatima Jahara, Mahsa Khoshnoodi, Yujie Lu, Aditya Sharma, and William Yang Wang. Who evaluates the evaluations? objectively scoring text-to-image prompt coherence metrics with t2IScorescore (TS2). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6

[77] Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. *Best–Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015. 6

[78] 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*, pages 811–817, San Diego, California, 2016. Association for Computational Linguistics. 6

[79] 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)*, pages 465–470, Vancouver, Canada, 2017. Association for Computational Linguistics. 6

[80] Bryan K. Orme. MaxDiff analysis: Simple counting, individual-level logit, and HB. *Sawtooth Software Research Paper Series*, 2009. 6

[81] Bruce Schneier. *Applied cryptography - protocols, algorithms, and source code in C, 2nd Edition*. Wiley, 1996. 7

[82] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online, 2020. Association for Computational Linguistics. 8

[83] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 8

[84] Anas Awadalla, Le Xue, Manli Shu, An Yan, Jun Wang, Senthil Purushwalkam, Sheng Shen, Hannah Lee, Oscar Lo, Jae Sung Park, Etash Kumar Guha, Silvio Savarese, Ludwig Schmidt, Yejin Choi, Caiming Xiong, and Ran Xu. BLIP3-KALE: Knowledge augmented large-scale dense captions. In *Synthetic Data for Computer Vision Workshop @ CVPR 2025*, 2025.

[85] Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, Yuyin Zhou, and Cihang Xie. What if we recaption billions of web images with LLaMA-3?, 2024. 8

[86] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742, 2006. 8

[87] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023. 8

[88] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. LanguageBind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*, 2024. 8

[89] OpenAI. Learning to reason with LLMs, 2024. 15

[90] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. 15

[91] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 15

[92] Urbano Lorenzo-Seva and Jos M. F. ten Berge. Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 2(2):57–64, 2006. 15

[93] 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*, pages 358–374, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. 15

[94] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China, 2019. Association for Computational Linguistics.

[95] Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1656–1671, Online, 2020. Association for Computational Linguistics.

[96] Yurun Song, Junchen Zhao, and Lucia Specia. SentSim: Crosslingual semantic evaluation of machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3143–3156, Online, 2021. Association for Computational Linguistics. 15

[97] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66, 1998. 15

[98] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966, Lille, France, 2015. PMLR. 15