Compositional Conditional Diffusion Model

Shih-Lun Lai, Pin-Chuan Chen, Ching-Wen Ma

College of Artificial Intelligence, National Yang Ming Chiao Tung University
Tainan, Taiwan
larry.ai10@nycu.edu.tw, (+886)908-797618

Abstract

Generative model often struggle to produce data beyond the training distribution. To address this, we explored various methods for unseen data generation, eventually focusing on compositional zero-shot image generation. By guiding the generation model with compositional class labels, we achieved better control over the generation process.Our model can generate unseen images whose compositional labels are not appear in the training set. While large language models like GPT-4 and image generation models like DALL-E offer similar zero-shot generation capabilities, our research emphasizes domain-specific zero-shot generation using smaller models. Through a series of tasks, we demonstrated the effectiveness of compositional zero-shot image generation across various complexities, showcasing its potential in contemporary machine learning.

Key words: Diffusion Model, Image Synthesis, Compositional Zero-shot Learning

Introduction

In traditional machine learning tasks, such as image generation, the testing dataset and training dataset typically share the same distribution. Images generated by the generation model also adhere to the distribution of the training dataset.

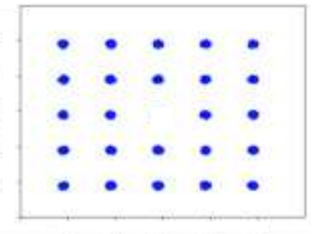
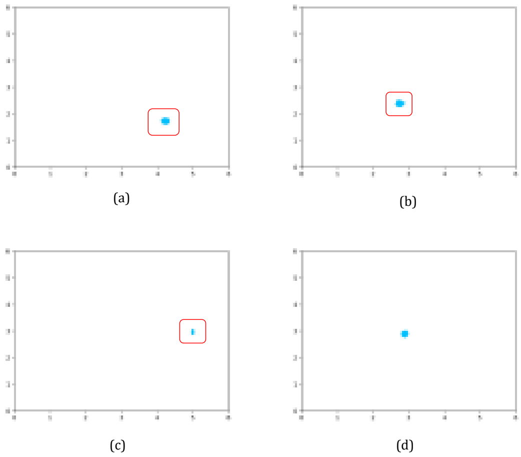


Figure 1 2D point dataset

Figure 1 shows a 2D point dataset with 24 clusters, we aim to generate the complete set of 25 clusters using the existing training data. This includes generating data that does not belong to the distribution of the training dataset, and it cannot be randomly generated; it must be generated based on the conditions we provide. We considered several methods for the scenario described above: directly labeling each cluster, compositional visual generation with energy-based model [1], compositional visual generation with composable diffusion model [2], compositional generation with compositional class labels (ours). We list three different annotation methods for these approaches, each corresponding to one of the four methods mentioned earlier. (a) Direct labeling represents labeling each cluster with a class label, training a class-conditional diffusion model. Clusters will be generated with given class labels. (b) and (c) Clusters in the same row or column share the same label, and training an energy-based model or diffusion model based on each label. Clusters will be generated by using the methods mentioned in [1] [2] to compose samples generated from two different labels (one from a row label and one from a column label). (d) Compositional class labeling indicates that each cluster is labeled with a composite label, which consists of two class labels representing different generation conditions. We train the diffusion model using compositional class labels. Clusters will be generated with given compositional class labels.



(a)

(b)

(c)

(d)

Figure 2. Unseen cluster sampled result on 2D point dataset. (a) Direct labeling (b) Compositional visual generation with energy-based model (c) Compositional visual generation with composable diffusion model (d) Compositional generation with compositional class label (Ours)

In figure 2, we compared the missing clusters generated by four different methods: direct labeling, energy-based model [1], composable diffusion models [2], and compositional class label (Ours). The unseen clusters produced by the first three methods deviated from the desired location (highlighted in red boxes). On the contrary, the unseen clusters generated using compositional class labels as conditions were accurately positioned at the center of the image. With this success on this simple dataset, we have gained more confidence in our method.

Proposed Method

We proposed a diffusion model conditioned on a compositional class label c, where $c = [c_1, \dots, c_n]$ comprises labels from different categories. We break down CCDM into several components: the denoising model, the conditioning module, the timestep encoder and the compositional class encoder, as illustrated in figure 3.

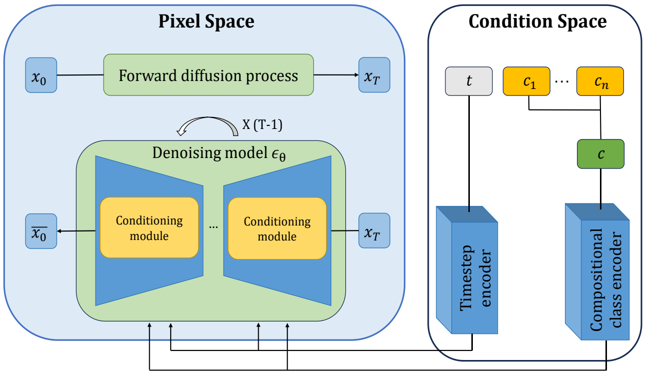


Figure 3. Block diagram of CCDM

In order to guide the diffusion model with compositional class information, we first use one-hot encoding to encode the compositional class label for different categories. Given $c = [c_1, \dots, c_n]$ is the compositional class label, the compositional class encoder is illustrated figure 4.

For the denoising model, we adopted the original U-Net [3] architecture. We modify the original loss function in DDPM [4] to cooperate with compositional class label c. $\beta_t \in (0, 1)$ is the variance schedule controlling the step size and can be regarded as a constant hyperparameter. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, the training objective of our denoising model becomes:

$$L = E_{x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t, c\right)\right\|^2\right]$$

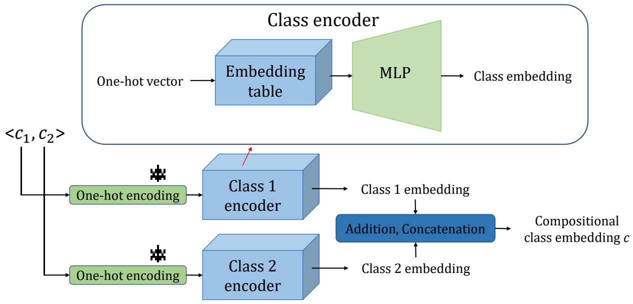


Figure 4. Compositional class encoder

We implemented four different approaches to compare how to better utilize the information of compositional class labels to guide the denoising model. Drawing inspiration from various papers. In the schematic diagram below, we illustrate how we utilize information from compositional class embeddings to guide our denoising model. Before introducing these conditioning modules, let's define some symbols: h represents the hidden state, t_{emb} is the timestep embedding vector, c_{emb} denotes the compositional class embedding vector, and $proj$ is the linear projection layer.

A. Addition (Add)

Follow [4] implementation, we preform element-wise addition on projected timestep embedding $proj(t_{emb})$ and compositional class embeddings c_{emb} into hidden state h. We define the conditioning module as Addition. We call this conditioning module Add for the rest of this thesis.

B. Adaptive Group Normalization (AdaGN)

Follow [5] implementation, we also implement this module on our model, let $c' = proj(c_{emb} + t_{emb})$ and $c' = [c_s, c_b]]$, which we split c' into two vectors with the same dimension. We then inject compute conditioning vector c_s and c_b into hidden state as $GroupNorm(h)(1 + c_s) + c_b$. We call this module AdaGN for the rest of this thesis.
vector

C. Image Class concatenation (IC)

Follow [6] implementation of super-resolution and inpainting diffusion model. We concatenate compositional class embeddings c_{emb} with hidden state h in the diffusion process, then we utilize a bottleneck layer to reduce output dimension. Finally, we add up the output of bottleneck layer and t_{emb}. We believe concatenation maximally preserves the information of compositional class labels. We call this module IC for the rest of this thesis.

D. Cross Attention (CA)

We have also implemented the commonly used conditioning mechanism: cross attention [6]. We use a shallow spatial transformer composed by N transformer blocks, each block consists of a multi-head self-attention layer, a position-wise feedforward network and a multi-head cross-attention layer. We use c_{emb} as key and value in multi-head cross attention layer in the shallow transformer. We call this conditioning module CA in the rest of this thesis.

Although CCDM has the capability of compositional zero-shot image generation, not every generated image aligns perfectly with the unseen compositional class label. Images produced by CCDM with unseen compositional class labels may exhibit deviations. Therefore, we introduced an image selector on unseen composition images composed by multiple binary classifiers. Using compositional class label ``Heel Slipper`` as an example, we would train a ``Heel`` binary classifier and a ``Slipper`` binary classifier. These classifiers are then utilized these classifiers to filter the images generated by CCDM, as shown in figure 5.

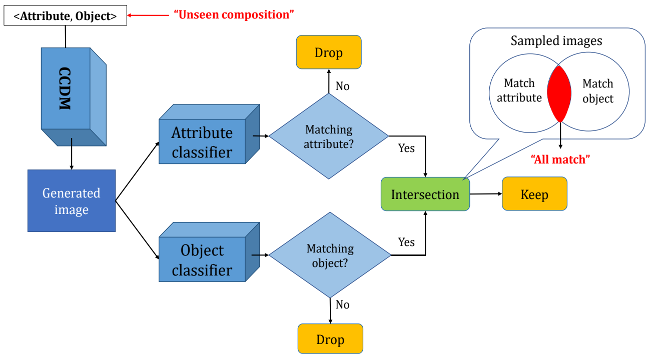


Figure 5. Image selector on unseen composition

Experiment results

We applied CCDM to test the ability of compositional zero-shot image generation on the UT-Zappo50K dataset and CelebFaces Attribute (CelebA) dataset. We divided these two datasets into two condition settings and three condition settings for different application scenarios.

To evaluate the performance of CCDM, the images generated by CCDM are filtered by the image selector to choose those that match the unseen composition labels. Sample accuracy is employed to assess CCDM's ability to generate unseen compositions, which includes whether the images generated by CCDM match the unseen compositional class label.

As shown in figure 6 and figure 7, even without any training data for unseen compositions during training phase, CCDM can accurately generate both training compositions and unseen compositions. CCDM demonstrates the ability to generalize from old concepts present in the training data to new concepts never seen during testing. These results illustrate CCDM's applicability to real-world datasets.

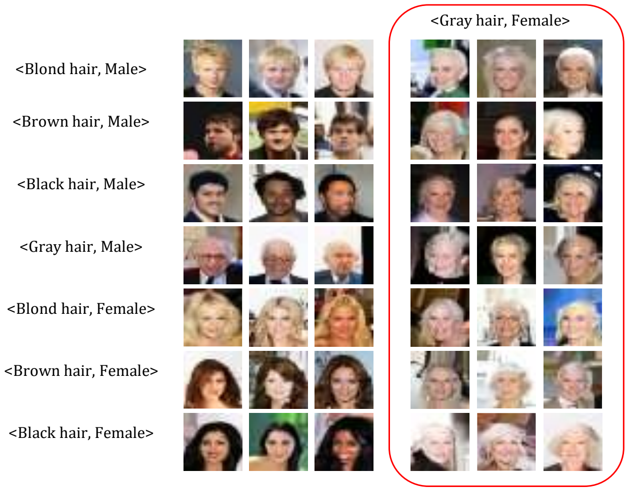


Figure 6. Sampled result of CelebA dataset. The unseen composition images are highlighted with a red box.

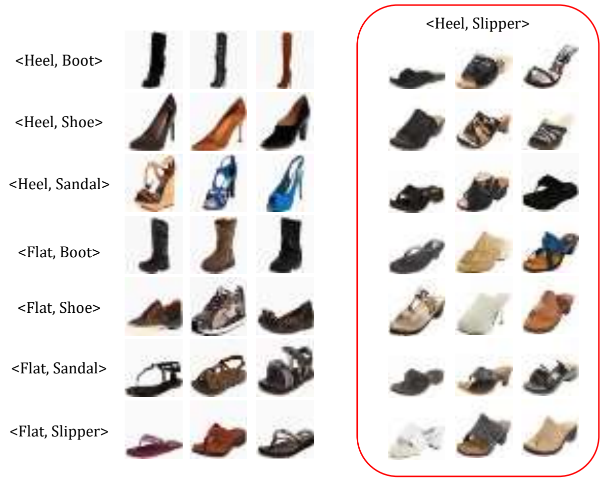


Figure 7. Sampled result of UT-Zappo50K dataset. The unseen composition images are highlighted with a red box.

Table 1 illustrates that among the four conditioning modules we used, IC performs the best for compositional zero-shot image generation. This closely aligns with our expectations. For the diffusion model with compositional class labels as conditions, element-wise addition on condition vectors may aggregate information from multiple class labels, potentially hindering the model's ability to generate correct samples. Regarding CA, it encounters limited dimensions in both keys and values compared to using text as conditions. Concatenation avoids mixing the class embeddings of different conditions using addition, making it less prone to confusion for CCDM.

Table 1 Overall architecture comparison

Architecture	Acc. ↑
Add	0.34
AdaGN	0.34
IC	**0.42**
CA	0.36

Conclusion

We proposed a diffusion model conditioned on compositional class labels for compositional zero-shot image generation. Starting from the most basic dataset, we employed various methods to achieve Unseen data generation. Eventually, we adopted the approach of using compositional class labels as conditions to guide the diffusion model. By succeeding with the simplest dataset, we further validated that our model can achieve compositional zero-shot image generation through different datasets. Applying compositional cognitive abilities to the domain of image generation, our model can achieve "learning from old compositional concepts and generalizing to new compositional concepts."

References

[1] Yilun Du, Shuang Li, and Igor Mordatch, "Compositional visual generation with energy based models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6637–6647, 2020.

[2] Nan Liu, et al. "Compositional visual generation with composable diffusion models," in *European Conference on Computer Vision*. Springer, 2022, pp. 423–439.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer, 2015*, pp. 234– 241

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[5] lexander Quinn Nichol and Prafulla Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[6] Robin Rombach, et al. "High-resolution im-age synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.

[7] Martin Heusel, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

ISBN 979-8-3503-9492-4