Reason out Your Layout: Evoking the Layout Master from Large Language Models for Text-to-Image Synthesis

Anonymous ACL submission

Abstract

Recent advancements in text-to-image (T2I) generative models have shown remarkable capabilities in producing diverse and imaginative visuals based on text prompts. Despite the ad-004 vancement, these diffusion models sometimes struggle to translate the semantic content from the text into images entirely. While conditioning on the layout has shown to be effective in improving the compositional ability of T2I diffusion models, they typically require manual layout input. In this work, we introduce a novel approach to improving T2I diffusion models us-012 ing Large Language Models (LLMs) as layout 014 generators. Our method leverages the Chainof-Thought (CoT) prompting of LLMs to interpret text and generate spatially reasonable ob-016 ject layouts. The generated layout is then used 017 to enhance the generated images' composition and spatial accuracy. Moreover, we propose an efficient adapter based on a cross-attention mechanism, which explicitly integrates the layout information into the stable diffusion models. Our experiments demonstrate significant improvements in image quality and layout accuracy, showcasing the potential of LLMs in augmenting generative image models.

1 Introduction

027

034

039

042

Recent developments in image generation, particularly with DELL-E 2 (Ramesh et al., 2022) and Stable Diffusion (SD) (Rombach et al., 2022). Specifically, text-to-image (T2I) models, which create images from textual descriptions using autoregressive and diffusion methods, have shown a notable ability to produce high-quality images (Saharia et al., 2022; Ramesh et al., 2021, 2022). SD-based models, in particular, have obtained significant attention in the research community due to their public availability. However, creating realistic images from complex descriptions still remains challenging. For instance, when dealing with descriptions that include multiple objects with complex spatial relationships, SD-based models often struggle to compose these elements within an image accurately. Fig. 1(a) shows some examples when multiple objects are described in the text prompt, SD-based models fail to capture them all in the images.

The compositional challenges in SD-based models, including attribute leakage, incorrect attribute binding, omission of objects, or misinterpretation of relationships between objects, are well documented (Wu et al., 2023; Feng et al., 2022). To improve compositional capabilities, a common strategy is to manually provide object positions as model inputs, circumventing the need for the model to infer the layout independently (Zhang et al., 2023; Li et al., 2023). Li et al. (2023), specifically, suggest using bounding boxes to guide image generation, encoding object positions and descriptions into vectors that influence the latent image development via an attention module. Additionally, other research, such as that by Chefer et al. (2023) and Liu et al. (2022), proposes modifications to the generation process by adjusting the attention mechanism. Additionally, Chefer et al. (2023); Liu et al. (2022) propose attention mechanism adjustments, e.g., modifying attention scores to ensure visual features adequately represent each object, thereby reducing object omissions in images. While these layout-augmented methods have been effective, they suffer from the reliance on human-annotated object locations. Furthermore, the integration methods for layout information in these models can be seen as unnatural, as they fail to utilize the spatial details explicitly (Li et al., 2023) or heavily alter the image formation process (Liu et al., 2022; Chefer et al., 2023).

To minimize human intervention in the training process of text-to-image (T2I) generative models, we leverage the capabilities of Large Language Models (LLMs) (Touvron et al., 2023a; Brown et al., 2020) for generating coherent layouts using Chain-of-Thought (CoT) prompting (Wei et al., 2022) (Fig. 1(b)). In particular, we activate LLMs' 043



Figure 1: (a) Our method enhances the compositional capability of a pre-trained text-to-image diffusion model (Rombach et al., 2022) by conditioning on object layouts; (b) Unlike GLIGEN or ControlNet, which requires manually annotating the layout modality, our method uses LLMs to generate one from the given text prompt.

potential in generating coherent layouts using CoT prompting (Wei et al., 2022). The layouts generated by LLMs provide bounding boxes for each object mentioned in the text prompts. For incorporating the LLM-generated layout into the input of SD-based models, we propose an effective adapter that integrates layout information through a crossattention mask. This adapter explicitly utilizes the spatial details provided by the layout and is designed to align seamlessly with the conditioning mechanisms of SD.

We empirically our method's efficacy in terms of generation quality, layout accuracy, and composition accuracy. Layout generation accuracy accesses the adapter's precision in placing objects within specified bounding boxes. Composition accuracy, a universal metric for T2I models, measures the successful depiction of text-mentioned objects in the generated images. We also explore how different prompting strategies influence layout creation and, consequently, image quality. We summarize our contributions as follows:

- we propose a new pipeline for layout-aware text-to-image diffusion models;
- we use LLMs as layout generators and improve their performance via CoT prompting, which elicits reasoning steps in LLMs for more accurate layout generation;
- we propose LACA, an adapter designed to incorporate spatial information from given layouts *explicitly* into Stable Diffusion models;
- our empirical study demonstrates that our proposed LLM-based layout generator can generate layouts that resemble the real ones, thereby enhancing the composition accuracy of the

generated images.

119

120

121

122

123

124

125

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

152

2 **Related Work**

2.1 **Text-to-Image Generation**

Text-conditional image generation is a key focus in multi-modal learning, with substantial progress in creating realistic images (Goodfellow et al., 2014; Reed et al., 2016; Ding et al., 2021; Yu et al., 2022; Rombach et al., 2022; Nichol et al., 2021). Diffusion models, particularly those introduced by (Sohl-Dickstein et al., 2015; Ho et al., 2020), have gained prominence in this field, thanks to their iterative refinement process and training stability. For instance, Rombach et al. (2022) introduced a latent diffusion model that achieves high performance on minimal computing power, while (Nichol et al., 2021) developed a method for effectively guiding text to create and edit photorealistic images. Despite these advancements, generating photorealistic images from complex text prompts remains a challenge, as highlighted by (Huang et al., 2023; Cong et al., 2023; Li et al., 2022b). Realworld descriptions often involve detailed scenes with complex object interactions, a task that has been approached previously through scene graph parsing (Johnson et al., 2018; Zellers et al., 2018). However, few studies have managed to generate images that closely match intricate text prompts. Some researchers have suggested conditioning spatial features, like segmentation or bounding boxes, to enhance spatial relation modeling (Li et al., 2023; Zhang et al., 2023). These methods, however, rely on manually created spatial features. For example, Li et al. (2023) require manual layout annotation with bounding boxes, and Zhang et al. (2023) ex-

103

107

111

112

114

117

tract the layout (edge map, segmentation, etc) from
a template image, and ask the diffusion models
to generate a new one conditioning on it. These
approaches require special and often manual treatments for layout creation.

159

160

161

162

163

164

165

166

170

171

174

175

176

178

179

180 181

182

183

187

190

192

193

194

195

196

198

199

202

2.2 Chain-of-Thought Prompting for LLMs

In this study, we explore the use of a Large Language Model (LLM) to automatically generate layouts directly from textual descriptions. LLMs, characterized by their transformer-based architecture and enormous size—often comprising hundreds of billions of parameters—include notable examples like GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), and LLaMA (Touvron et al., 2023a,b). Trained on extensive textual datasets (Shanahan, 2022), these models exhibit exceptional ability in understanding natural language and performing complex text generation tasks.

The specific technique we use to extract the layout from an LLM is CoT prompting (Wei et al., 2022). This is a method that directs LLMs to deconstruct problems into logical steps, facilitating complex reasoning without iterative modifications to the model's parameters. This approach is particularly effective in scenarios with limited examplebased learning. It functions by using straightforward instructional sentences to prompt LLMs to process information sequentially (Kojima et al., 2022; Wang et al., 2023), or by presenting them with a series of examples that illustrate the reasoning process step-by-step (Wei et al., 2022; Zhang et al., 2022). Typically, CoT is implemented in single interactions, where the model generates a continuous chain of reasoning before arriving at an answer. In this work, we leverage the common knowledge the LLMs accumulated during their training processes to generate plausible object layouts of an image. We further use CoT prompting to enhance the accuracy of the LLMs' responses.

3 Preliminaries

3.1 Latent Diffusion Models

Our method builds upon the Stable Diffusion model (Rombach et al., 2022). The Stable Diffusion model consists of an autoencoder and a latent denoising model. In this setup, an encoder, \mathcal{E} , first transforms an image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ into a latent code $\mathbf{z} = \mathcal{E}(\mathbf{x})$. This latent code is then decoded back to the original image by the decoder, \mathcal{D} , as $\tilde{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$. The latent code $\mathbf{z} \in \mathbb{R}^{h \times w \times c}$ can be viewed as a downsampled visual feature map,



Figure 2: Generation pipeline of our proposed method: Given a caption, we employ LLM to generate an object layout. This layout is injected into the Stable Diffusion model's noise prediction via our proposed LACA.

where h < H and w < W. The training process involves first training the autoencoder, followed by the latent denoising model.

203

204

205

206

207

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

227

228

229

230

231

232

Once the autoencoder is trained, a denoising model ϵ_{θ} is trained to generate the latent code $\mathbf{z} = \mathcal{E}(\mathbf{x})$ starting from a Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. The generation process is the reverse of the diffusion process, which progressively adds noise over a series of timesteps T (Ho et al., 2020). Specifically, given an image \mathbf{x} and the corresponding text y, the denoising model ϵ_{θ} is optimized using the following training loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z} \sim \mathcal{E}(\mathbf{x}), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}), t} \left[\|\epsilon - \epsilon_{\theta}(\mathbf{z}_{t}, t, c(y)\|_{2}^{2} \right].$$
(1)

Here \mathbf{z}_t is a noisy version of \mathbf{z} at timestep t, and the text information is extracted by a CLIP text encoder $c(\cdot)$ (Radford et al., 2021). During training, the conditioning text y is occasionally replaced with empty input \emptyset with some probability p to enable classifier-free guidance (CFG) (Ho and Salimans, 2022). The denoising model ϵ_{θ} incorporates the UNet architecture (Ronneberger et al., 2015) with self-attention and cross-attention layers. The cross-attention layers are primarily used to integrate text information.

3.2 Guided Diffusion via Cross-Attention

In Stable Diffusion, text information is introduced through cross-attention between the intermediate visual features from the score network ϵ_{θ} and the text embeddings c(y). Let L be the number of downsampling blocks or upsampling blocks in ϵ_{θ} , and $p_l : l \in \{1, ..., L\}$ be the resolution of the visual feature map output by the *l*-th downsampling blocks.¹ Note that $p_0 = h = w$ represents the resolution of the latent code \mathbf{z}_t . Given the text input y, its corresponding CLIP text embeddings c(y) are in dimension $\mathbb{R}^{N \times d}$. Here N is the number of tokens and d is the dimension of each token vector.

For each intermediate visual feature map \mathbf{z}_t^l and the text input c(y), An attention map $\mathbf{A}_t^l \in \mathbb{R}^{p_l \times p_l \times N}$ is computed using the query (Q) linearly projected from \mathbf{z}_t^l and the key (K) linearly projected from the text embeddings c(y). Intuitively, in the attention map, each N-dimensional slice $\mathbf{A}_t^l[i, j, :]$ is a probability vector that represents the portion of information each semantic (token) should be aggregated into the visual feature vector at location (i, j) of \mathbf{z}_t^l . The vectors that carry the token semantics are another linear projection (V) of the text embeddings.

4 Methodology

234

235

240

241

242

246

247

248

251

254

259

261

262

264

265

270

271

272

273

274

276

277

278

279

4.1 LLM as Layout Generator

In this section, we explore the potential of LLMs in generating bounding boxes for objects based on text prompts. Two key factors enable LLMs to generate object layouts: defining **precise task instructions** and providing **sufficient in-context examples**. We elaborate on how to turn a general LLM into a layout generator in § 4.1.1. In § 4.1.2, we show how to improve its spatial sense by providing in-context examples with CoT prompting. Fig. 2 demonstrates our proposed generation pipeline.

4.1.1 Task Instructions

We define clear and comprehensive task instructions for the LLMs to create accurate visual layouts. These instructions focus on distinguishing between visible and abstract elements, resolving ambiguities, understanding spatial relationships, and providing specific object coordinates within a defined canvas area.

Correctly identifying visual objects. When parsing objects from the caption, not all objects can be depicted in visual format because they pertain to the other senses or are abstract concepts. For example, in the phrase "in an office", the entity "office" represents a scenario composed of multiple objects. Thus, it should be considered as the background of an image rather than an element in object arrangement.

¹For notation simplicity, we focus on downsampling blocks here.

Resolving ambiguity. Ambiguities in language, such as vague descriptions or pronouns lacking clear referents, need clarification. For example, pronouns like "it" or "they" require specific antecedents to avoid confusion in visual representation. Additionally, exact object quantities should be clarified when descriptions like "a group of" or "several" are used.

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

330

331

Interpreting spatial relations. Grasping how objects relate to each other in space is crucial. Specifically, recognizing spatial cues such as "behind", "in front of", and "next to" can help generate accurate object placement. Moreover, it is also essential to correctly infer the spatial relationship in the absence of cues. For example, the sentence "a man holding a tennis racket" implies that the object "tennis racket" should be placed near the "man's" hand. Generating valid answers. We ask the LLMs to arrange the objects within a 512×512 canvas. The top-left coordinate of the canvas is the origin (0, 0). For each object, LLMs are asked to provide the bounding box's top-left and bottom-right coordinates. LLMs use the exact phrase from the caption to represent the identified object.

4.1.2 In-Context Examples with CoT Prompting

In addition to the specific task instructions, we also offer in-context examples to enhance LLMs' ability to produce higher-quality responses. While constructing input-output pairs is straightforward, in this section, we illustrate how to develop CoT prompting to assist the LLMs in generating better outcomes.

For intricate captions involving several objects, it's advantageous for the LLMs to break down the task into multiple steps and address each sequentially before generating the object layouts. Introducing examples for reasoning in these in-context examples enables the LLMs to emulate such reasoning patterns and engage in CoT reasoning for the caption query. More precisely, we design reasoning frameworks that adhere to the sequence outlined in the task instructions in Section 4.1.1: (1) identifying both visual and non-visual elements in the caption; (2) resolving any ambiguities regarding the quantity of objects or their attributes, often by making gentle assumptions for clarity; (3) deducing the spatial relations among visual objects in the scene, based on spatial indicators or common knowledge; (4) concluding the arrangement of objects and ultimately deriving the answer from

In-Context Examples LLM Input LLM Input Caption: A glass bowl full of oranges and apples Caption: A glass bowl full of oranges and apples. Reasoning: We can identify the following objects: A glass bowl, oranges, apples - A glass bowl:[(153, 153, 358, 358)] oranges and apples, number that could fit within a bowl, such as three of each. The "glass bowl" is the container for the other objects, meaning the oranges and apples are within it. The "oranges and apples" are grouped together as they are contained in the bowl. As a result, the glass bowl will take a central position on the canvas, - Oranges: [(179, 179, 230, 230), (281, 179, 332, 230), (230, 281, 281, 332)] Apples: [[(79, 281, 230, 332), (281, 281, 332, 332), (256, 204, 307, 256)1 with a large bounding box, and there will be multiple bounding boxes for orange and apples within the bowl. Caption: A black and white photo of a desk with a keyboard and monitor. Objects: LLM Output A glass bowl:[(153, 153, 358, 358)] - Oranges: [(179, 179, 230, 230), (281, 179, 332, 230), (230, 281, 281, 332)] - Apples: [(179, 281, 230, 332), (281, 281, 332, 332), (256, 204, 307, 256)] Objects: - A desk: [(37, 238, 475, 513)] - A keyboard: [(148, 275, 420, 325)] - A monitor: [(171, 214, 397, 274)] Caption: A black and white photo of a desk with a keyboard and monitor. LLM Output Reasoning: We can identify the following objects: a desk. a keyboard. a monitor. and white photo The "black and white photo" aspect suggests that the entire scene should be imagined in grayscale, but this does not affect the spatial arrangement. The "desk" is likely the base for the other objects, which means the keyboard and the monitor will be placed on top of the desk. The monitor would be placed at the back of the desk, with the keyboard in front of it based on a standard desk Identifying visual objects As a result, the desk would stretch across the bottom of the canvas, covering the entire width but not extending too far upwards. The box of the monitor would be vertical and centered horizontally on the desk. The Keyboard Clarifying ambiguity wider than it is deep, so the box would be placed in front of the monitor, wide but not extending too far back or forward. Parsing object Hierarchy and relationship Objects: - A desk: [(50, 190, 462, 462)] - A keyboard: [(143, 350, 368, 420)] Inferring bounding box arrangement - A monitor: [(165, 135, 347, 315)]

Figure 3: In-context examples with CoT reasoning enable LLM to give a more nuanced layout solution. Reasoning processes of different perspectives are highlighted with different colors.

the preceding analysis. Figure 3 illustrates how an LLM employs CoT prompting to successfully determine a logical layout, while using standard 334 in-context examples fails to do so.

338

339

341

342

343

346

354

4.2 Stable Diffusion with Layout Conditions

We propose an attention-based adapter that incorporates the object layout into the score network ϵ_{θ} by manipulating the attention mask $\mathbf{M}^{l} \in$ $\{0,1\}^{p_l \times p_l \times N}$. Following Li et al. (2023), we freeze the original model weight of ϵ_{θ} , and introduce a learnable cross-attention module in each transformer block. The original transformer block in ϵ_{θ} consists of a self-attention layer and a crossattention layer. After adding the layout-aware cross-attention module (LACA), the computation of \mathbf{z}_t can be written as

$$\mathbf{z}_t = \mathbf{z}_t + \mathrm{SA}(\mathbf{z}_t; \theta); \tag{2}$$

$$\mathbf{z}_t = \mathbf{z}_t + \text{LACA}(\mathbf{z}_t, c(y), \mathbf{M}; \phi); \qquad (3)$$

$$\mathbf{z}_t = \mathbf{z}_t + CA(\mathbf{z}_t, c(y); \theta).$$
(4)

We omit the superscript l for notation simplicity. We denote ϕ to be the parameters of the adapter. In the following, we elaborate on the design details of the proposed adapter.

4.2.1 Layout-Aware Cross-Attention

Recall that the visual feature vector $\mathbf{z}_t[i, j, :]$ is updated with the weighted sum of the linear projection (V) of c(y), where the weight is determined by the cross-attention map $\mathbf{A}_t[i, j, :]$. When the layout indicates the presence of an object in a specific location (i, j) of the feature map \mathbf{z}_t , it is intuitive to aggregate only the semantics of the specific object into this area.

Composing attention mask from layout. Since we specify the LLM to use the same phrase in the caption to represent the object, it is easy to locate the token indices corresponding to an object. Given the token position information and the layout spatial information of each object, we can construct the attention mask M with the following rules:

- if the *n*-th token describes an object who is assigned to location (i, j), $\mathbf{M}[i, j, n] = 1$;
- if the *n*-th token does not describe any object, $\mathbf{M}[i, j, n] = 1;$

• otherwise,
$$\mathbf{M}[i, j, n] = 0$$

When the *n*-th token is assigned to a specific object, we use rule 1 and rule 3 to decide the attention mask value based on the object's location. When the n-th token is not assigned to any object, we

In-Context Examples with CoT

356

359

360

361

362

363

364

365

366

367

368

369

371

372

373

374

375

376

Construct attention mask



Figure 4: LACA injects layout information via crossattention mask. The cross attention mask indicates what semantics (e.g., "a blue bird") should attend to the (i,j)-th location (i,j) in the visual feature map. After computing the attention score, the cross-attention mask will enforce the (i,j)-th visual vector only aggregate the semantics from the designated token embeddings.

set the attention mask M[:, :, n] to be 1 such that the token attends to all visual features. Such construction ensures the visual features z_t aggregate desired semantics from the text embeddings. Fig. 4 provides an example of constructing such an attention mask and how LACA computes a single visual feature $z_t[i, j, :]$.

Module designs. We directly initialize LACA with the model weight from the cross-attention module that follows it in the Stable Diffusion. Moreover, we add zero convolution layer (Zhang et al., 2023) at the output stage of LACA such that the training is more stable.

4.2.2 Sampling

381

384

390

394

400

401

402

During generation, we only use LACA for the first 20% denoising steps, then the standard denoising scheme for the rest steps. This is because once the latent code z_t encodes a certain amount of the semantics of the objects, the original cross-attention mechanism will function properly. In other words, to improve compositional capability, all an LDM needs is a boost in the early stage of the generation. When using LACA, we employ the classifier-free

guidance similar to Brooks et al. (2023):

$$\tilde{\epsilon} = \epsilon_{\theta}(\mathbf{z}_t, t, \emptyset) + g_1(\epsilon_{\theta}(\mathbf{z}_t, t, c(y)) - \epsilon_{\theta}(\mathbf{z}_t, t, \emptyset))$$
404

403

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

$$+ g_2(\epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y), \mathbf{M}) - \epsilon_{\theta}(\mathbf{z}_t, t, c(y)))).$$
(5)

Empirically we find such a setting works best for our method. We also investigate other possible CFG options in our experiment (see Appendix C).

5 Experiments

In this section, we evaluate the quality of the LLMgenerated layouts and validate the effectiveness of the proposed adapter. Our experimental details can be found in Appendix A.

5.1 Quantitative Evaluation

We first evaluate our method's generative quality and its ability to accurately compose elements using the Flickr30K (Plummer et al., 2015) and the COCO2017 (Lin et al., 2014). Both datasets are standard benchmarks, in which each image sample comes with caption and object annotations like bounding boxes and semantic masks. We focus on bounding boxes in our work. A main distinction between the two datasets is that Flickr30K derives the bounding boxes' noun entities directly from the captions, whereas COCO2017's boxes and captions might refer to different objects in the images.

Baselines. We use GLIGEN (Li et al., 2023), Stable Diffusion (Rombach et al., 2022) and Attendand-Excite (Chefer et al., 2023) as our baseline methods. We train GLIGEN following the setting in Li et al. (2023) from the training data. For Stable Diffusion and Attend-and-Excite, we use the v1-5 model weight from Huggingface (Wolf et al., 2019).

Layout conditions. We consider two layout conditions - one provided by the datasets, referred to as ground-truth (GT), and one generated by LLMs. The GT layouts in COCO2017 do not match the ground-truth captions and are limited to 80 object classes, whereas the GT layout for Flickr30K and all LLM-generated layouts are matched with their captions. Our proposed method, LACA, only addresses the latter scenario. For COCO2017, with its GT layouts, we modify the captions to include all noun entities present in the layout, making them compatible with LACA.

Evaluations. We evaluate the models in terms of image quality and layout accuracy. Specifically, we use FID (Heusel et al., 2017) for image quality,

			Flickr30K				COCO2017				
Layout source	Method	FID↓	YOL AP	O / <mark>GLII</mark> AP ₅₀	score↑ AP ₇₅	GLIP rate↑	FID↓	YOLC AP) / <mark>GLIP</mark> AP ₅₀	score↑ AP ₇₅	GLIP rate↑
-	Stable Diffusion	22.53	-	-	-	78.3	20.91	-	-	-	71.2
-	Attend-and-Excite	34.05	-	-	-	84.1	32.12	-	-	-	77.6
GT	GLIGEN	27.70	45.0	50.9	45.2	84.5	27.09	19.1	30.5	20.8	71.0
01	LACA	22.65	50.4	56.6	50.9	84.3	25.33	17.4	28.9	20.1	70.1
LLM	GLIGEN	30.08	56.2	63.8	56.5	83.3	25.97	57.6	68.1	46.2	70.5
	LACA	28.96	58.5	68.1	59.4	83.3	23.28	58.8	67.3	50.1	78.1

Table 1: Generative performance on Flickr30K and COCO2017 datasets. FID evaluates image quality, YOLO/GLIP score evaluates correspondence to the conditioned layout, and GLIP rate evaluates composition accuracy. The conditioned layout comes from either the dataset (GT) or the LLM generation.

450 and for layout accuracy, we consider using YOLO score (Li et al., 2021) for independent layout con-451 dition (COCO2017 with GT) and GLIP score (Li 452 et al., 2023) for derived layout condition (Li et al., 453 2023). We report average precision (AP) for both 454 YOLO and GLIP scores. While layout accuracy 455 only applies to layout-conditioned models, we ad-456 ditionally evaluate the composition accuracy of any 457 T2I models. Specifically, the composition accuracy 458 measures how many objects mentioned in the text 459 are present in the generated image. This can be 460 quantified by again using GLIP (Li et al., 2022a). 461 Since GLIP derives all entities from a caption and 462 tries to detect the entities in its corresponding im-463 ages, we can quantify composition accuracy by 464 465 computing the ratio:

466

467

468

469

470

471

$$\frac{\sum_{i} \# \text{ entities detected in image } i}{\sum_{i} \# \text{ entities derived from caption } i}, \quad (6)$$

dupped GLIP rate. The numerator is always smaller than the denominator as we only consider known entities. All metrics composition-related metrics (YOLO score, GLIP score, and GLIP rate) are scaled by 100.

Results. Table 1 demonstrates the performance 472 of different models when combined with different 473 layout sources. We interpret the numbers with the 474 following perspectives: (1) There exists a trade-475 off between image quality and compositional 476 accuracy in the investigated models. (2) By com-477 paring the performance of different models under 478 the same layout source, we can see that LACA 479 consistently achieves better FID and GLIP scores. 480 The difference is more prominent in the Flickr30K 481 dataset. While in the COCO2017 dataset paired 482 with GT layouts, LACA works slightly falls short 483 484 of GLIGEN regarding the YOLO score. We hypothesize this is because the adapted captions for 485 LACA are not as coherent as the original captions. 486 When compared to Stable Diffusion, all methods 487 except Stable Diffusion demonstrate superior GLIP 488

rates, indicating they are more effective at incorporating objects mentioned in the captions into the generated images. (3) By comparing the performance of the same models under different layout sources, we can observe that in Flickr30K, the FID scores from LLM-generated layouts are worse than from the GT layouts, while in COCO2017, the FID scores from LLM-generated layouts are better. This discrepancy likely stems from the text-layout consistency of the datasets – a text-consistent layout eases the burden of the diffusion models to integrate objects from multiple modalities.

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

5.2 Ablation Study of Prompting Strategies

We investigate the effect of different prompting approaches on layout generation using the Flickr30K dataset. Our focus is to compare the resemblance of LLM-generated bounding boxes to the original GT bounding boxes.

Evaluation protocol. We randomly sample 5000 caption-layout pairs from the dataset. For each caption, an LLM is used to generate layouts using different prompting strategies. Given the GT layout and the LLM-generated layouts, we measure two metrics: (1) the object hit rate, which measures how many objects in the GT layouts have been identified by the LLMs; (2) the intersection over union (IoU) of the GT and the LLM-generated bounding boxes. While the generated boxes can be completely different from the ground-truths for an individual comparison, the mIoU is still a reliable measure of the goodness of a layout generator. Additionally, we assess the image quality produced from LLM-generated layouts using the FID score.

Prompting variants. We study three types of prompting strategies: (1) task instructions only; (2) task instructions followed by eight in-context examples, but without CoT reasoning; (3) task instructions followed by eight in-context examples, each come with a CoT reasoning.



Figure 5: Demonstration of how in-context examples with CoT improve the quality of the generative layout. When the caption describes a counterfactual scene (left) or does not explicitly provide object relation(right), CoT enables LLMs to make assumptions and inferences before generating the layout.

Prompting strategy	hit rate	mIoU	FID
Instruction only	92.2%	16.54	32.43
+ ICL	97.1%	19.84	31.95
+ ICL w/ CoT	97.4%	23.98	28.96

Table 2: In-context examples with CoT help improve the layout performance, thus the generated image quality. ICL stands for in-context learning.

Results. Table 2 reports and mIoU and the object hit rate of the generated layouts. As we can observe, when providing in-context examples, the object hit rate reaches almost 100% and no longer increases. Moreover, when CoT reasoning is employed, we achieve the highest mIoU, indicating the generated layouts are most similar to the real ones.

5.3 Generative Counting Accuracy

528

530

531

534

535

537

539

540

546

547

550

We evaluate the performance of different models in generative counting accuracy, which is their ability to replicate the exact number of objects as described in the text. For this purpose, we curated a set of captions from the COCO2017 validation set that specifically include the numeral keywords "two", "three" or "four", in which 150 captions were collected for each number category. Using LLMs, we generate object layouts for the collected captions. From Fig. 6, we can see that LLMgenerated layout generally improves the model's accuracy in depicting the precise count of objects. The improvement becomes more notable when the object count rises. Moreover, we can observe that LACA surpasses GLIGEN in its effectiveness in



Figure 6: LACA achieves better generative counting accuracy compared to GLIGEN and Stable Diffusion.

551

552

553

554

555

556

557

558

559

561

562

563

564

565

567

569

employing the layout modality.

6 Conclusion and Discussion

In this work, we propose to use LLMs to generate object layouts from the given captions. By introducing the CoT prompting and carefully designing the reasoning steps for the in-context examples, we enable the LLMs' to generate more nuanced layout solutions and showcase the potential of LLMs in understanding and generating complex visual layouts. We further propose LACA, an adapter that explicitly incorporates the object layout information into the Stable Diffusion. We demonstrate that LACA is superior in yielding high composition accuracy, without conditioning on human-annotated layout modality. This work provides clear direction for ongoing research to refine the synergy between textual descriptions and visual generation, paving the way for more sophisticated and accurate visual content creation by LLMs.

References

570

571

573

574

581

582 583

584

586

589

590

591

592

594

595

597

598

610

611

613

617

618

619

621

625

- Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics* (*TOG*), 42(4):1–10.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Yuren Cong, Martin Renqiang Min, Li Erran Li, Bodo Rosenhahn, and Michael Ying Yang. 2023. Attribute-centric compositional text-to-image generation. arXiv preprint arXiv:2301.01413.
- Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. 2021. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835.
- Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2022. Training-free structured diffusion guidance for compositional text-to-image synthesis. arXiv preprint arXiv:2212.05032.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840– 6851.
- Jonathan Ho and Tim Salimans. 2022. Classifierfree diffusion guidance. arXiv preprint arXiv:2207.12598.

- Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. 2023. T2i-compbench: A comprehensive benchmark for open-world compositional text-toimage generation. *arXiv preprint arXiv:2307.06350*.
- Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199– 22213.
- Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022a. Grounded language-image pre-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10965– 10975.
- Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521.
- Zejian Li, Jingyu Wu, Immanuel Koh, Yongchuan Tang, and Lingyun Sun. 2021. Image synthesis from layout with locality-aware mask adaption. In *Proceedings* of the IEEE/CVF International Conference on Computer Vision, pages 13819–13828.
- Zhiheng Li, Martin Renqiang Min, Kai Li, and Chenliang Xu. 2022b. Stylet2i: Toward compositional and high-fidelity text-to-image synthesis. In *Proceedings* of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18197–18207.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision– ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. 2022. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer.
- Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.
- OpenAI. 2023. ChatGPT. https://openai.com/chatgpt.

669

670

671

672

673

674

675

676

677

678

679

680

788

790

791

792

- 681 682
- 683 684
- 698 699 700 701 702 703 704 705 706 707 708 709
- 709 710 711 712 713 713 714 715 716
- 718 719 720 721
- 723 724 725
- 726 727 728
- . 7 7 7
- 731 732 733

.

- 734
- 735 736

- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer imageto-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
 - Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical textconditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.
- Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016.
 Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. Highresolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing* and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494.
- Murray Shanahan. 2022. Talking about large language models. *arXiv preprint arXiv:2212.03551*.
- Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. 2019. Objects365: A large-scale, high-quality

dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.

- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Planand-solve prompting: Improving zero-shot chain-ofthought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-ofthe-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Qiucheng Wu, Yujian Liu, Handong Zhao, Trung Bui, Zhe Lin, Yang Zhang, and Shiyu Chang. 2023. Harnessing the spatial-temporal attention of diffusion models for high-fidelity text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7766–7776.
- Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*.
- Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing

- with global context. In *Proceedings of the IEEE con- ference on computer vision and pattern recognition*,
 pages 5831–5840.
- 796 Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023.
 797 Adding conditional control to text-to-image diffusion 798 models. In *Proceedings of the IEEE/CVF Interna-*799 *tional Conference on Computer Vision*, pages 3836– 800 3847.
- 801 Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex
 802 Smola. 2022. Automatic chain of thought prompt803 in large language models. *arXiv preprint*804 *arXiv:2210.03493*.

05 Appendix

810

811

812

814

815

817

A Implementation Details

A.1 Experimental Setup

We follow Li et al. (2023) to train the LACA adapter on a combination of four grounding datasets: Object365 (Shao et al., 2019), GoldG (Li et al., 2022a), CC3M (Sharma et al., 2018) and SBU (Ordonez et al., 2011). We train LACA with batch size 128 and a learning rate 5e-5 for 700k iterations using 8 A100 GPUs, which requires approximately two days to finish. To enable classifier-free guidance, we randomly drop captions with 5% probability and both caption and layout with another 5% probability. We set $g_1 = g_2 = 5.5$ during generation for all experiments. We generate all layouts using GPT-3.5turbo (OpenAI, 2023).

A.2 Prompt Designs

We first elaborate on the realization of task instructions, and then we explore different variants of chain-ofthought prompting provided in the In-Context examples.

Task instructions. The task instructions implemented are shown in Table 3. While most of the instructions are straightforward, we specifically focus on the different ways to depict object placement in the responses from the LLMs. We explore variations from two perspectives: (1) using a normalized 1x1 canvas as opposed to a more expansive 512x512 canvas, and (2) two methods of representing object locations - either through top-left and bottom-right coordinates (XYXY) or by indicating the top-left coordinate with width and height descriptions for the bounding box (XYWH). Our ablation study, which assesses the quality of images produced under these different settings via the FID score, revealed a slightly better performance with the combination of a 512x512 canvas and the XYXY method for bounding box representation.

In-Context Examples. We explore the three formats of CoT reasoning. The first CoT variant includes a two-step reasoning process: first interpreting the visual objects from text, then creating their arrangements. This basic approach generally suffices for accurate layout creation. However, for cases with complex or vague object relationships, it's advantageous for LLMs to engage in inference and assumption before solution generation. In the second variant, we integrate clarification steps within the CoT reasoning of the first version. The clarification steps are mostly useful for prompts that do not carry enough information on the object position or specification information, such as e.g., quantity, attribute, etc. For the first two variants, each reasoning step in the first two formats is separated by bullet points in order to make it easier for the LLMs to mimic. The final CoT format adheres to the reasoning approach of the second variant but combines all reasoning steps continuously without explicitly delineating each reasoning phase. We provide an example demonstrating those variants in Table 4

A.3 Layout Conversion from LLMs Response

We extract bounding boxes for objects identified in the responses from LLMs. These bounding boxes are then transformed into the cross-attention mask \mathbf{M}^l used by LACA. Without loss of generality, we denote $\mathbf{M} := \mathbf{M}^0 \in \mathbb{R}^{64 \times 64 \times N}$ and use \mathbf{M} for the discussions below.

Bounding boxes extraction. As stated in our task instructions, the canvas that an LLM operates on is a size 512x512. An LLM will first parse objects from the caption and then classify them into either "visual" or "non-visual". Only "visual" objects are assigned bounding boxes, which are determined by their top-left and bottom-right coordinates. If an object is mentioned in multiple quantities, the LLM will generate a corresponding number of bounding boxes, based on either the specified quantity or its own estimations. After extraction, these coordinates are normalized to a scale ranging from 0 to 1.

Cross-attention mask construction. For each object, we first locate its indices in the text caption in order to properly assign values to M. For example, in the text prompt "a red apple and a blue bird", the indices of the object "a red apple" will be [0, 1, 2] (assuming the index starts from 0). In practice,

	 Given the description, parse the objects that appear in the text in a hierarchical manner. Based on your parsed result, arrange the objects within a canvas with a width of 512 and a height of 512.
	 The top-left coordinate in the canvas is the origin (0, 0). 3. For each object, you need to specify its location by listing the top-left coordinate and the bottom-left coordinate. Your answer for each object should be (x1, y1, x2, y2), where (x1, y1) is the top-left coordinate and (x2, y2) is the bottom-right coordinate. 4. In the description, if there is any ambiguity about the number of objects or the spatial relationship between objects, you should first concretize it through reasoning before giving the answer. 5. When representing the identified objects in your answer, you should use the exact same words that appear in the caption. Below are a few examples:
In-Context	## Caption: A man in a white shirt and blue shorts swinging a tennis racket.
Example 1	### Parsing the description into objects From this caption, we can identify the following objects: A man, A white shirt, Blue shorts, A tennis racket.
	### Hierarchy and relationships- "A man" is the main subject, and his clothing (a white shirt and blue shorts) is part of his description.- "A tennis racket" is being swung by the man, so it will be in motion, likely extending from one side of the man.
	 ### Arranging objects on the canvas The canvas is in a size of width and height of 512, with the origin at the top-left (0, 0). The man should be centrally located to be the focus, with space around him to show the movement of swinging the racket. The clothing (white shirt and blue shorts) is part of the man's bounding box, with the shirt on the upper part of the torso and the shorts below. The tennis racket, since in motion, should extend out from the man's hand, likely to the right side if we imagine the swing.
	 ### Reasoning and concretizing ambiguity The exact positions of the shirt and shorts within the man's bounding box are based on their natural position on the body. The tennis racket's position is determined by the typical posture of swinging, which generally extends to the side and slightly upward.
	 ### Specifying locations "A man" will have a bounding box that covers a significant portion of the canvas to show his presence and the action. "A white shirt" will have a bounding box within the upper half of the man's bounding box, representing the torso area. "Blue shorts" will have a bounding box below the shirt's, indicating the lower part of the torso and upper legs. "A tennis racket" will have a bounding box that overlaps with the man's hand and extends outward to represent the swing.
In-Context	<pre>### Answer - **A man**: visual [[158, 51, 337, 404]] - **a white shirt**: visual [[204, 153, 317, 256]] - **blue shorts**: visual [[220, 235, 327, 307]] - **a tennis racket**: visual [[153, 46, 235, 143]]</pre>
Ouery	Now given the caption below, can you give a similar reasoning and derive the resulting bounding box for
2	those objects? then give the answer, strictly following the format of the answer given in the examples.
	## Caption: a red apple and a blue bird.
	###

Table 3: Prompts for LLMs to generate layout for caption "a red apple and a blue bird.".

Caption: A glass bowl full of oranges and apples.

CoT variant 1

CoT variant 2

Identifying Objects

From this caption, we can identify the following objects: A glass bowl, oranges, and apples.

Specifying Locations

- The glass bowl, being the central object, will have a bounding box in the middle of the canvas, perhaps taking up a significant area but not touching the edges to allow for visual clarity.

- The oranges and apples will each have their own bounding box within the bowl. Since they are grouped together, their boxes may overlap or be side by side. ### Identifying Objects

From this caption, we can identify the following objects: A glass bowl, oranges, and apples.

Hierarchy and Relationships

- The glass bowl serves as the container for the oranges and apples.

Arranging objects on the canvas

- Canvas Size: 512x512 square with the origin at the top-left (0, 0).

- Bowl Placement: Centrally on the canvas to emphasize its role as a container.

- Fruit Placement: Oranges and apples inside the bowl, possibly overlapping or side by side.

Reasoning and concretizing ambiguity

- Quantity of Fruit: Assuming a reasonable number, such as three oranges and three apples.

- Fruit Arrangement: Random scattering within the bowl.

Specifying Locations

- The glass bowl, being the central object, will have a bounding box in the middle of the canvas, perhaps taking up a significant area but not touching the edges to allow for visual clarity.

- The oranges and apples will each have their own bounding box within the bowl. Since they are grouped together, their boxes may overlap or be side by side.

Answer

- **A glass bowl**: visual [[153, 153, 358, 358]]

- **Oranges**: visual [[179, 179, 230, 230], [281, 179, 332, 230], [230, 281, 281, 332]]

- **Apples**: visual [[179, 281, 230, 332], [281, 281, 332, 332], [256, 204, 307, 256]]

Table 4: An in-context example with different CoT variants.

We can identify the following objects: A glass bowl, oranges, apples. There is no specified number of oranges and apples, so we assume a reasonable number that could fit within a bowl, such as three of each. The "glass bowl" is the container for the other objects, meaning the oranges and apples are within it. The "oranges and apples" are grouped together as they are contained in the bowl. As a result, the glass bowl will take a central position on the canvas, with a large bounding box, and there will be multiple bounding boxes for oranges and apples within the bowl.

CoT variant 3

since Stable Diffusion uses a CLIP model to encode the prompt, we find such correspondence after the text prompt and the object description are both tokenized. Suppose we have "a red apple" matched the (i, i + 1, ..., j) tokens in the tokenized text prompt, and the top-left coordinate being (x_1, y_1) , the bottom-right coordinate being (x_2, y_2) , we then will set 854

$$\mathbf{M}\big[\lfloor 64x_1 \rfloor : \lfloor 64x_2 \rfloor, \lfloor 64y_1 \rfloor : \lfloor 64y_2 \rfloor, i : j\big] = 1.$$

$$\tag{7}$$

Note that **M** is initialized with all zeros before composing the layouts on it. $\lfloor \cdot \rfloor$ is the floor operation. For any token index *i* that does not represent an object, we set the cross-attention mask $\mathbf{M}[:,:,i] = 1$.

A.4 Classfier-Free Guidance

Recall that in § 4.2.2, while we estimate the predictive noise using

$$\tilde{\epsilon} = \epsilon_{\theta}(\mathbf{z}_{t}, t, \emptyset) + g_{1}(\epsilon_{\theta}(\mathbf{z}_{t}, t, c(y)) - \epsilon_{\theta}(\mathbf{z}_{t}, t, \emptyset))$$

$$+g_2(\epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y), \mathbf{M}) - \epsilon_{\theta}(\mathbf{z}_t, t, c(y))), \tag{8}$$

we also explore other possible choices. The first alternative is the one used by Li et al. (2023), which jointly considers the text and the layout modality and has

$$\tilde{\epsilon} = \epsilon_{\theta,\phi}(\mathbf{z}_t, t, \emptyset) + g(\epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y), \mathbf{M}) - \epsilon_{\theta,\phi}(\mathbf{z}_t, t, \emptyset)).$$
(9)

For this alternative, we drop both modalities at the same time with a probability of 10% during the training. The second alternative regarding the choice between using score networks ϵ_{θ} and $\epsilon_{\theta,\phi}$. Intuitively, one should use $\epsilon_{\theta,\phi}$ for all combination of input modality, which yields

$$\tilde{\epsilon} = \epsilon_{\theta,\phi}(\mathbf{z}_t, t, \emptyset) + g_1(\epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y)) - \epsilon_{\theta,\phi}(\mathbf{z}_t, t, \emptyset)) + g_2(\epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y), \mathbf{M}) - \epsilon_{\theta,\phi}(\mathbf{z}_t, t, c(y))).$$
(10)

However, we empirically found that the setting in Eq. 8 works slightly better than Eq. 10 in terms of generated image quality. And both of Eq. 8 and Eq. 10 demonstrate better performance than the one using Eq. 9. We report the FID score of those settings in the experiment proposed by Appendix C.2.

A.5 Details for mIoU Computation

In § 5.2, we measure how the LLM-generated layouts resemble the ground-truths via mIoU. The mIoU score directly computed between two sets of bounding boxes is extremely low since first, the object labels are open-set, and second, there are many possible layouts for a given caption. To better match the bounding boxes, we make two modifications to the original matching algorithms to increase the reliability of the metric.

Relaxed object matching. While current tools for mIoU computation find correspondence between objects from the GT and LLM-generated layouts by matching their noun entities, we build such correspondence by checking whether one is a substring of the other. For example, while the GT layouts describe the phrase "a woman in a blue shirt" with one bounding box, an LLM might provide two bounding boxes for "a woman" and "a blue shirt" respectively. Our approach relates both the bounding boxes of "a woman" and "a blue shirt" to the GT bounding box for "a woman in a blue shirt".

Bounding box flipping. For each image, we compute two sets of IoU values, one with the GT bounding boxes and one with the horizontally flipped ones. We then take the set that has a higher mIoU score as the result. We only perform horizontal flips on the bounding boxes since it does not change the spatial sense of an image.

A proper metric used to measure the layout generative performance under the open-set setting is still underexplored. While our proposed method is shown effective, there might exist a principal solution to such a problem. We would like to leave it to the future work.

Model	# Parameters (in billions)
Stable Diffuison	1.06B
GLIGEN	1.27B
LACA	1.12B
LACA+LASA	1.16B

Table 5: Model parameters.

893 A.6 Model Size

897

900

901

902

903

904

905

906

907

908

909

910

911

912 913

914

915

916

917

918

919

920

921

922

923

925

We list the number of model parameters in Table 5. The architecture of adapter LACA+LASA is detailed in Appendix B. Our proposed adapters have significantly fewer parameters compared to GLIGEN.

B Layout-Aware Self-Attention Module

Inspired by LACA, we further investigate a possible variant of the proposed adapter - Layout-Aware Self-Attention Module, dupped LASA. In this section, we first demonstrate the development of LASA, then we show how to jointly compose the LACA and LASA adapters in the Stable Diffusion model. The proposed LASA adapter aims to make sure an object has coherent visual features during the generation. The object coherence is enforced by the layout modality – visual features that belong to the same object should self-attend to each other. Similar to LACA, the *l*-th LASA adapter injects such spatial information explicitly through the self-attention mask SM^l . Note that $SM^l \in \mathbb{R}^{p_l^2 \times p_l^2}$ is designed to specify whether the visual features should attend to one another or not. We omit the superscript *l* in the following discussion.

Constructing self-attention mask from layout. We propose to compose **SM** from the layout. First, we make an assumption that there are K objects depicted by the layout. Note that objects that share the same noun entities are considered differently, for example, "four apples and an orange" leads to K = 5. Second, we need to define the flattened visual index set \mathcal{I}_k for each object k:

$$\mathcal{I}_k = \{pi + j | (i, j) \text{-th visual feature belongs to object } k\}$$

Then, we can compose **SM** using the following rules:

• if $i \in \bigcup_k \mathcal{I}_k$, then $\mathbf{SM}[i, j] = 1$ if we have i and j assigned to the same object k, otherwise 0. Mathematically, the condition can expressed as $\sum_{k=1}^{K} \mathbf{1}[i \in \mathcal{I}_k] \mathbf{1}[j \in \mathcal{I}_k] > 0$

• if $i \notin \bigcup_k \mathcal{I}_k$, then we have $\mathbf{SM}[i, :] = 1$.

Here $1[\cdot]$ is an indicator function. The intuition of rule 1 is that a visual feature will aggregate information from all other visual features that share the same objects. Rule 2 allows non-object visual features to aggregate information from all the others. We highlight how to compose such an attention mask in Alg.1.

Module designs. Similar to LACA, we directly initialize LASA using the self-attention weight from the Stable Diffusion models. We only apply LASA to the low-resolution visual maps (the intermediate visual feature map with resolutions 16x16 and 8x8), which are more computationally affordable. We also add a zero convolution layer on top of LASA's output.

Composing LACA and LASA adapters. For each transformer block, we add a LASA module in between the LACA module and the cross-attention module. We only add LASA to the last two downsampling blocks and the first two upsampling blocks.

C Additional Results

927 C.1 Ablation Study on CoT Variants

The CoT strategy is directly linked to the performance of the generated layout. Following § 5.2, we assess the effectiveness of different CoT variants by evaluating the performance of the LLM-generated layout. In

Algorithm 1 Construction of self-attention mask SM.

Input: *K* index sets $\mathcal{I}_1, \ldots, \mathcal{I}_K$, and an all-zeros tensor $\mathcal{M} \in \mathbb{R}^{p^2 \times p^2 \times K}$. [Composing self-attention mask for each object] for k = 0, ..., K - 1 do Set $\mathcal{M}[i, j, k] = 1, \forall i, j \in \mathcal{I}_k$ Set $\mathcal{M}[i, :, k] = \mathbf{1}, \forall i \notin \mathcal{I}_k$ end for [Reducing \mathcal{M} to SM] Intialize $\mathbf{SM} = \mathcal{M}[:,:,0]$ for k = 1, ..., K - 1 do Obtain non-zero index set $\mathcal{I}_{\emptyset,k} = \{i | \mathcal{M}[i,:,k] \neq 1\}$ Obtain non-zero index set $\mathcal{I}_{\emptyset, SM} = \{i | SM[i, :, k] \neq 1\}$ For $i \in \mathcal{I}_{\emptyset,k} \cap \mathcal{I}_{\emptyset,\mathbf{SM}}$, we set $\mathbf{SM}[i,:] = \mathbf{SM}[i,:] \text{ or } \mathcal{M}[i,:,k]$ For $i \in \mathcal{I}_{\emptyset,k} \setminus (\mathcal{I}_{\emptyset,k} \cap \mathcal{I}_{\emptyset,\mathbf{SM}})$, we set $\mathbf{SM}[i,:] = \mathcal{M}[i,:,k]$ end for Output: Self-attention mask SM

CoT strategy	hit rate	mIoU	FID
no CoT	97.1%	19.84	31.95
variant 1	97.4 %	21.31	31.64
variant 2	97.4 %	23.98	28.96
variant 3	96.9%	19.28	32.49

Table 6:	Layout	performance of	different (СоТ	variants o	n Flickr30K.
----------	--------	----------------	-------------	-----	------------	--------------

particular, We report the object hit rate, mIoU, and the FID score of generated images in Table 6. The 930 second variant, which we use in our major experiment, has shown superior performance over others. 931 Surprisingly, variant 3, which does not separate each reasoning step explicitly, performs even worse than 932 the one without CoT reasoning. 933

C.2 Ablation Study on Classifier-Free Guidance

Performance vs CFG variants. We investigate the generative performance under different CFG guidances (Eq. 8, Eq. 9 and Eq. 10). We report the FID score on Flickr30K in Table 7. Empirically, 936 we observe that the employed CFG guidance works the best among others. We only incorporate layout 937 modality into SD via the mentioned CFG guidances for the first 20% denoising steps. 938

CFG guidance	FID
Eq. 8	28.96
Eq. 9	30.20
Eq. 10	29.19

Cable 7: Generative performant	ice of different CFG	guidance on Flickr30K
--------------------------------	----------------------	-----------------------

Sensitivity Analysis. We explore how the hyperparameters g_1, g_2 from classifier-free guidance affect 939 the quality of the generated images. We choose Eq. 8 as our inverstigation target. We explore $g_1 \in$ 940 $\{1.5, 3.5, 5.5, 7.5\}$ and $g_2 \in \{3.5, 5.5, 7.5\}$ and report the FID scores in Table 8. We can observe that 941 $g_1 = g_2 = 5.5$ yields the best FID score. 942

935

			g	1	
		1.5	3.5	5.5	7.5
	3.5	27.11	26.34	26.78	28.16
g_2	5.5	26.45	23.80	23.28	23.28
	7.5	24.92	23.67	23.35	23.30

Table 8: Classifier-free guidance weights over text and text-layout conditions. g_1 controls the text-layout condition and g_2 controls the text-only condition.

943 C.3 Generative Performance of LASA Adapter

We validate the effectiveness of the LASA adapter on both the Flickr30K and COCO2017 datasets. This
study involved a comparative analysis of the LACA adapter alone and its integration with the LASA
adapter. We use LLM-generated layouts to generate images and evaluate the FID score, GLIP score, and
GLIP rate of the models. Table 9 shows that the combination of LASA with LACA resulted in enhanced
capabilities in producing more realistic images. Notably, there was an improvement of 2.54 and 1.15 in
performance on the Flickr30K and COCO2017 datasets, respectively. The measures of layout accuracy
and composition accuracy demonstrated that both adapters were comparably effective in integrating
objects into images. Although LASA contributes to higher-quality image generation, it also leads to
increased sampling time. In speed tests conducted on an A100 GPU, the sampling time averaged 2.5
seconds for LACA but extended to 5.2 seconds when utilizing LACA combined with LASA.

		Flick30K			
	EID	C	GLIP sco	CL ID rate	
	FID	AP	AP_{50}	AP_{75}	GLIP rate
LACA	28.96	58.5	68.1	59.4	83.3
LACA+LASA	26.42	59.4	68.4	60.0	83.3
	COCO2017				
	GLIP score				
	FID	AP	AP_{50}	AP_{75}	GLIP rate
LACA	23.28	58.8	67.3	50.1	78.1
LACA+LASA	22.13	58.4	68.9	49.7	78.2

Table 9: Generative performance of LASA adapter

C.4 Visualizations

953

954 955

957

959

960

961

964

We provide additional visualizations of the generated images in Fig. 7 and Fig. 8. Specifically, we showcase 6 caption examples and their generated images. For each caption, we use an LLM to generate two layouts. Then we generate two images from the layout. As we can observe, our proposed method can generate reasonable layouts most of the time. Note that the generated objects do not necessarily lie within the given bounding boxes. We hypothesize this is because the layout information is only injected via LACA at the early stage of the denoising process. When LACA is no longer employed, the Stable Diffusion model takes its liberty to compose the objects. We believe this is beneficial since the generative error from the layout can be alleviated by the Stable Diffusion model, thereby achieving higher image quality.

D Limitations and Societal Impact

Limitations While our proposed method empowers LLMs with the ability to generate object layouts
 via in-context learning, further finetuning may be a more effective strategy for generating high-quality
 layouts. Moreover, it is observed when injecting layout information, the FID score of images generated
 from Stable Diffusion variants is often worse than the FID of images generated only conditioned on text.



<u>A red book</u> and <u>a yellow vase</u>.



A dog is standing to the left of a parking meter.



(c)

Figure 7: Additional visualization 1.

Societal Impact Generative models unlock a range of creative uses, thus broadening access and encouraging wider exploration. However, this ease of access also raises concerns about the potential for generating and spreading altered data, misinformation, and spam. Moreover, there's the risk that these models might inadvertently expose the data they were trained on. This is particularly worrisome when that data includes sensitive or personal information gathered without clear consent. Finally, since our proposed method generated objects following the given layout, the generated objects will be disproportionated. While deep generative models are now becoming more prominent and are frequently used for intellectual creation, how to maintain the authenticity of the generated image conditioned on the layout remains a significant area for future research.

969
970
971
972
973
974
975
976



<u>A blue car</u> is <u>in front on a elephant</u>.



<u>A dog is jumping in the air looking at a frisbee</u>.



A dog stands and four balloons are in the air.

Figure 8: Additional visualization 2.

⁽f)