Scene-Graph Guided Conditioning for Text-to-Image Generation

Hirunima Jayasekara, Christabel Acquaye, Ashira Jayaweera

Motivation

Diffusion models produce photorealistic images, but they still fail on compositional prompts: multiple objects, attributes, and spatial relations.

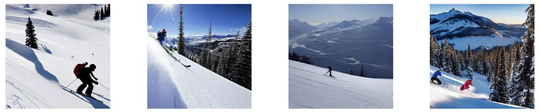
- Objects are omitted, fused, or assigned the wrong color / size.
- Relations such as “cat under a table” or “person smaller than an airplane” are not reliably realized.
- CLIP-style text encoders capture semantics, not structured relations.
- Layout-conditioned methods (GLIGEN, MultiDiffusion) improve placement but lock objects to boxes and reduce diversity.

Human scene descriptions are relational. Fixed boxes encode geometry, not “who does what to whom.”

We ground generation in a scene graph; objects as nodes, attributes and relations as edges: so the model can keep relational structure without a hard layout map.

A woman is standing in front of a mirror, blow-drying her long hair with a white, handheld hair dryer.

Two skiers are zipping down a powdery mountain slope, their skis carving graceful arcs in the snow.



Stable Diffusion | Attend and Excite | Attention Refoc | Ours

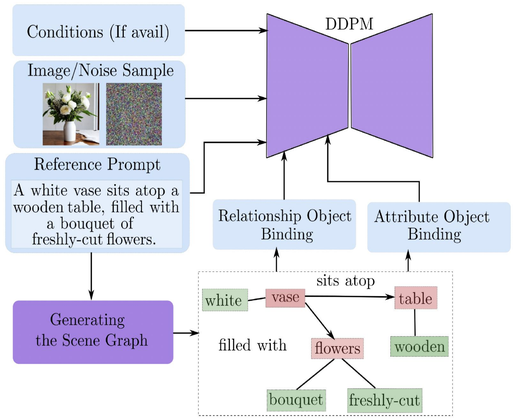
Contributions

- Introducing a novel approach incorporating scene graph-based conditioning to enhance spatial comprehension and diversify image generation outcomes.
- The proposed Graph Fusion layer addresses the challenge of integrating spatial information into the image synthesis process, even in the absence of additional conditional inputs to guide generation. Optional compatibility with layout tokens (GLIGEN-style) when boxes are available.
- Proposed method demonstrate competitive performance on two benchmarks.

Model

We keep a pretrained latent diffusion backbone and add a structured side path. The caption is parsed into a scene graph, two Graph Attention Networks turn that graph into object and relation embeddings, and a Graph Fusion layer injects those embeddings into every U-Net block. Layout maps are optional extras, not the main signal.

Text still sets global style and wording. The graph answers a different question: which object owns which attribute, and how two objects interact. Fusion blends the two so the denoiser can place objects freely while still respecting relations such as “sits atop” or “filled with,” instead of snapping every entity to a pre-drawn box.



Algorithm 1: Scene Graph-Guided Text-to-Image Synthesis: This algorithm details the generation process, including scene graph extraction from text, scene graph embedding, fusion with text and latent image representations, and diffusion-based image synthesis.

Input: Prompt t, (optional) image x, (optional) conditions c
Output: Generated image $\hat{x}$

1. Scene Graph Extraction
Use LLM to map $t \mapsto G = (O, A, R)$,
where $O = \{o_i\}$ (objects), $A = \{a_i\}$ (attributes), $R = \{r_{ij}\}$ (relations)
2. Scene Graph Embedding
foreach $o_i \in O$ **do**
Obtain embedding $\mathbf{o}_i = \mathrm{CLIP}(o_i)$
foreach $a_i \in A$ *attached to* o_i **do**
Fuse attributes: $\tilde{\mathbf{o}}_i = \mathrm{GAT}_{\mathrm{obj}}(\mathbf{o}_i, \mathbf{a}_i)$
foreach $(o_i, r_{ij}, o_j) \in R$ **do**
Update: $\tilde{\mathbf{o}}_i = \mathrm{GAT}_{\mathrm{rel}}(\tilde{\mathbf{o}}_i, r_{ij}, \tilde{\mathbf{o}}_j)$
Let $\mathbf{G} = \{\tilde{\mathbf{o}}_i\}$
3. Text Embedding and Visual Latent Preparation
$\mathbf{h}_t = \mathrm{CLIP}(t)$
if *Training* **then**
$z = \mathrm{Encoder}(x)$
else
Sample $z_T \sim \mathcal{N}(0, I)$
4. Diffusion Process with Graph Fusion
for $t = T, \ldots, 1$ **do**
Integrate scene graph:
$z_{t-1} = \mathrm{DDPM}(z_t \mid \mathbf{h}_t, \mathbf{G}, c)$ via Graph Fusion Layer
(incorporate cross-attention and fusion as in Eq. (13)-(14))
5. Image Synthesis
$\hat{x} = \mathrm{Decoder}(z_0)$

Graph Fusion Architecture

Each U-Net block: self-attention and gated attention stay frozen (❄). Graph attention and Graph Fusion are trained (🔥). Fusion reweights spatial maps with graph objects and attribute kernels.

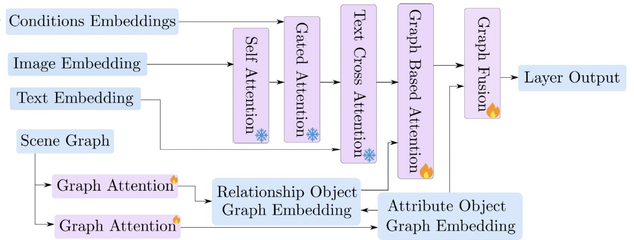
This keeps pretrained visual knowledge while steering tokens toward the right object–attribute regions, even when no layout is given.

Scene graph from text. GPT-4 extracts objects O, attributes A, and triplets (subject, predicate, object). Nodes are encoded with the same CLIP text encoder used by Stable Diffusion.

Two GAT stages. An Object GAT fuses each entity with its attributes. A Relationship GAT updates subject/object nodes with active vs. passive edge context so relations are not bag-of-words.

Graph Fusion Layer. Visual tokens attend to graph object embeddings. Those maps are blended with text cross-attention maps of the matching words, then convolved with attribute embeddings as kernels. Output is added back to the visual stream. Fusion (simplified). For each object j, mix text and graph attention, then convolve with attribute kernels:

Training. Frozen SD / GLIGEN backbone. Train GAT + GFL (+ gated attention if layouts are used) with the standard noise-prediction loss on Flickr + SBU + CC3M (~6M pairs), AdamW 5e−5, 200k steps, 4× A6000.



Results

Method	Counting (F1 ↑)	Spatial (Acc ↑)	Size (Acc ↑)	Color (Acc ↑)
Stable Diffusion [33]	58.31	8.48	9.18	12.61
Attend-and-excite [8]	60.47	9.98	10.58	19.56
Ours (w/o layout suggestions)	65.53	11.16	10.91	21.74
Layout-guidance [9]	56.22	16.47	12.38	14.39
MultiDiffusion [19]	55.18	14.27	10.58	17.15
GLIGEN [21]	66.58	30.74	26.75	18.78
Attention Refocus [27]	67.54	40.22	27.74	26.32
Ours (with layout suggestions)	71.92	28.15	29.34	17.54

We evaluate two settings that match how the model is used in practice: text + scene graph only and text + scene graph + layout boxes.

Compositional accuracy is measured on HRS (counting, spatial, size, color).

Image quality (FID ↓). On COCO-2014 val we obtain 18.92, better than Stable Diffusion (20.82), GLIGEN (20.63), and Attention Refocus (20.37). Prior grounded methods often train from scratch or clamp objects to boxes; we keep a large pretrained generator and only add graph layers, which preserves photorealism while still injecting structure. Lower FID with no layout is consistent with the claim that soft relational conditioning does not collapse sample diversity.

Across counting, size, spatial, and color prompts, our samples more often keep the right number of objects and relative scale and bind colors better than Stable Diffusion, while spatial layout stays coherent but not always box-precise and crowded multi-color scenes can still fail.

Scene graphs are a structured middle ground between raw text and boxed layouts: they encode who, what, and how entities interact, and Graph Fusion lets a frozen diffusion backbone use that structure at every denoising step.