Image generation with graph scene conditioning

Diffusion-Based Scene Graph to Image Generation with Masked Contrastive Pre-Training

Tymoteusz Kwieciński

Outline

1. Paper overview and main contributions
2. Introduction - Latent Diffusion Models
3. Scene Graphs
4. SGDiff method description
5. Experiments
6. Overview of different methods and approaches for spatial image generation

Diffusion-Based Scene Graph to Image Generation with Masked Contrastive Pre-Training

Ling Yang[1*] Zhilin Huang[2*] Yang Song[3] Shenda Hong[1] Guohao Li[4] Wentao Zhang[5]
Bin Cui[1] Bernard Ghanem[4] Ming-Hsuan Yang[6,7]
[1]Peking University [2]Tsinghua University [3]OpenAI [4]KAUST
[5]Mila [6]University of California, Merced [7]Google Research

Ling Yang
Source: Google Scholar

Zhilin Huang
Source: Google Scholar

Main contributions

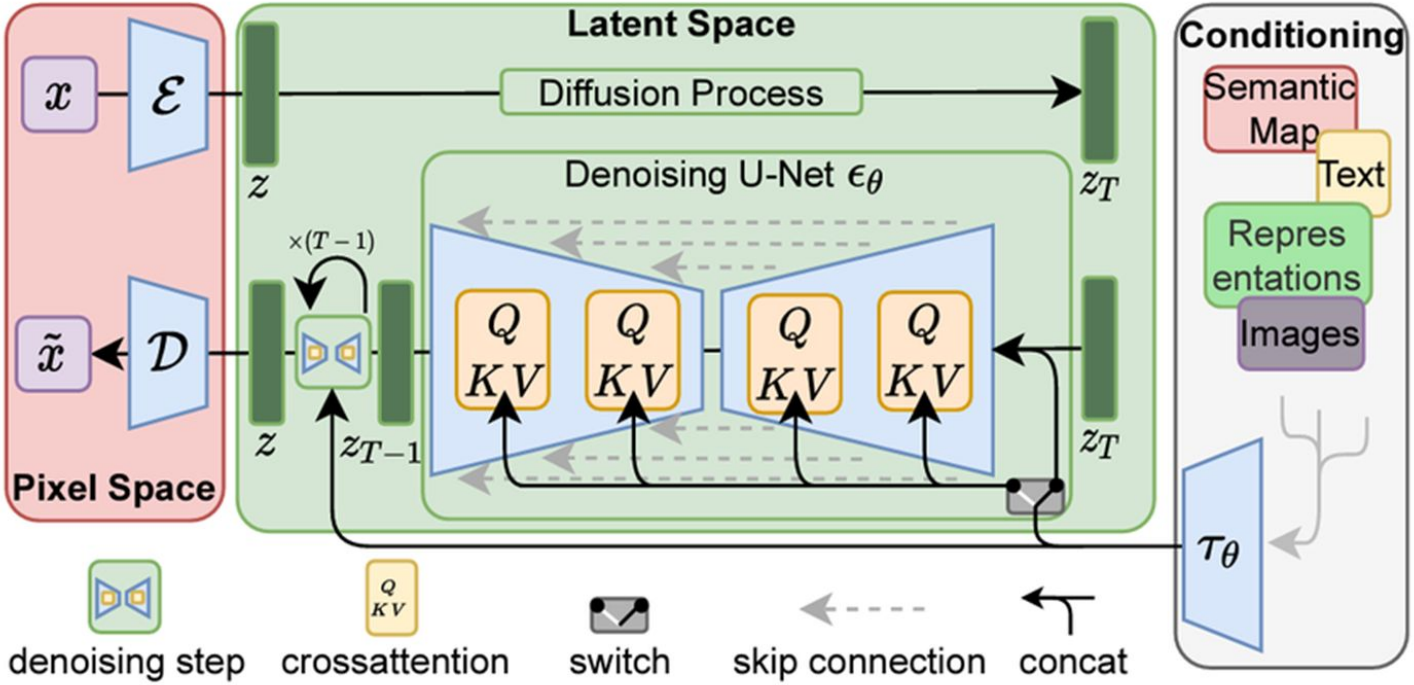
Paper proposes **SGDiff** method for image generation conditioned with scene graphs

This framework is a **first diffusion model** adapted to such task

Authors proposed unique approach utilizing contrastive and masked autoencoding loss

Thanks to conditioning the model directly on the scene graph embeddings, the model is the new **state of the art** (as of 2022)

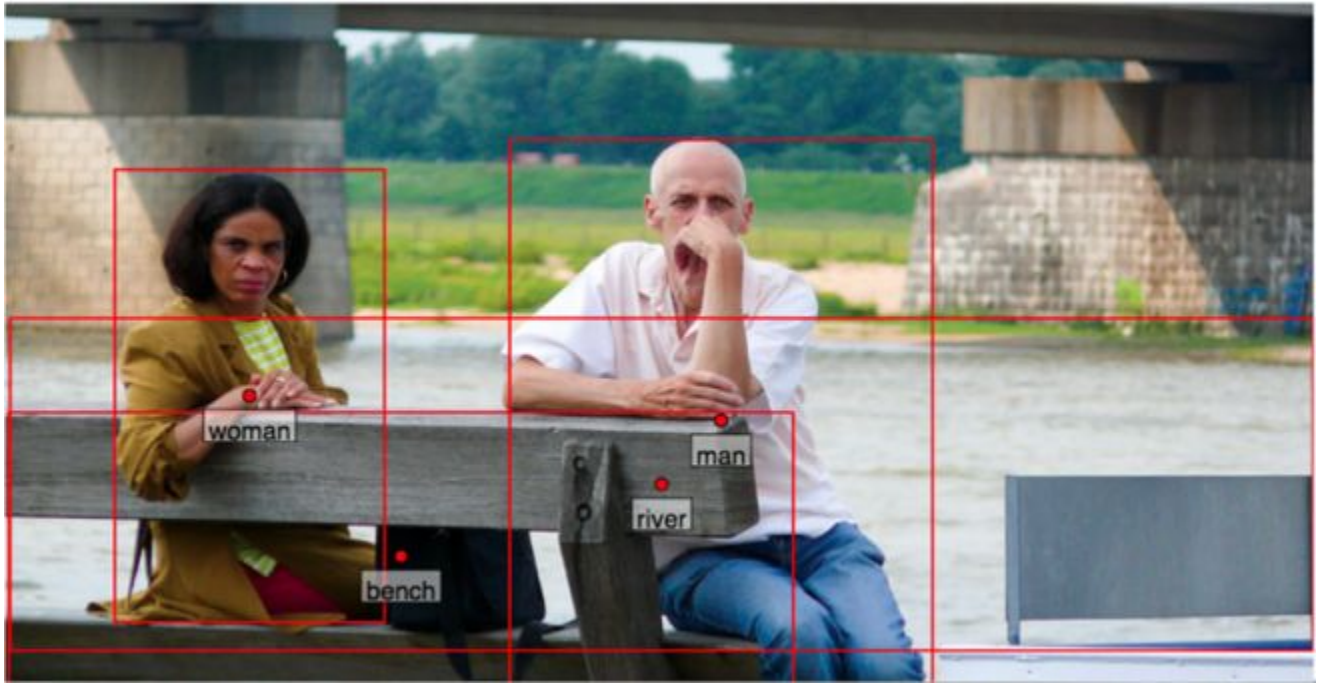
Latent Diffusion Model



The overall pipeline for Latent Diffusion Models (LDM) as proposed by Rombach et al. (*High-Resolution Image Synthesis with Latent Diffusion Models*, 2022)

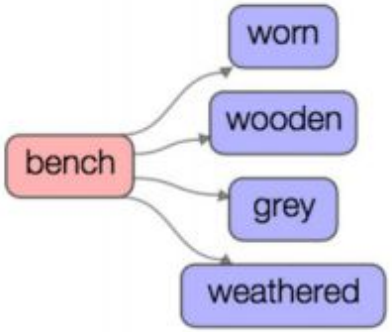
How to represent spatial relationships between objects?

How to represent spatial relationships between objects?



Sample image from the Visual Genome dataset with annotation boxes. Source: Krishna et al., *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*, 2017.

Example of a scene graph



Park bench is made of gray weathered wood

Full scene graph

complex full scene graph of the sample image - it contains plenty of information which would be difficult to encode in a different way, e.g. text or simple bounding boxes, Source:Krishna et al., *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*, 2017.

Scene graphs

Visual Genome (2017) dataset contains **108K images,** 35 objects, 26 attributes, and 21 pairwise relationships between objects**.**

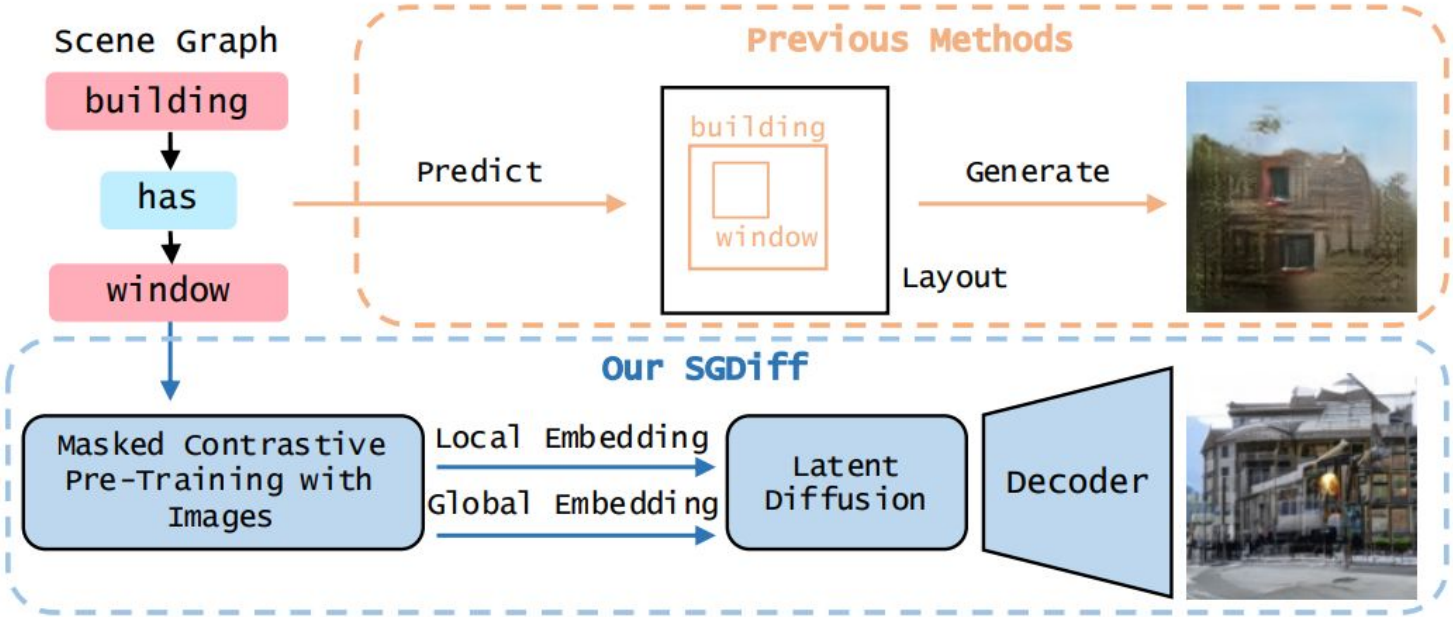
Each image has corresponding **dense scene graph**, which describes the image

Aim of this dataset was to enable training of models that understands complex relationships between objects and are capable of answering difficult questions

SGDiff

first diffusion model conditioned directly on scene graphs

High level method description



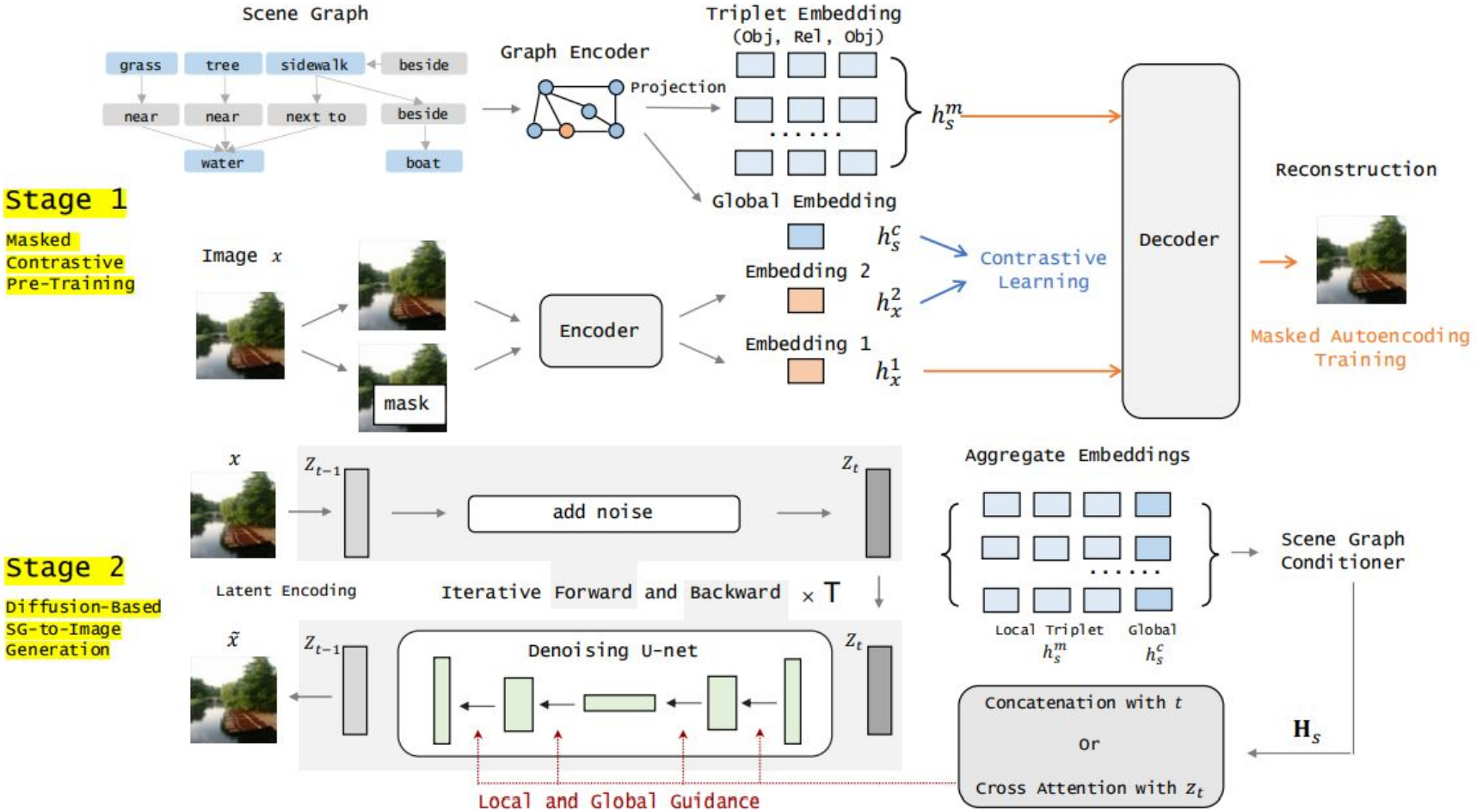
high-level overview of the SGDiff model, compared to the previous methods

Detailed overview of the method

To properly utilize the information from the scene graphs, SGDiff utilizes a separate SG encoder

The encoder produces both **local** and **global** embeddings with **masked autoencoding** and **contrastive loss** respectively

The embeddings are merged relation-wise and after concatenation passed to latent diffusion model


Scene Graph
grass
tree
sidewalk
beside
near
near
next to
beside
water
boat
Graph Encoder
Projection
Triplet Embedding
(Obj, Rel, Obj)
h_s^m
Global Embedding
h_s^c
Stage 1
Masked Contrastive Pre-Training
Image x
mask
Encoder
Embedding 2
h_x^2
Embedding 1
h_x^1
Contrastive Learning
Decoder
Reconstruction
Masked Autoencoding Training
Stage 2
Diffusion-Based SG-to-Image Generation
x
z_{t-1}
add noise
z_t
Latent Encoding
Iterative Forward and Backward $\times T$
$\tilde{x}$
Denoising U-net
Aggregate Embeddings
Local Triplet h_s^m
Global h_s^c
Scene Graph Conditioner
$\mathbf{H}_s$
Concatenation with t
Or
Cross Attention with z_t
Local and Global Guidance

Notation on scene graphs

For a given set of objects $\mathcal{C}_o$ and relations $\mathcal{C}_r$, **scene graph *s*** is represented with a tuple $(O, \mathcal{R})$, where $O = \{o_i \in \mathcal{C}_o\}_{i=1}^{n}$ represents set of objects and $\mathcal{R} = \{r_{ij} \in \mathcal{C}_r\}_{1 \leq i,j \leq n}$ represents relations

(o_i, r_{ij}, o_j) denotes the directed connection from o_i to o_j

$\mathcal{N}_{\text{out}}(o_i)$ (resp. $\mathcal{N}_{in}(o_i)$) the set of children (resp. parents) for node o_i

Objects and relations are embedded as

$$h_{o_i} = \text{Pool}\left(\left\{f_o^{\text{out}}\left(h_{o_i}, h_{r_{ij}}, h_{o_j}\right)\right\}_{j \in \mathcal{N}_{\text{out}}(o_i)} \cup \left\{f_o^{in}\left(h_{o_j}, h_{r_{ji}}, h_{o_i}\right)\right\}_{j \in \mathcal{N}_{\text{in}}(o_i)}\right),$$

$$h_{r_{ij}} = f_r\left(h_{o_j}, h_{r_{ij}}, h_{o_i}\right),$$

where $f_o^{\text{out}}, f_o^{\text{in}}, f_r$ denote separate graph convolutional layers

Pretraining with Masked Autoencoding

For scene graph *s* authors mask out objects o_i and o_j from a random triplet (o_i, r_{ij}, o_j) in the corresponding scene image *x*.

Masked image is denoted as x_m unmasked region as $x_{\setminus m}$. The SG encoder will predict x_m from $x_{\setminus m}$. Specifically the encoder will create embeddings of form:

$$h_s^m = \texttt{concat}(\{(f_o^m(h_{o_i}),\ f_r^m(h_{r_{ij}}),\ f_o^m(h_{o_j}))\})$$

where $f_o^m : \mathbb{R}^{d_o} \to \mathbb{R}^d, f_r^m : \mathbb{R}^{d_r} \to \mathbb{R}^d$ are MLPs with one hidden layer and ReLU

The encoder and auxiliary decoding model d_θ is trained to minimize the loss:

$$\mathcal{L}_{\text{masked}} = \mathop{\mathbb{E}}_{(s,x)\sim D} \|x_m - d_\theta(x_{\setminus m}, h_s^m)\|_2^2,$$

Contrastive loss

This type of embeddings focuses on **global structure** of the image

Graph level embedding h_s^c is obtained by concatenating all the object and relation embeddings from encoder:

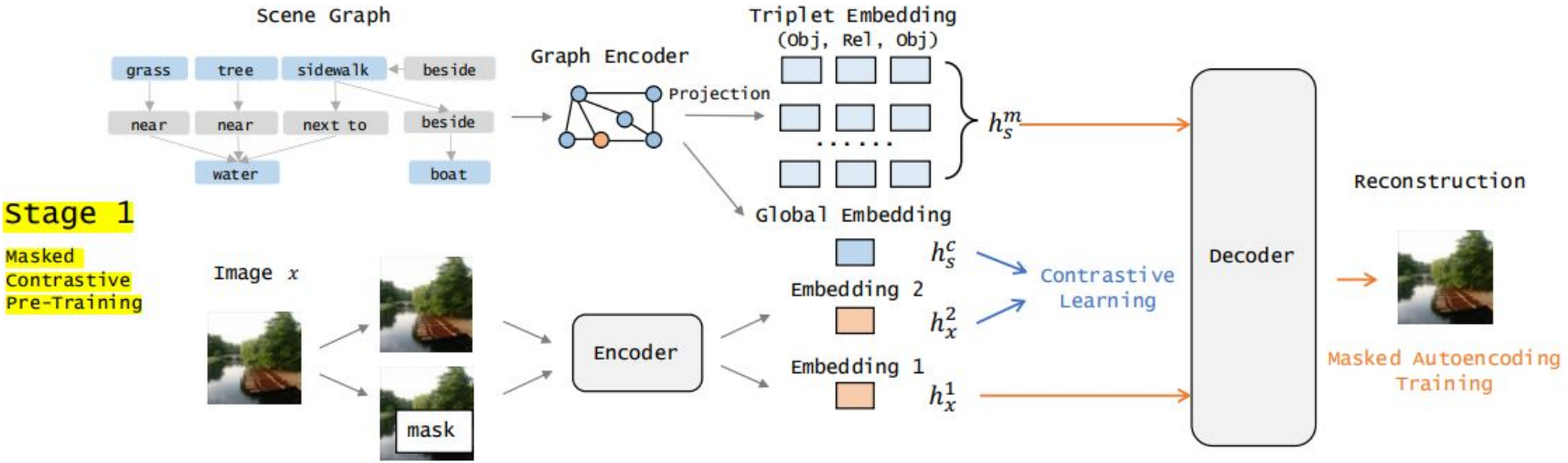
$$h_s^c = f^c(\texttt{concat}(\textbf{Pool}(\{h_o\}_{o\in\mathcal{O}}),\ \textbf{Pool}(\{h_r\}_{r\in\mathcal{R}})))$$

To utilize the contrastive loss, for each scene graph we define (s, x^+) a positive pair if x^+complies with *s*, and (s, x^-) otherwise. The objective equals to:

$$\mathcal{L}_{\text{contrastive}}(f;\tau,k) = \mathop{\mathbb{E}}_{\substack{(s,x^+)\sim D \\ \{x_i^-\}_{i=1}^k \sim D_s}} \left[-\log \frac{\exp\left(h_s^{c\top} h_{x^+}/\tau\right)}{\exp\left(h_s^{c\top} h_{x^+}/\tau\right) + \sum_i \exp\left(h_s^{c\top} h_{x_i^-}/\tau\right)} \right]$$

Stage 1 - encoding the images

Two losses are combined to get the final objective: $\mathcal{L} = \mathcal{L}_{\text{masked}} + \lambda \mathcal{L}_{\text{contrastive}}$



Training the latent diffusion model

The neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(z_t, t\right)$, where z_t is the noisy latent code at the timestep t was trained in a standard way, with conditioning on the SG embeddings

The joint embedding for the whole scene is firstly summed by the relations:

$$h_{r_{ij}}^{sum} = f_o^m(h_{o_i}) + f_r^m(h_{r_{ij}}) + f_o^m(h_{o_j}) + h_s^c.$$

And later concatenated with trainable model ψ_{cond}, called the SG conditioner

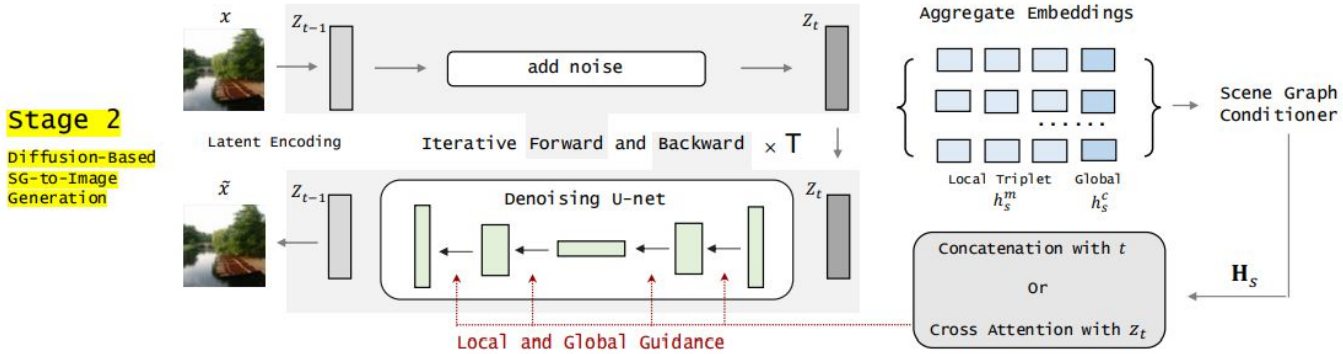
$$\mathbf{H}_s = \psi_{\text{cond}}\left(\text{concat}\left(\left\{h_{r_{ij}}^{\text{sum}}\right\}_{r_{ij}\in\mathcal{R}}\right)\right)$$

Training the latent diffusion model

Authors utilize two methods of conditioning:

1. using timestep concatenation: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(z_t, t; \mathbf{H}_s\right) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(z_t, \text{concat}\left(t, \mathbf{H}_s\right)\right)$

2. combing the matrix H with latent code: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(z_t, t; \mathbf{H}_s\right) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\text{ Cross-Attention }\left(z_t, \mathbf{H}_s\right), t\right)$



Experiments

Experiments setup

To evaluate the proposed method, authors conducted series of experiments on the **COCO-Stuff** and **Visual Genome** datasets

- Quantitative Comparisons
- Qualitative Evaluations
- Ablation Studies

Quantitative Comparisons

Experiments were conducted with different model architectures, to measure the generation quality IS and FID

Authors examined the influence of time conditioning *t* and noisy latent code z_t

SGDiff with noisy latent code conditioning is the new SOTA

Method	Inception Score ↑ COCO	Inception Score ↑ VG	FID ↓ COCO	FID ↓ VG
Real Img (64×64)	16.3 ± 0.4	13.9 ± 0.5	-	-
Sg2Im [22]	6.7 ± 0.1	5.5 ± 0.1	82.8	71.3
WSGC [17]	5.6 ± 0.1	8.0 ± 1.1	91.3	45.3
SOAP [1]	7.9 ± 0.2	-	65.3	-
PasteGAN [30]	9.1 ± 0.2	6.9 ± 0.2	50.9	58.5
SGDiff (with t)	**10.6 ± 0.4**	**8.9 ± 0.5**	**26.8**	**27.5**
SGDiff (with z_t)	**11.4 ± 0.4**	**9.3 ± 0.2**	**22.4**	**16.6**
Real Img (128×128)	24.2 ± 0.9	17.4 ± 1.1	-	-
Sg2Im [22]	7.1 ± 0.2	6.1 ± 0.1	93.3	82.7
WSGC [17]	5.1 ± 0.3	7.2 ± 0.3	108.6	80.4
SOAP [1]	10.4 ± 0.4	-	75.4	-
PasteGAN [30]	11.1 ± 0.7	7.6 ± 0.7	70.7	61.2
SGDiff (with t)	**13.1 ± 0.4**	**9.5 ± 0.5**	**32.7**	**29.6**
SGDiff (with z_t)	**14.6 ± 0.9**	**11.4 ± 0.5**	**30.2**	**20.1**
Real Img (256×256)	30.7 ± 1.2	27.3 ± 1.6	-	-
Sg2Im [22]	8.2 ± 0.2	7.9 ± 0.1	99.1	90.5
WSGC [17]	6.5 ± 0.3	9.8 ± 0.4	121.7	84.1
SOAP [1]	14.5 ± 0.7	-	81.0	-
PasteGAN [30]	12.3 ± 1.0	8.1 ± 0.9	79.1	66.5
SGDiff (with t)	**16.0 ± 0.9**	**13.6 ± 0.7**	**40.1**	**36.4**
SGDiff (with z_t)	**17.8 ± 0.8**	**16.4 ± 0.3**	**36.2**	**26.0**

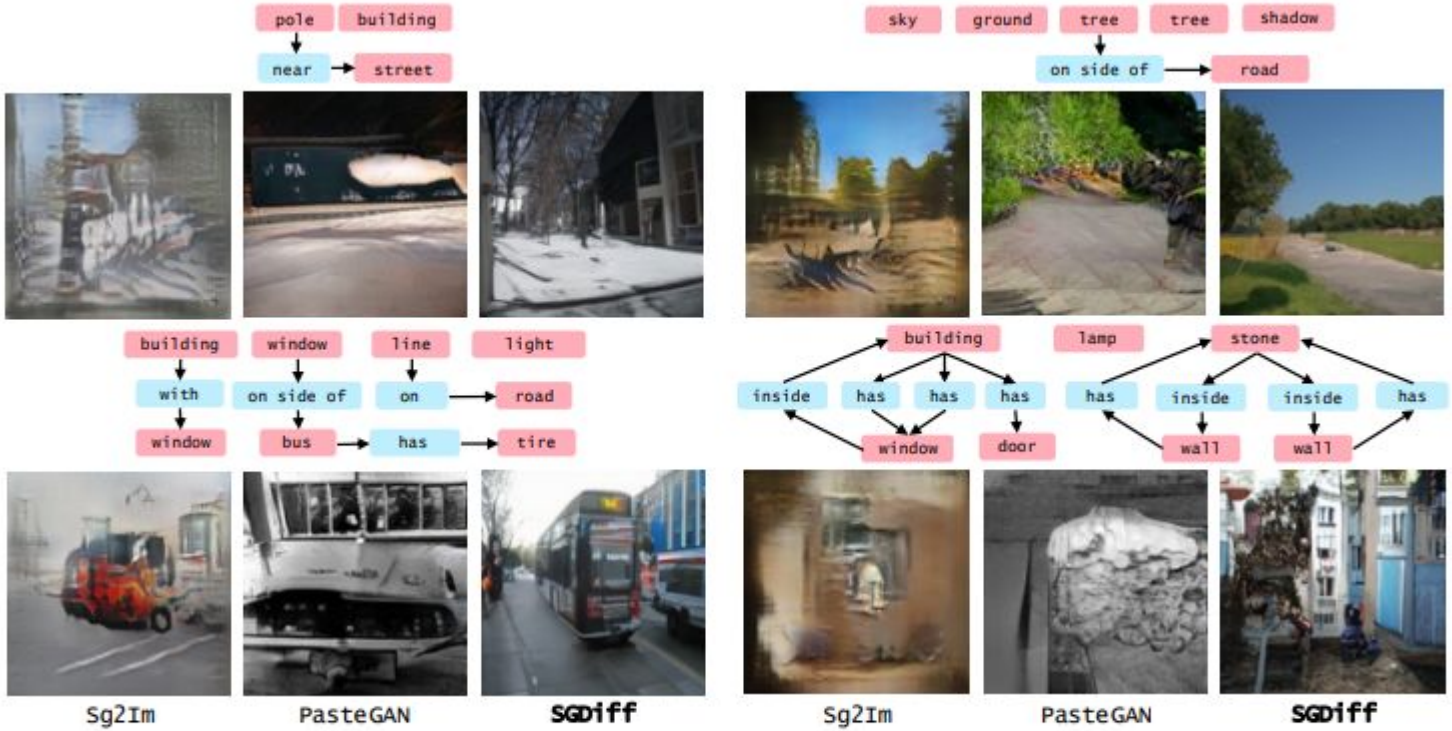
results of methods' evaluation on different sample size image generation task

Qualitative Evaluations

To compare the generated image quality, **SGDiff** was compared with **Im2Seg** and **PasteGAN**

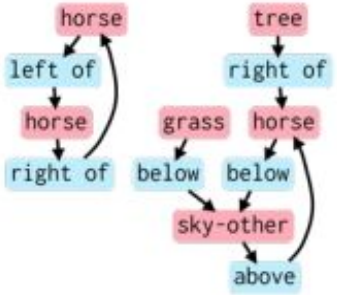
Authors examined visually several examples comparing characteristics of the generated images

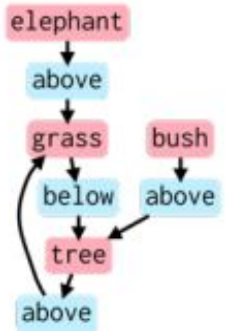
Image manipulation task was also tested to verify if the generated image have consistent scenes



Qualitative comparison of 3 different generative methods on 128x128 images - the complexity of scene graphs gradually rises

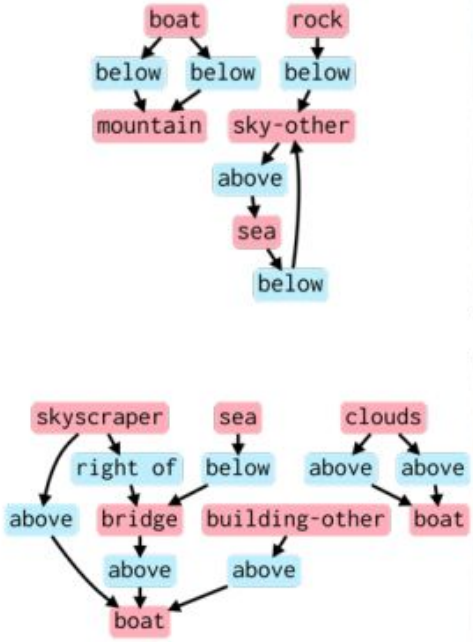
Qualitative Evaluations

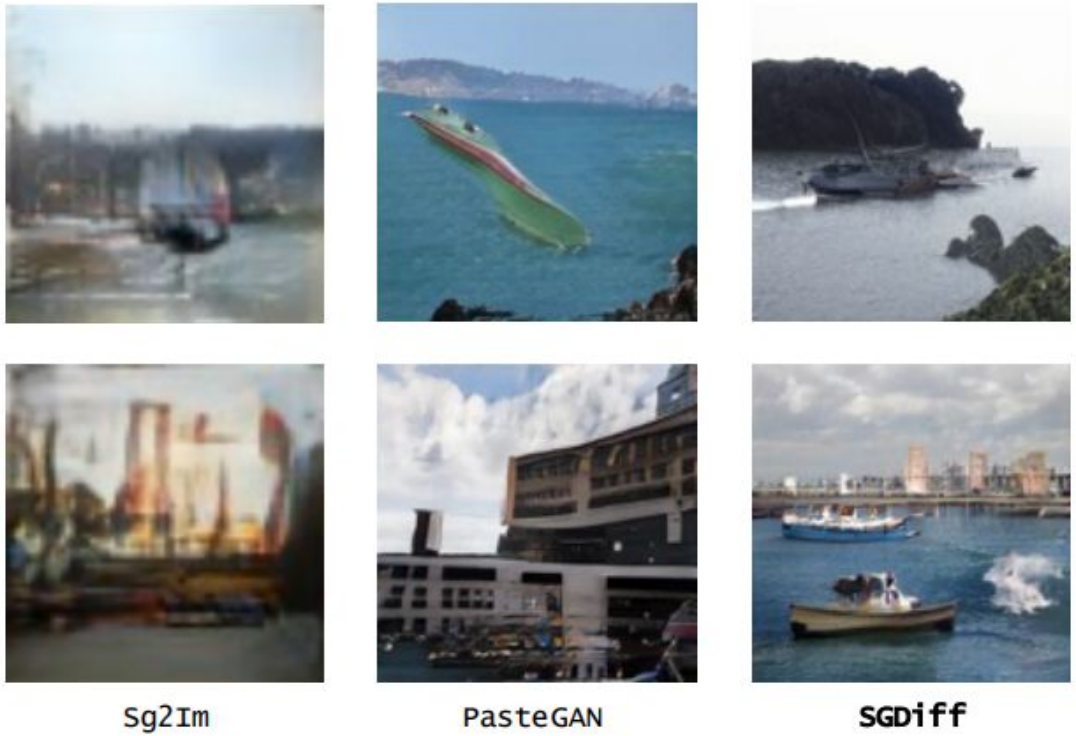




Comparison of generated image on 256x256 images

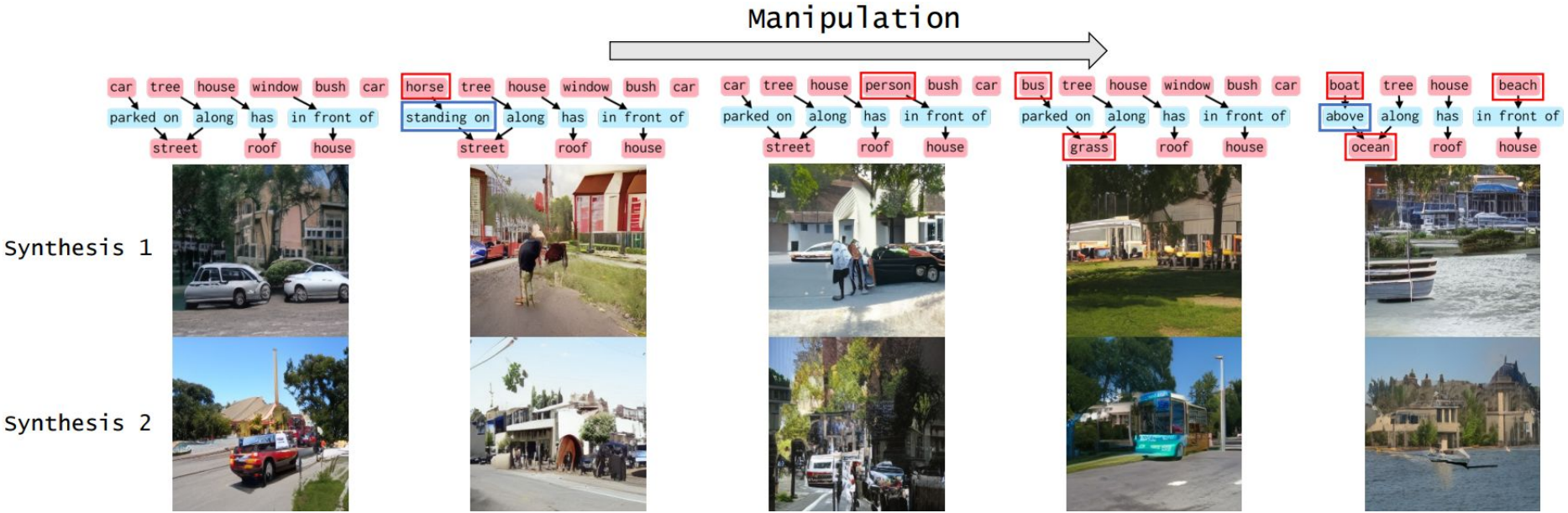
Qualitative Evaluations





Comparison of generated image on 256x256 images

Semantic Image Manipulation



Comparison of 256x256 images generated by SGDiff conditioned on modifications of a scene graph. Red and blue boxes denote object and relation modifications respectively.

Ablation studies

Ablation studies focused on understanding the importance of loss function choice in encoder training.

Authors also verified if scene graph embeddings improve the generation quality

Retrieval Tasks

To verify the importance of **autoencoding loss** and **contrastive loss** in the encoder training, authors measured the similarity between encoder outputs and ground truth values

Cross modal semantic alignment: the most similar element of the generated modality was compared to the ground truth value

Average Accuracy	Graph-to-Image	Image-to-Graph
Obj.	68.6	69.8
Obj. + Rel.	70.3 (**+1.7**)	70.6 (**+0.8**)
Contrastive	70.3	70.6
Contrastive + Masked	73.4 (**+3.1**)	74.1 (**+3.5**)

Ablation study comparing performance of encoders with different settings “Obj.” - only object embeddings trained “Obj. + Rel.” - both object and relation embeddings utilized

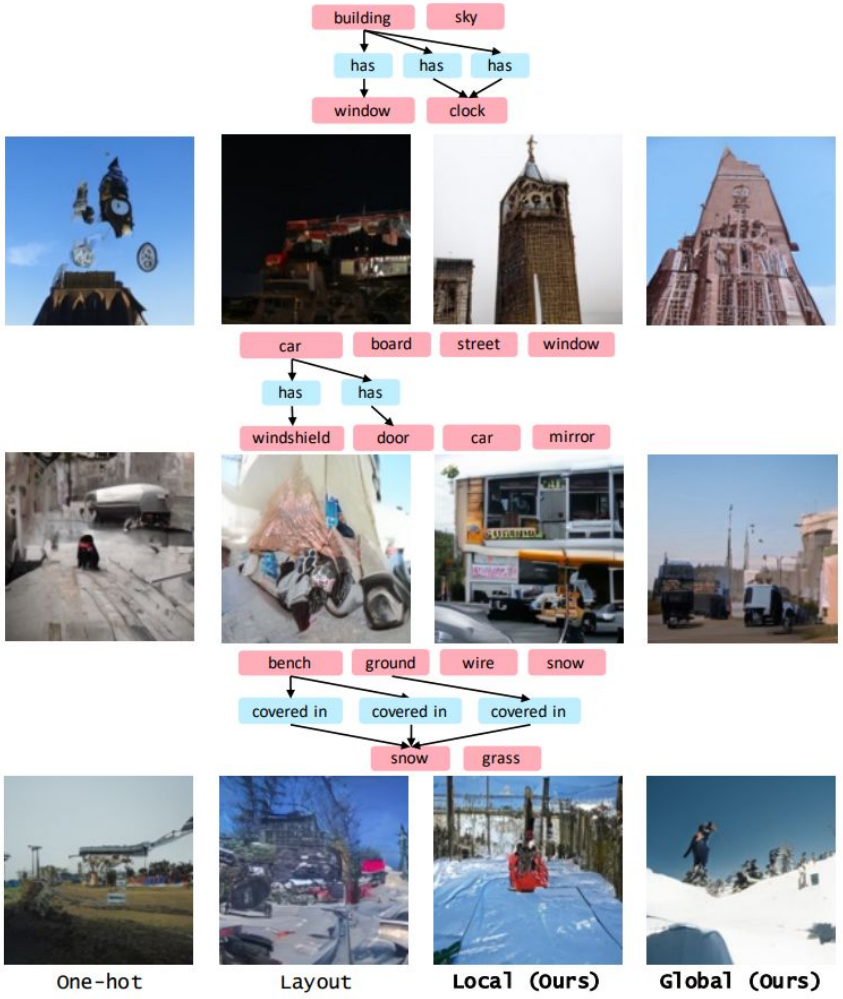
Generation from Scene Graphs

The same diffusion model was trained with different types of embeddings

Embeddings	Inception Score ↑	FID ↓
One-Hot	10.1 ± 1.3	87.1
Layout	12.3 ± 1.0	52.7
Masked (Ours)	15.8 ± 0.6	26.2
Contrastive (Ours)	16.1 ± 0.6	26.9
Masked + Contrastive (Ours)	**16.4 ± 0.6**	**26.0**

Quantitative comparison of diffusion model performance conditioned on different types of embeddings

Visual comparison of generated images using different embeddings. Local and global classes correspond to masked and contrastive embeddings respectively



Summary

SGDiff is a generative model conditioned directly on scene graphs

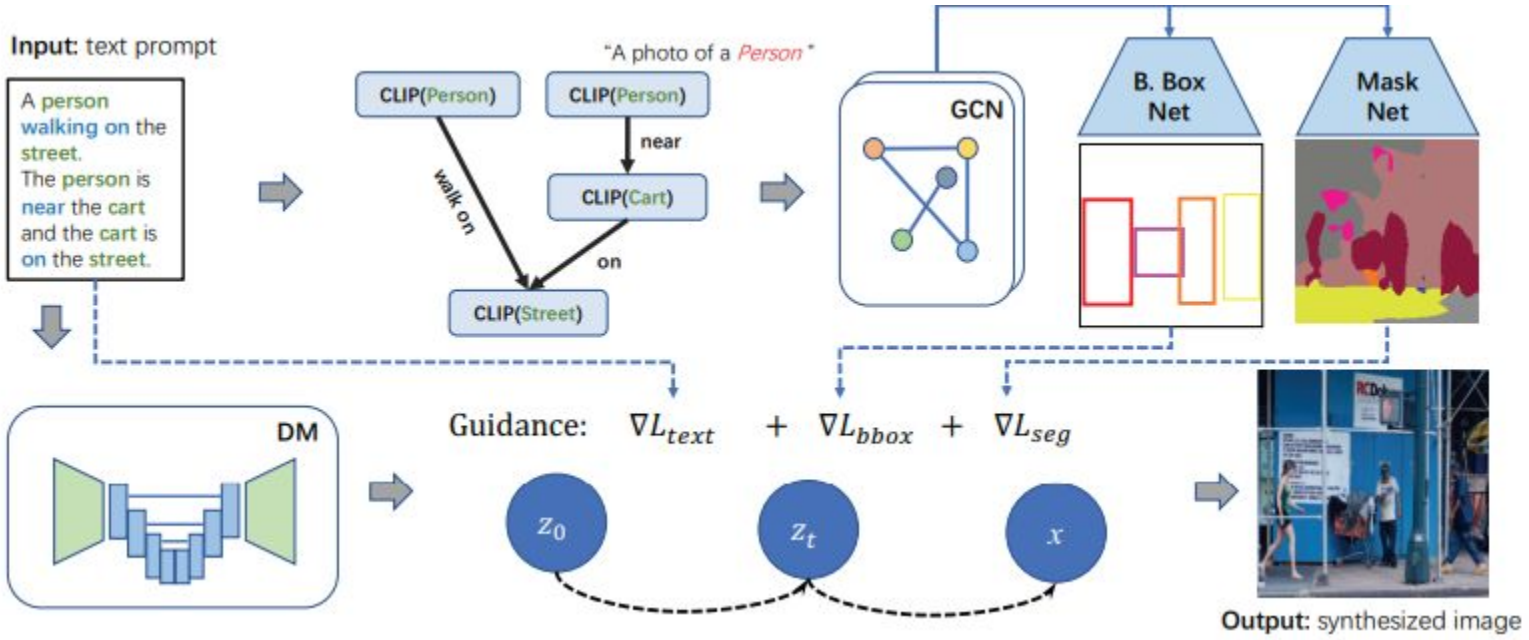
It outperforms existing methods of 2022

The proposed approach is well defined and described, it makes sense mathematically and experimentally

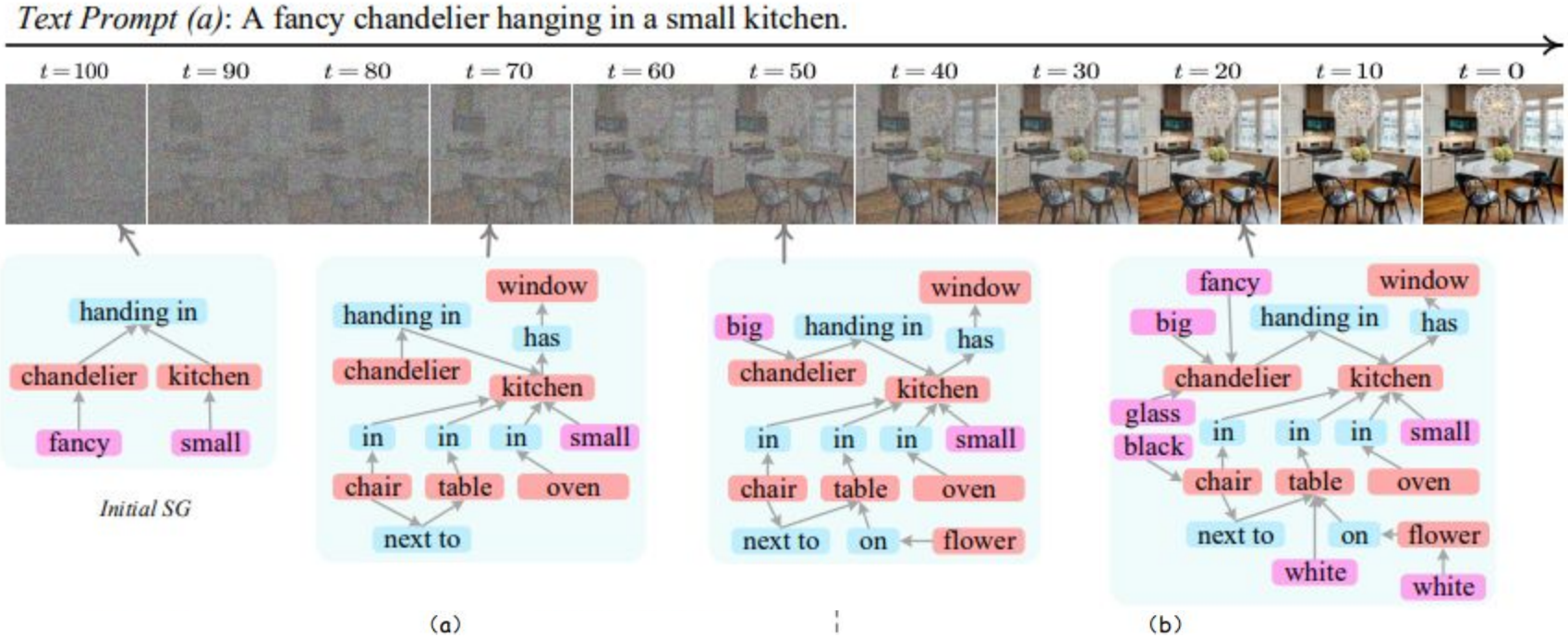
The paper is easy to ready and well-structured

Different methods and approaches for image generation with spatial context

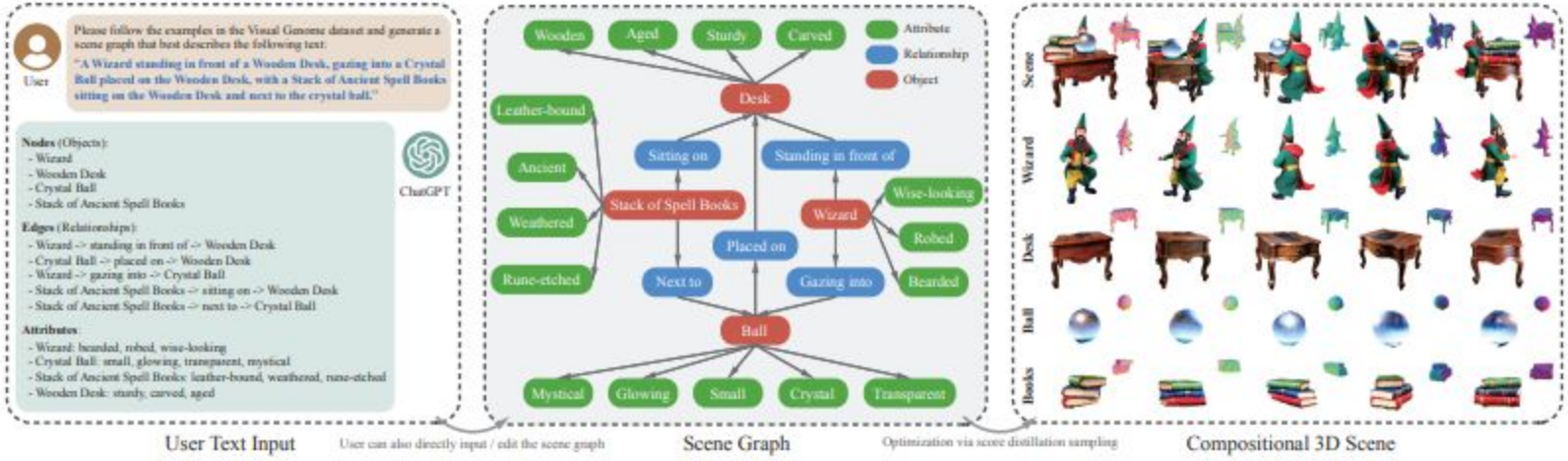
SceneGenie: Scene Graph Guided Diffusion Models for Image Synthesis (ICCV 2023)



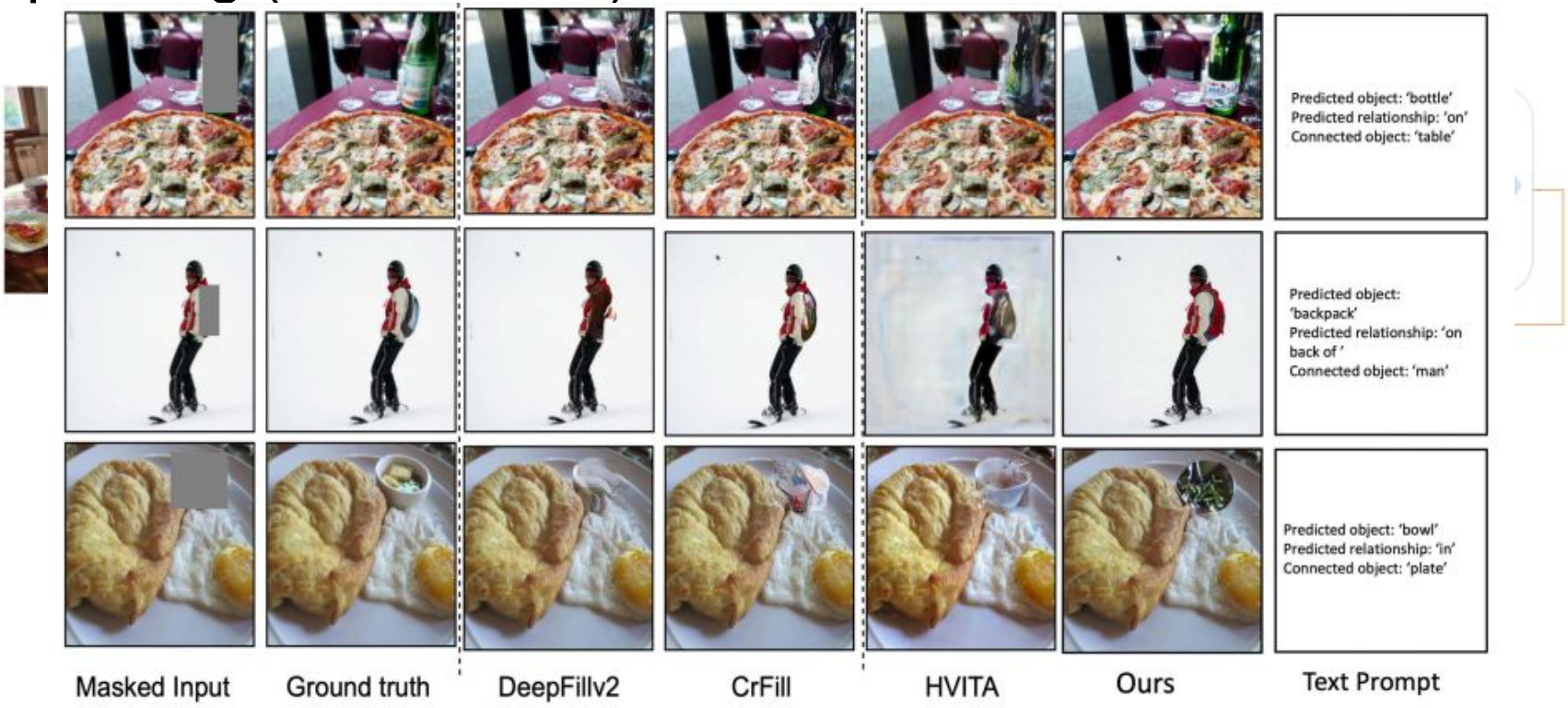
Imagine That! Abstract-to-Intricate Text-to-Image Synthesis with Scene Graph Hallucination Diffusion (NeurIPS 2024)

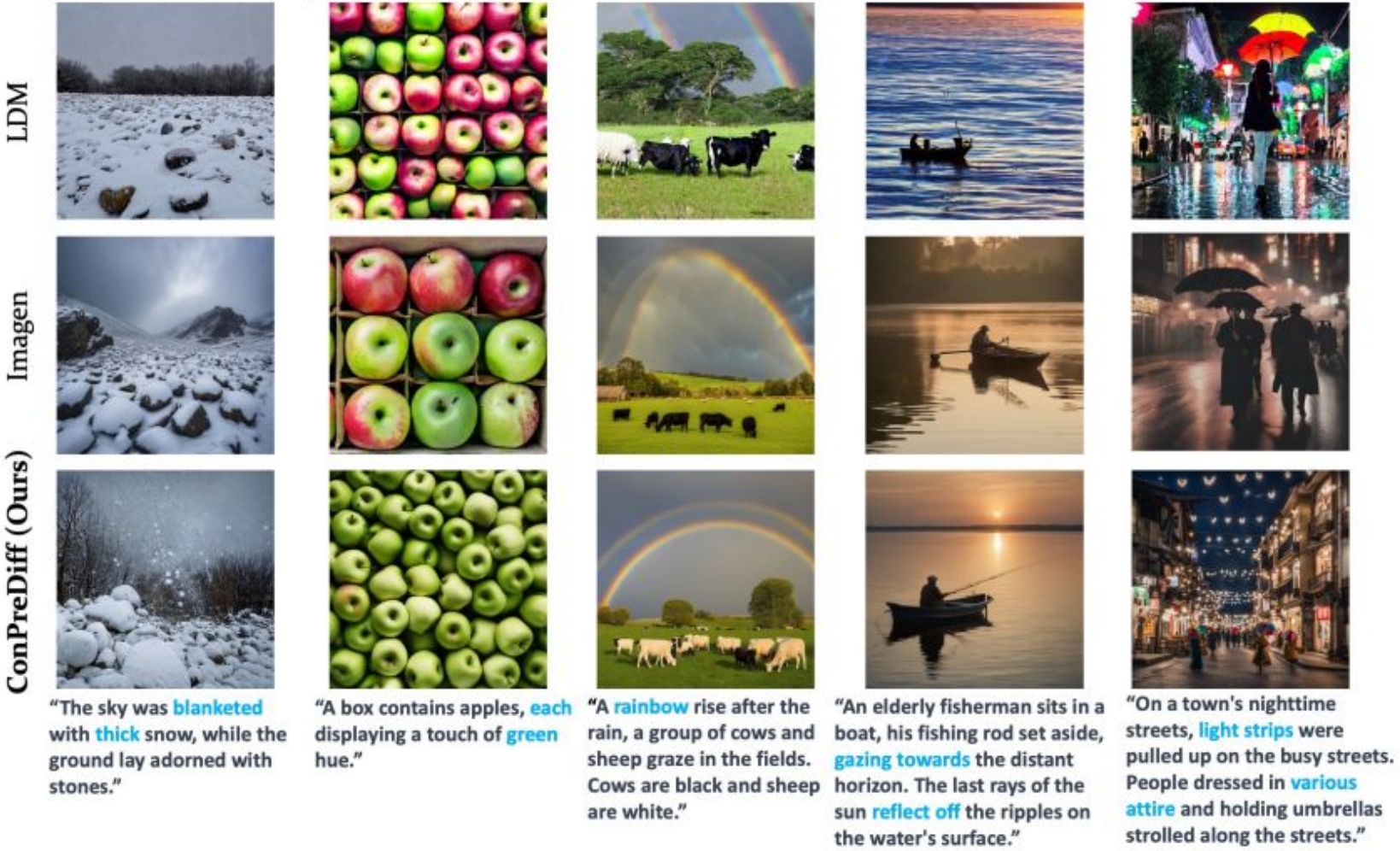


GraphDreamer: Compositional 3D Scene Synthesis from Scene Graphs (CVPR 2024)



Scene Graph Driven Text-Prompt Generation for Image Inpainting (CVPR 2023)




LDM
Imagen
ConPreDiff (Ours)
"The sky was blanketed with thick snow, while the ground lay adorned with stones."
"A box contains apples, each displaying a touch of green hue."
"A rainbow rise after the rain, a group of cows and sheep graze in the fields. Cows are black and sheep are white."
"An elderly fisherman sits in a boat, his fishing rod set aside, gazing towards the distant horizon. The last rays of the sun reflect off the ripples on the water's surface."
"On a town's nighttime streets, light strips were pulled up on the busy streets. People dressed in various attire and holding umbrellas strolled along the streets."

Thank you for your attention!

References

1. L Yang et al. *Diffusion-Based Scene Graph to Image Generation with Masked Contrastive Pre-Training.* CVPR 2022.
2. R Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. CVPR 2022.
3. R Krishna et al. *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*. International Journal of Computer Vision. Springer Science and Business Media LLC.
4. A Farshad et al. *SceneGenie: Scene Graph Guided Diffusion Models for Image Synthesis* ICCV 2023
5. T Shukla et al. *Scene Graph Driven Text-Prompt Generation for Image Inpainting*. CVPR 2023
6. G Gao et al. *GraphDreamer: Compositional 3D Scene Synthesis from Scene Graphs.* CVPR 2024
7. L Yang et al. Improving Diffusion-Based Image Synthesis with Context Prediction. NeurIPS 2023