

Think Before You Diffuse: Reasoning-Augmented Scene Graphs for Structured Text-to-Image Generation

Anonymous Authors

Abstract

Text-to-image models can produce high-quality images while remaining brittle on prompts that require explicit object counts, attributes, or relations. We study whether an editable symbolic intermediate representation can bridge language-model-based prompt decomposition and scene-graph-conditioned diffusion. Our pipeline converts a caption into a reasoning-augmented scene program containing standard subject–relation–object triples together with global presence, attribute, count, and relational constraints, and then maps this program through a graph encoder into the conditioning space of a frozen SDXL generator. We construct a small aligned caption–graph–image fine-tuning set from Visual Genome and compare scene-graph-only fine-tuning, constraint augmentation, and constraint augmentation with graph–image contrastive alignment. In a compute-limited pilot evaluation, the contrastively aligned variant improves Inception Score from 3.68 to 5.31 and lowers FID from 293.6 to 223.5 relative to our scene-graph baseline, whereas constraint nodes alone yield only a small change. The explicit scene program also provides an editable state for iterative image revision; however, the present results are preliminary and do not yet establish improved logical or compositional correctness on a dedicated benchmark.

1 Introduction

Modern text-to-image diffusion models have substantially improved visual fidelity, but complex prompts remain difficult when correctness depends on structured requirements such as exact counts, attribute binding, or relations among multiple objects. Natural-language conditioning is expressive, yet it does not expose a representation that can be directly inspected or edited when the generated scene violates one of these requirements.

A complementary line of work conditions image generators on explicit structure. Scene graphs encode objects, attributes, and pairwise relations, and recent graph-conditioned diffusion systems demonstrate that this structure can improve semantic control (Yang et al., 2022; Farshad et al., 2023; Li et al., 2024). Other approaches use large language models (LLMs) to improve prompt understanding before image generation, for example by producing layouts or richer prompt representations (Lian et al., 2024). Reasoning-oriented diffusion work has separately explored chain-of-thought-style computation in text and multimodal settings (Ye et al., 2024; Mi et al., 2025). These directions motivate a natural question: *can prompt-level reasoning be converted into an explicit symbolic program that is directly consumed by a structured image generator?*

We investigate this question with a two-stage text-to-image pipeline. The first stage maps a prompt to an augmented scene graph, which we call a *scene program*. In addition to standard subject–relation–object triples, the program contains four classes of global constraints: object presence, attributes, counts, and relations. A two-pass frozen LLM proposes graph content, while deterministic extractors preserve explicit constraints from the input prompt. The second stage encodes the resulting graph using the scene-graph encoder of an SDXL-based graph-conditioned generator and uses the resulting embeddings to condition a frozen diffusion backbone.

Our study makes three concrete contributions:

- We introduce an explicit intermediate representation that augments scene graphs with global constraint nodes designed to preserve prompt-level requirements that are awkward to represent as ordinary local triples.
- We implement a prompt-to-program pipeline that combines LLM decomposition with deterministic constraint extraction, yielding an intermediate state that can be inspected and edited independently of image generation.
- We present a controlled pilot ablation of graph-only fine-tuning, constraint-node fine-tuning, and graph-image contrastive alignment on Visual Genome. The results show a substantial improvement from contrastive alignment in our setting, while also showing that constraint nodes by themselves do not measurably improve FID/IS at the current scale.

The last point is important for interpreting the paper. Our current experiments are compute-limited, use a small fine-tuning set and a small evaluation sample, and do not include a dedicated constraint-satisfaction benchmark. We therefore treat the quantitative results as evidence about the feasibility of the representation and conditioning pipeline rather than as a claim of state-of-the-art text-to-image generation.

2 Related Work

2.1 Scene Graphs and Structured Image Generation

Scene graphs represent a visual scene as objects connected by labeled relations, optionally with object attributes. Surveys of scene graph generation emphasize both their utility as structured semantic representations and the difficulty of recovering complete relational structure from pixels or language (Zhu et al., 2024; Chang et al., 2023). Visual Genome (VG) provides dense object, attribute, and relationship annotations and is a standard source of scene-graph supervision (Krishna et al., 2017).

Several image-generation methods exploit this structure directly. SGDiff learns scene-graph embeddings with masked reconstruction and contrastive objectives and conditions a latent diffusion model on the learned representation (Yang et al., 2022). SceneGenie converts prompts into graph-structured geometric guidance and enforces bounding-box and segmentation constraints during diffusion sampling (Farshad et al., 2023). LAION-SG scales structural annotation to a large image-text corpus and trains an SDXL-based model that consumes scene-graph information (Li et al., 2024). Earlier compositional diffusion work also shows that decomposing generation into separately represented semantic factors can improve structured composition (Liu et al., 2022).

Our work differs in focus: rather than treating the scene graph as a complete input specification, we study the interface between an underspecified natural-language prompt and a graph-conditioned generator. We augment ordinary graph triples with explicit global constraints and retain this representation as an editable program between prompt interpretation and image generation.

2.2 Reasoning Before Generation

LLM-Grounded Diffusion uses an LLM to convert complex prompts into spatial layouts that guide a pretrained diffusion model, substantially improving numeracy and spatial prompt following without retraining the base generator (Lian et al., 2024). Diffusion-of-Thought studies chain-of-thought reasoning in diffusion language models, while ThinkDiff transfers multimodal in-context reasoning capabilities into a diffusion-based image-generation setting (Ye et al., 2024; Mi et al., 2025). These works motivate reasoning before or within generation, but the reasoning state is not the same as an editable scene graph with explicit constraint types.

We use an LLM as a front-end decomposition module rather than claiming that the image diffusion process itself performs chain-of-thought reasoning. This distinction is central to our formulation:

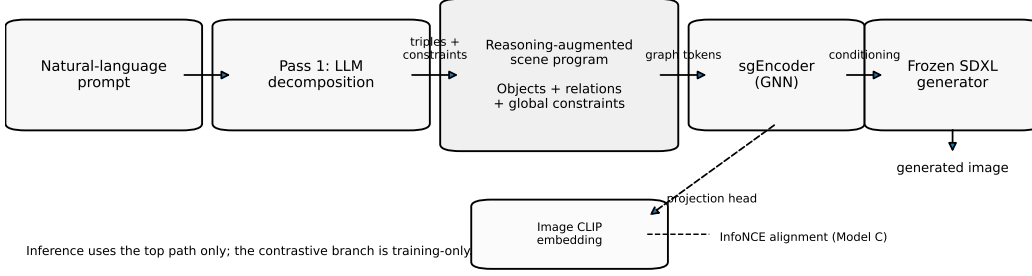


Figure 1: System overview. A natural-language prompt is decomposed into an explicit, constraint-augmented scene program. The scene program is encoded by the trainable graph encoder and used to condition a frozen SDXL generator. Model C additionally uses a training-only graph-image contrastive objective; inference follows the top path only.

the desired output of the reasoning stage is a symbolic object that can be inspected, modified, and re-used.

3 Method

3.1 Problem Formulation

Given a natural-language prompt x , our goal is to generate an image I that is semantically consistent with x while preserving explicit structured requirements. We factor generation into

$$x \xrightarrow{f_{\text{prog}}} G^+ \xrightarrow{f_{\text{img}}} I, \quad (1)$$

where G^+ is a reasoning-augmented scene graph. We write

$$G^+ = (V, E, C), \quad (2)$$

with ordinary scene-graph nodes V , labeled edges E , and a set of global constraint nodes C . This factorization makes the intermediate representation independently observable and editable.

3.2 Constraint-Augmented Scene Programs

Standard scene graphs are dominated by local object–relation–object triples. We extend them with four constraint types extracted from the prompt:

- **Presence:** an object must appear in the generated scene;
- **Attribute:** an object must possess an explicitly stated property such as color or material;
- **Count:** a lower-bound or explicit multiplicity requirement on an object category;
- **Relation:** a required spatial or functional relation between entities.

The implementation represents each constraint using the same triple interface consumed by the graph encoder. Example encodings include `[MUST_HAVE] --requires--> car`, `[ATTR] --describes--> red car`, and relation-specific marked edges. By expressing constraints as graph nodes or edges rather than as a separate text vector, all structured information participates in the same message-passing computation.

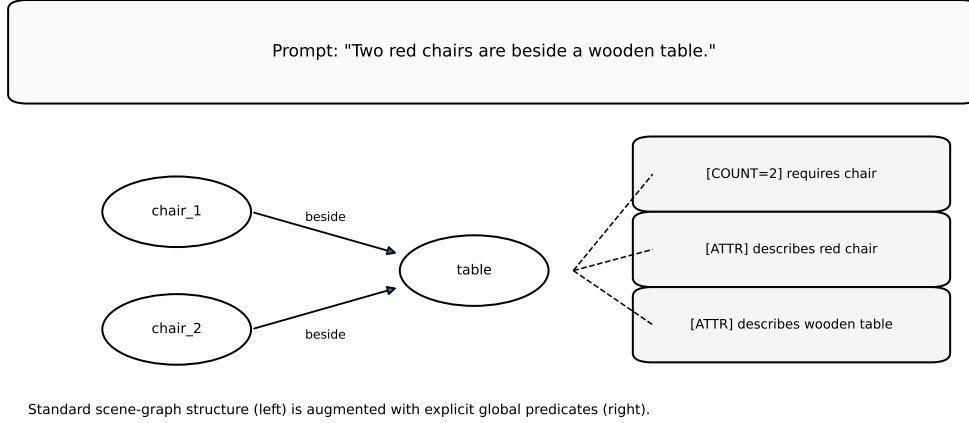


Figure 2: Illustrative scene-program representation. Ordinary object/relation structure is augmented with explicit global predicates such as count and attribute requirements. This is a schematic example of the representation, not an example from the evaluation set.

3.3 Prompt-to-Program Generation

Our program generator combines deterministic extraction with a two-pass frozen LLM. In the first pass, few-shot examples instruct the model to decompose the prompt into explicitly mentioned entities, plausible implied entities, salient attributes, and pairwise relations. This intermediate reasoning trace is used only to construct the symbolic program. In the second pass, the LLM receives the original caption and the decomposition and emits a set of subject–relation–object triples. Finally, four deterministic extractors scan the original caption and add any explicit presence, attribute, count, or relational constraints that should not depend on free-form generation.

This hybrid design separates two roles. The LLM supplies contextual completion when the text under-specifies a plausible scene, while deterministic rules preserve requirements that are directly stated by the user. The resulting G^+ is passed to the image generator and can also be edited directly during interactive refinement.

3.4 Graph-Conditioned Diffusion

Our image generator builds on the SDXL-based scene-graph conditioning architecture released with LAION-SG (Li et al., 2024; Podell et al., 2024). SDXL’s U-Net, VAE, and text encoders are frozen. The trainable component is a scene-graph encoder (`sgEncoder`) that converts graph triples into the token-level conditioning representation expected by SDXL.

For each triple component, the encoder tokenizes its textual label with SDXL’s frozen CLIP text encoders (Radford et al., 2021), projects the resulting features into a lower-dimensional graph space, and applies `GraphTripleConv`/`GraphTripleConvNet` message passing. The graph features are then projected back into SDXL’s conditioning space, producing token-level and pooled embeddings that replace the usual caption conditioning. Because constraint information is encoded inside G^+ , the same graph encoder can process ordinary nodes and constraint nodes without introducing a new cross-attention pathway.

Contrastive alignment. In addition to diffusion training, our strongest variant adds a projection head and an InfoNCE-style graph–image alignment loss. Conceptually, the training objective is

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \quad (3)$$

where $\mathcal{L}_{\text{diff}}$ is the standard diffusion objective and $\mathcal{L}_{\text{con}}$ encourages graph conditioning features to align with the corresponding image representation. The coefficient λ_{con} controls the relative contribution of graph–image alignment to the diffusion objective.

Why constraints are graph-native. We initially represented constraints as separate CLIP text embeddings concatenated with the scene-graph representation. Training loss stagnated under this design. Moving constraints into the graph itself produced a stable training trajectory, suggesting that the graph encoder’s relational inductive bias is a better interface for these signals than an additional weak text-conditioning stream.

4 Data and Experimental Setup

4.1 Data Construction

We use Visual Genome for its dense object, attribute, and relation annotations (Krishna et al., 2017). Because the desired training unit is a (text, graph, image) triplet, we construct captions for VG images using BLIP-Large (Li et al., 2022). Raw captions often omit graph-annotated objects, so a second language-model pass refines each caption using the BLIP description together with allowed object information from the VG scene graph. This produces a small aligned corpus for fine-tuning the prompt-to-program-to-image pipeline.

This construction is pragmatic but introduces a limitation: the training captions are synthetic and are partially informed by the target graph. As a result, they are cleaner and more graph-aligned than unconstrained user prompts. We return to this issue in Section 6.

4.2 Model Variants

We evaluate three variants:

1. **A: Graph baseline.** Fine-tune the released LAION-SG graph encoder on 1,000 VG samples for 100 epochs.
2. **B: + constraints.** Continue fine-tuning with approximately 2,000 VG images for 50 epochs, augmenting each scene graph with the constraint nodes described above.
3. **C: + constraints + contrastive alignment.** Use the same 2,000-image training setup as B and add the graph–image contrastive objective.

All SDXL components remain frozen; only the graph-conditioning pathway is trained. The experiments ran on an AWS EC2 g5.2xlarge instance with one NVIDIA A10G GPU and 8 vCPUs. Model A required approximately 25 hours of training; Models B and C required approximately 26.5 hours each.

4.3 Optimization and Evaluation

We use AdamW with learning rate 5×10^{-4} , weight decay 0.2, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, batch size 4, and four gradient-accumulation steps. The graph encoder uses embedding dimension 512 and num_graphs=5, following the LAION-SG training configuration where possible (Li et al., 2024).

Table 1: Training variants used in the pilot study. Dataset sizes and times are those reported in the original experiments.

Variant	VG images	Epochs	Constraints	Time
A	1,000	100	no	25 h
B	~2,000	50	yes	26.5 h
C	~2,000	50	yes + contrastive	26.5 h

Table 2: Pilot Visual Genome evaluation. All three rows use our training/evaluation pipeline; lower FID and higher IS are better. The evaluation contains only 150 generated images, so the absolute FID values are high-variance.

Variant	IS $\uparrow$	FID $\downarrow$
A: Graph baseline	3.684 ± 0.688	293.63
B: + constraint nodes	3.782 ± 0.679	286.25
C: + contrastive alignment	5.312 ± 0.785	223.51

We report Fréchet Inception Distance (FID; lower is better) and Inception Score (IS; higher is better) as coarse measures of image distribution quality. The available evaluation used only 150 generated images, so the absolute FID values should be interpreted cautiously. This evaluation also does not directly measure whether presence, count, attribute, or relation constraints are satisfied. Inference takes approximately 4.5–5.5 seconds per image in the reported setup.

5 Results

Table 2 shows two clear patterns. First, adding constraint nodes alone produces only a small change relative to the graph-only baseline: IS increases from 3.684 to 3.782 and FID decreases from 293.63 to 286.25. On the metrics currently available, this is not evidence that the constraint mechanism itself improves generation quality.

Second, the contrastively aligned variant produces a much larger change, increasing IS to 5.312 and reducing FID to 223.51. This supports the narrower hypothesis that aligning learned graph representations with image representations improves their compatibility with the frozen SDXL generator in our setting. It does *not* by itself establish better compositional reasoning, because neither FID nor IS measures satisfaction of the program’s symbolic constraints.

Absolute performance remains substantially below published scene-graph-to-image systems trained at larger scale (Yang et al., 2022). Direct numerical comparison is not appropriate: prior graph-to-image models receive fully specified scene graphs, whereas our end-to-end pipeline must infer an intermediate program from text, and our experiments use much less training and evaluation data.

5.1 Editable Intermediate State

A practical benefit of the scene program is that it can persist across interaction. We implemented a simple editing interface that stores the current graph, maps user requests such as “add a window” or “change the couch color” to local graph edits, and re-runs only the graph-conditioned generation stage. This demonstrates the intended interface property: generation is mediated by an explicit symbolic state rather than by an opaque prompt rewrite alone.

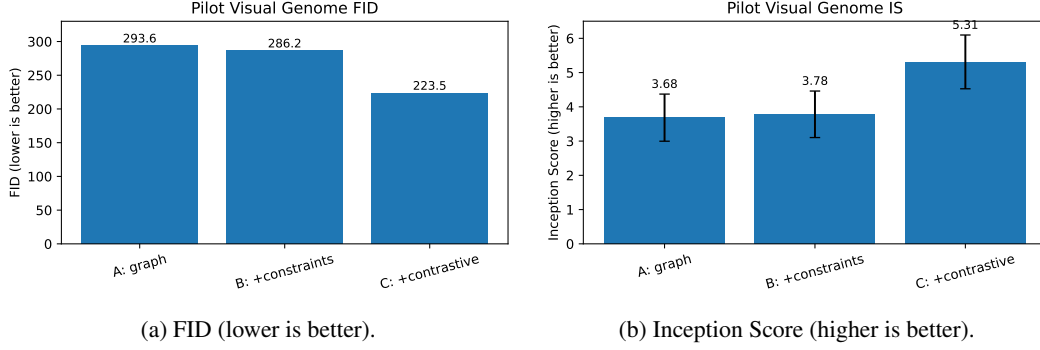
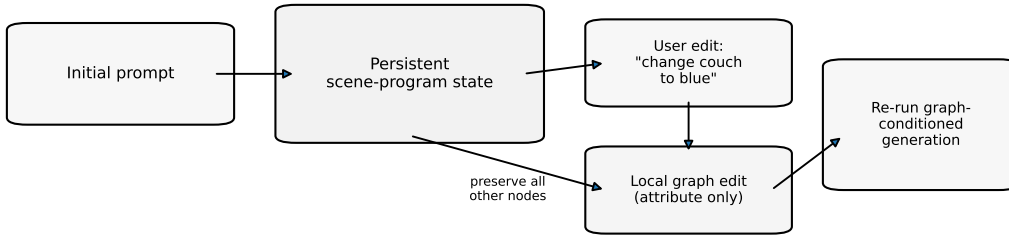


Figure 3: Visualization of the pilot results in Table 2. The evaluation contains only 150 generated samples, so these plots should be read as a visualization of the reported pilot numbers rather than a precise benchmark estimate.



Schematic only: this illustrates the state-editing interface and does not depict an experimental output image.

Figure 4: Schematic of graph-mediated interactive editing. A user request is applied as a local edit to the persistent scene-program state, after which graph-conditioned generation is rerun. The diagram illustrates the interface; it is not presented as an experimental output sample.

6 Limitations and Required Evaluation

The current experiments establish an implementation path but leave several scientific questions open. First, the training set is small and none of the variants was trained to full convergence. Second, FID was estimated from only 150 generated samples, which makes it unsuitable for fine-grained comparison. Third, the central motivation of the method is structured correctness, yet the current evaluation does not measure count accuracy, attribute binding, relation satisfaction, scene-program consistency, or prompt faithfulness. A conference-scale evaluation should therefore include targeted compositional benchmarks and program-to-image consistency metrics in addition to distributional image metrics.

Fourth, the VG captions used for training are synthetic and are refined with information from the ground-truth scene graph. This makes them useful for constructing aligned triplets but may underestimate the difficulty of mapping naturally occurring user prompts to complete scene programs. Fifth, the prompt-to-program LLM and heuristic rules were selected through manual qualitative iteration; their contribution has not yet been isolated with a systematic ablation. Finally, the graph generator and diffusion model are not trained end-to-end, so errors introduced by the program generator cannot be corrected using downstream image-level supervision.

These limitations constrain the claims we make from the present results. The strongest supported conclusion is that an explicit, editable program can be integrated into a graph-conditioned SDXL

pipeline and that contrastive graph–image alignment materially improves the resulting conditioning representation in our pilot study.

7 Conclusion

We presented a reasoning-aware text-to-image pipeline that converts natural-language prompts into editable, constraint-augmented scene graphs before diffusion-based image generation. The representation extends ordinary scene graphs with presence, attribute, count, and relational constraints and is consumed by an SDXL-compatible graph encoder. In our pilot ablation, constraint nodes alone produce little change in FID/IS, whereas adding graph–image contrastive alignment produces a substantially stronger result. The current evidence is preliminary, but it motivates a more rigorous evaluation of symbolic intermediate representations as interfaces between language-model reasoning and generative image models.

Ethics Statement

This work studies structured control for image generation. More precise controllability can improve interpretability and editing, but it can also enable more precisely specified synthetic or misleading imagery. The present system does not introduce new provenance, watermarking, or misuse-prevention mechanisms. We therefore view provenance and appropriate content-safety safeguards as important considerations for any deployment.

Reproducibility Statement

Section 4 reports the model variants, optimization settings, hardware, training durations, and evaluation protocol used in the current experiments. Appendix A describes the preprocessing and constraint encodings. For a final submission, we will additionally report random seeds, exact data splits, contrastive-loss hyperparameters, and an anonymous code artifact sufficient to reproduce the reported models and evaluation.

AI Use Statement

Generative AI tools are part of the research method described in this paper, including LLM-based caption refinement and prompt-to-scene-program generation. Generative AI was also used to assist with editing and restructuring the manuscript for readability and conference formatting. The authors reviewed the resulting text and remain responsible for all claims, experiments, citations, and artifacts in the submission.

References

- Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1–26, 2023. doi: 10.1109/TPAMI.2021.3137605.
- Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Bjorn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 88–98, 2023. doi: 10.1109/ICCVW60793.2023.00016.

- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2017. doi: 10.1007/s11263-016-0981-7.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.
- Zejian Li, Chenye Meng, Yize Li, Ling Yang, Shengyuan Zhang, Jiarui Ma, Jiayi Li, Guang Yang, Changyuan Yang, Zhiyuan Yang, Jinxiong Chang, and Lingyun Sun. LAION-SG: An enhanced large-scale dataset for training complex image-text models with structural annotations. *arXiv preprint arXiv:2412.08580*, 2024.
- Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024.
- Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.
- Zhenxing Mi, Kuan-Chieh Wang, Guocheng Qian, Hanrong Ye, Runtao Liu, Sergey Tulyakov, Kfir Aberman, and Dan Xu. I think, therefore i diffuse: Enabling multimodal in-context reasoning in diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44017–44036. PMLR, 2025.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.
- 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*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.
- Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. Diffusion-based scene graph to image generation with masked contrastive pre-training. *arXiv preprint arXiv:2211.11138*, 2022.
- Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thoughts: Chain-of-thought reasoning in diffusion language models. *arXiv preprint arXiv:2402.07754*, 2024.
- Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, and Mohammed Bennamoun. Scene graph generation: A comprehensive survey. *Neurocomputing*, 566:127052, 2024. doi: 10.1016/j.neucom.2023.127052.

A Implementation Details

The preprocessing pipeline contains scripts for VG caption generation, caption/scene-graph alignment, conversion of VG annotations into the LAION-SG JSON representation, heuristic constraint extraction, LLM-based caption-to-graph generation, and the interactive editing interface. The constraint encodings used in the implementation include:

- `presence: [MUST_HAVE] --requires--> car;`
- `attribute: [ATTR] --describes--> red car;`
- `relation: man --[MUST_BE] on--> skateboard;`
- `count: [COUNT>2] --requires--> person.`

The training code adds a projection head and InfoNCE-style contrastive term to the diffusion objective. The exact contrastive-loss weight, temperature, negative-sampling procedure, random seeds, and train/validation/test split are implementation details that should be reported alongside the final experimental configuration.

B Development Ablations

We initially attempted to reproduce SGDiff directly. The available unofficial checkpoint was reported as having been trained for only 150 epochs and produced poor samples in our setup, motivating a pivot to LAION-SG’s modular SDXL-based conditioning pathway (Yang et al., 2022; Li et al., 2024). We also tested a separate constraint-conditioning stream using concatenated CLIP embeddings; its training loss stagnated. Encoding constraints as graph-native nodes was substantially more stable and became the design used in Models B and C.

During heuristic development, the average number of extracted constraints per image increased from approximately 0.6 to 1.6 through manual rule and prompt refinement. This statistic was used only as a diagnostic to avoid producing graphs with effectively no global constraints; it is not an evaluation metric for semantic correctness.