Relation-Augmented Diffusion for Layout-to-Image Generation

Presented by: Shuo XU   Juntao DONG   Félix BOS

Introduction

- **Context:** Layout-to-image (L2I) synthesis generates high-fidelity images guided by spatial scaffolding (**bounding boxes**) and text prompts.
- **Objective:** Implement and evaluate **Relation-Augmented Diffusion**, a framework designed to bridge the gap between abstract layout constraints and complex semantic interactions.

Architecture

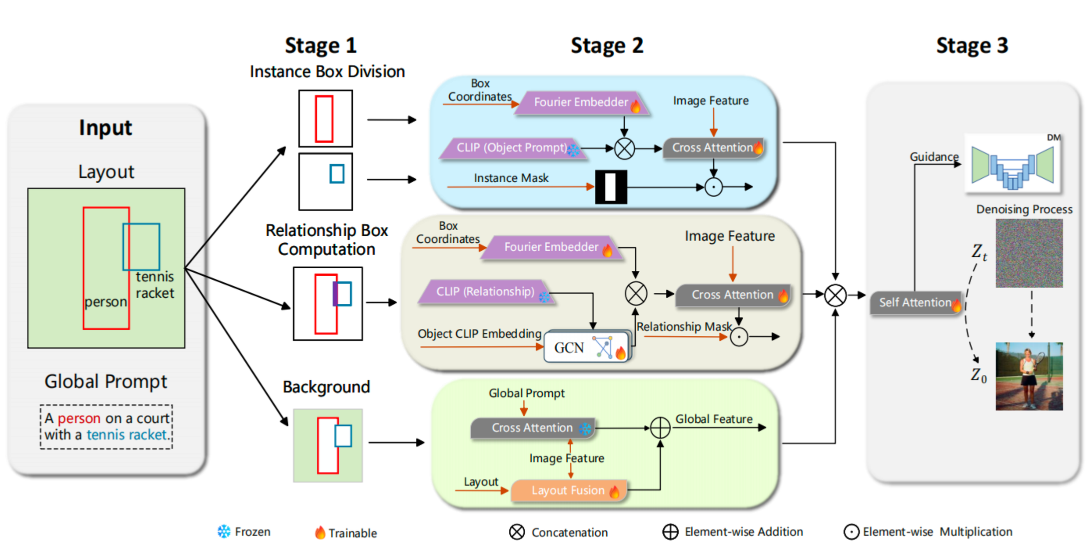

Figure 1:The 3-stage Relation-Augmented Pipeline

Key Methodology

Stage 1: Spatial Zone Partitioning
The layout is decomposed into three semantic masks:**Instance Masks**, **Relation Masks** **Background Masks**.

Stage 2: Relation Feature Injection
Subject–Predicate–Object triples are encoded with a GCN and injected via cross-attention, spatially constrained by relation masks.

Stage 3: Joint Diffusion Denoising
The fused features (Instance + Relation) guide the latent diffusion process, ensuring that the generated pixels in interaction zones adhere to both spatial and relational constraints.

Limitations and Our Motivation

The original Relation-Augmented Diffusion framework achieves strong relation grounding, but has two limitations:

- **Scalability:** Object pairs grow quadratically ($O(N^2)$), increasing relation boxes, GCN cost, and attention complexity.
- **Structured Triplet Dependency:** Relies on predefined subject–predicate–object triples, requiring explicit relation annotations instead of inferring relations directly from prompts.

Key Idea: Instead of explicit graph-based relation modeling, we adopt a lightweight strategy based on spatial conditioning. We retain the concept of the *relation box* (interaction region), but remove structured triplets and graph reasoning. Instead, we model relation effects as spatial modulation of cross-attention.

Algorithm 1: Relation-Aware Attention Scaling

Require: hidden state h, cross-attn output o, mask m, scales α, β
Ensure: updated hidden state h'
1: $\Delta \leftarrow o - h$ ▷ Compute attention residual
2: $s \leftarrow \beta + (\alpha - \beta) \cdot m$ ▷ Calculate spatial scaling map
3: $h' \leftarrow h + \Delta \cdot s$ ▷ Apply relation-aware scaling
4: **return** h'

where $m \in \{0, 1\}$ is a spatial relation mask ($m = 1$ inside the relation box and $m = 0$ outside). Here, α controls the strength of textual influence within the relation region, while β controls the update strength in non-relation areas.

Experiment Setup

Baselines.

- **SD:** Text-only Stable Diffusion.
- **GLIGEN:** Text + Layout to Image (L2I).

Ours.

- **Relation-Aware Attention Modulation:** We extract predicate cues from prompts using a text parser (**Stanza**) and spatially scale cross-attention within the relation box.

Experiment 1: Spatial Attention Scaling in SD

Goal: Test whether *spatially scaling* the text-conditioned cross-attention update improves box adherence in **Stable Diffusion** under a fixed layout.
Protocol: We fix one object box and vary the guidance scale α to control the prompt influence *inside* the box (keeping β fixed for the background), then compare generations.
Hyper-parameters:

- Steps: 30 (DDPM)
- Seed: fixed
- Prompt: "A pig is washing in a bathtub"
- Fixed box: $el\_box = [0.35, 0.3, 0.75, 0.7]$
- Background scale: $\beta = 1.0$
- Sweep: $\alpha \in \{1.2, 1.6, 2.0\}$

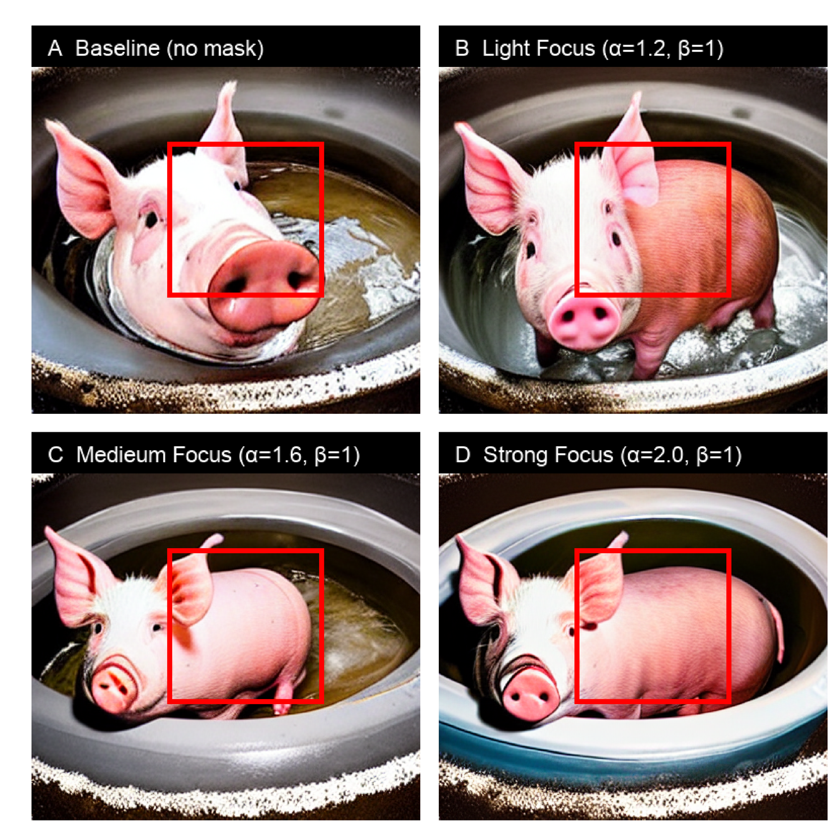

Figure 2:Preliminary Study Result

Experiment 2: Predicate-Guided GLIGEN Training

Motivation: After validating spatial attention scaling in SD, we incorporate the same modulation strategy into GLIGEN and fine-tune the model using LoRA. The key modification is increasing the relation-region scale from $\alpha = 1.0$ (default) to $\alpha = 1.4$ during training.

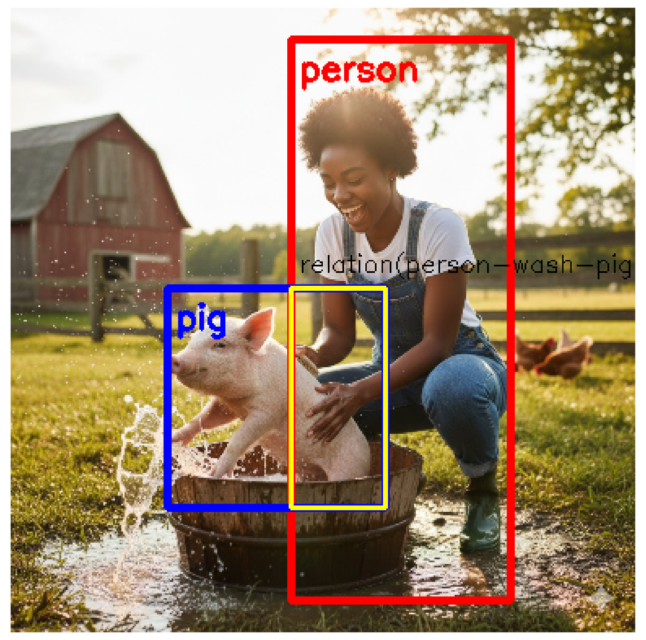

Figure 3:Training data example (image + layout + relation box).

Implementation

Algorithm 2: GLIGEN LoRA Training (with Attention Scaling)

1: **Initialize** GLIGEN pipeline
2: **Freeze** VAE, text encoder, and UNet base weights
3: **Install** LoRA attention processors
4: **Set** attention scales: α=1.4 (inside), β=1.0 (outside)
5: **for** each training step **do**
6: Load image x and relation box B_{rel}
7: Encode image to latent z via VAE
8: Sample timestep t and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
9: $z_t \leftarrow \text{add\_noise}(z, \epsilon, t)$
10: Build GLIGEN conditioning (phrase + B_{rel}) and mask m
11: Predict noise with scaled cross-attn: $\hat{\epsilon} \leftarrow \text{UNet}_{\text{LoRA}}(z_t, t, \text{cond}; \alpha, \beta, m)$
12: Compute loss: $\mathcal{L} \leftarrow \|\hat{\epsilon} - \epsilon\|^2$
13: Backpropagate and update LoRA parameters
14: **end for**

Training Results

Figure 4:Training Results: SD v.s. GLIGEN v.s. GLIGEN Trained

Observation: Compared to SD, GLIGEN follows the layout constraints; after fine-tuning with attention scaling (α=1.4), the interaction is more consistently placed within the target region.