
FastComposer: Tuning-Free Multi-Subject Image Generation with Localized Attention [\[website\]](#) [\[demo\]](#)[\[replicate api\]](#)



Abstract

Diffusion models excel at text-to-image generation, especially in subject-driven generation for personalized images. However, existing methods are inefficient due to the subject-specific fine-tuning, which is computationally intensive and hampers efficient deployment. Moreover, existing methods struggle with multi-subject generation as they often blend features among subjects. We present FastComposer which enables efficient, personalized, multi-subject text-to-image generation without fine-tuning. FastComposer uses subject embeddings extracted by an image encoder to augment the generic text conditioning in diffusion models, enabling personalized image generation based on subject images and textual instructions with only forward passes. To address the identity blending problem in the multi-subject generation, FastComposer proposes cross-attention localization supervision during training, enforcing the attention of reference subjects localized to the correct regions in the target images. Naively conditioning on subject embeddings results in subject overfitting. FastComposer proposes delayed subject conditioning in the denoising step to maintain both identity and editability in subject-driven image generation. FastComposer generates images of multiple unseen individuals with different styles, actions, and contexts. It achieves 300x-2500x speedup compared to

fine-tuning-based methods and requires zero extra storage for new subjects. FastComposer paves the way for efficient, personalized, and high-quality multi-subject image creation.

Usage

Environment Setup

```
1 conda create -n fastcomposer python
2 conda activate fastcomposer
3 pip install torch torchvision torchaudio
4 pip install transformers==4.25.1 accelerate datasets evaluate diffusers
   ==0.16.1 xformers triton scipy clip gradio facenet-pytorch
5
6 python setup.py install
```

Download the Pre-trained Models

```
1 mkdir -p model/fastcomposer ; cd model/fastcomposer
2 wget https://huggingface.co/mit-han-lab/fastcomposer/resolve/main/
   pytorch_model.bin
```

Gradio Demo

We host a demo here. You can also run the demo locally by

```
1 python demo/run_gradio.py --finetuned_model_path model/fastcomposer/
   pytorch_model.bin --mixed_precision "fp16"
```

Inference

```
1 bash scripts/run_inference.sh
```

Evaluation

```
1 python evaluation/single_object/run.py --finetuned_model_path model/
   fastcomposer/pytorch_model.bin --mixed_precision "fp16" --
   dataset_name data/celeba_test_single/ --seed 42 --
   num_images_per_prompt 4 --object_resolution 224 --output_dir
   OUTPUT_DIR
2
```

```
3 python evaluation/single_object/single_object_evaluation.py --
  prediction_folder OUTPUT_DIR --reference_folder data/
  celeba_test_single/
```

Training

Prepare the FFHQ training data:

```
1 cd data
2 wget https://huggingface.co/datasets/mit-han-lab/ffhq-fastcomposer/
  resolve/main/ffhq_fastcomposer.tgz
3 tar -xvzf ffhq_fastcomposer.tgz
```

Run training:

```
1 bash scripts/run_training.sh
```

TODOs

- ☒ Release inference code
- ☒ Release pre-trained models
- ☒ Release demo
- ☒ Release training code and data
- ☐ Release evaluation code and data

Citation

If you find FastComposer useful or relevant to your research, please kindly cite our paper:

```
1 @article{xiao2023fastcomposer,
2     title={FastComposer: Tuning-Free Multi-Subject Image
3         Generation with Localized Attention},
4     author={Xiao, Guangxuan and Yin, Tianwei and Freeman,
5         William T. and Durand, Fr  do and Han, Song},
6     journal={arXiv},
7     year={2023}
8 }
```