Compositional Generative Modeling

Introduction

Generative Models have had tremendous success in recent years, to train them to generalize in complex and changing environments is challenging, which is why compositionality is a desirable property.
What we want, is for two learned generative models(i.e. Diffusion, Flow Matching etc.), represented by their learned densities p_{D_1} and p_{D_2} on some datasets D_1 and D_2 to be able to sample from

$$p_{comp} \propto p_{D_1} \cdot p_{D_2}$$

where the normalization constant Z given by:

$$Z = \int_{\mathbb{R}^d} p_{D_1}(x) \cdot p_{D_2}(x) \, dx.$$

is not tractable in practice (if we assume data in $\mathbb{R}^d$ which is not too restrictive for now). To do this, multiple naive and more involved ideas exist, which we will try to explore in this document, focussing on toy datasets.

Intuitively, in the case seems clear at first glance in the case of diffusion models where we have trained score $\nabla \log p_t^i$ with $p_1 \approx p_{D_1}$ for $i \in \{1, 2, ..., n\}$ (in general, for now we consider $n = 2$) and $t \in [0, 1]$ then, we have

$$\nabla \log(p_{D_1} \cdot p_{D_2}) = \nabla \log p_{D_1} + \nabla \log p_{D_2} \approx \nabla \log p_1^2 + \nabla \log p_1^1$$

which seems to solve the issue. However, as pointed out in [1] problems arise (we actually got aware of this paper after conducting similar experiments to theirs). Namely, when sampling we utilize the score evolving through time, for the true product p_{comp}^t this can be written as

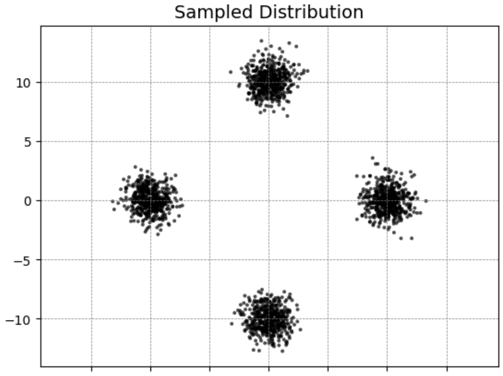
$$\nabla \log p_{comp}^t(x_t) = \nabla \log \left(\int_{\mathbb{R}^d} \cdot p(x_t \mid x) p_{D_1}(x) p_{D_2}(x) dx \right)$$

which in general will not equal to

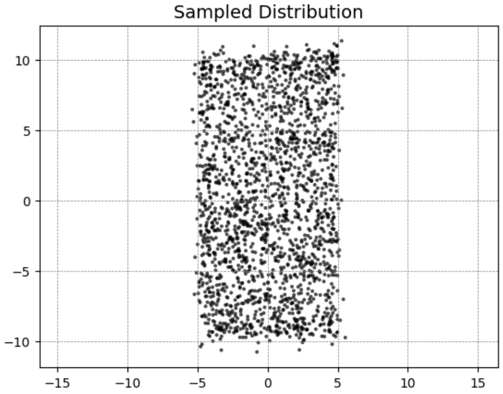
$$\nabla \log \left(\int_{\mathbb{R}^d} \cdot p(x_t \mid x) p_{D_2}(x) dx \right) + \nabla \log \left(\int_{\mathbb{R}^d} \cdot p(x_t \mid x) p_{D_1}(x) dx \right)$$

Thus, the above only really works well for independent D_1 and D_2 which is rarely the case.

Imagine we have the following diffusion models trained on the following distributions.

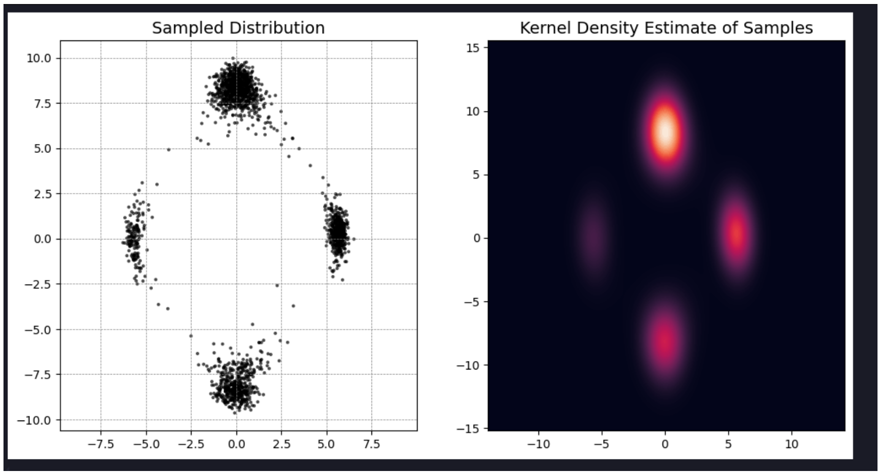


p_{D_1}



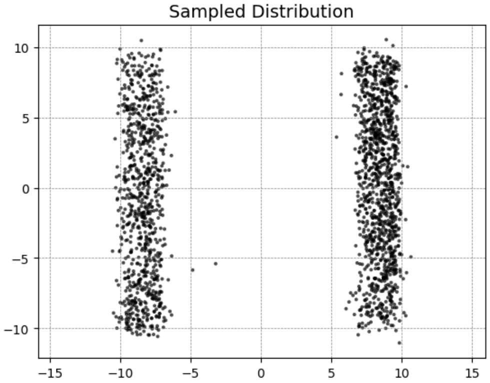
p_{D_2}

Then, just adding their scores gives the following, which is of course not the true product.

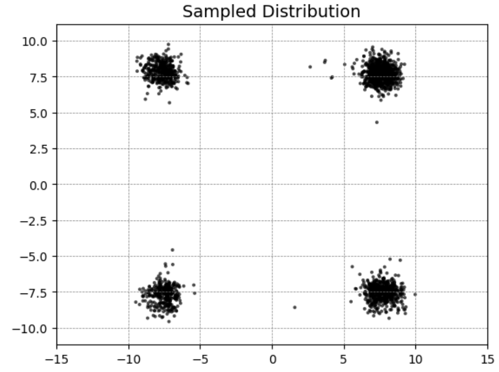


p_{comp} ?

However, in the case where we have simpler distributions with better behaved overlap, it works a bit better (why?). (here D_2 is the same pair of rectangles just rotated 90 degrees)

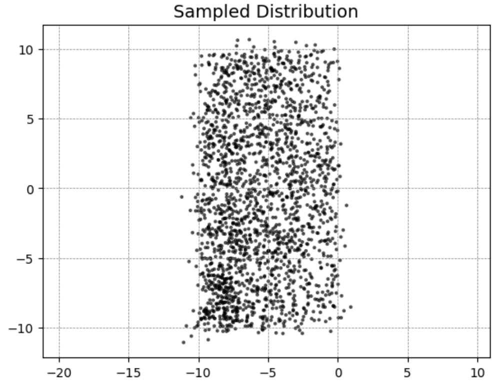


p_{D_1}

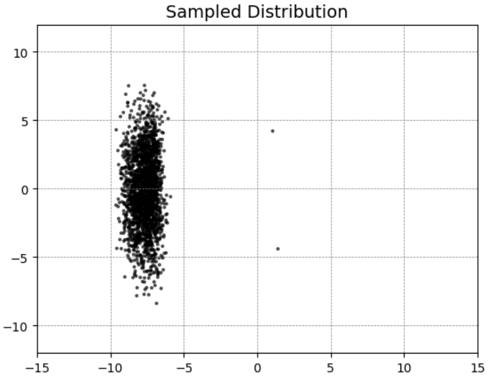


p_{comp}

Even if the supper of the product is not 'symmetric', it still somewhat works, although not the entire support is covered. Here D_2 is the distribution of the two rectangles above.



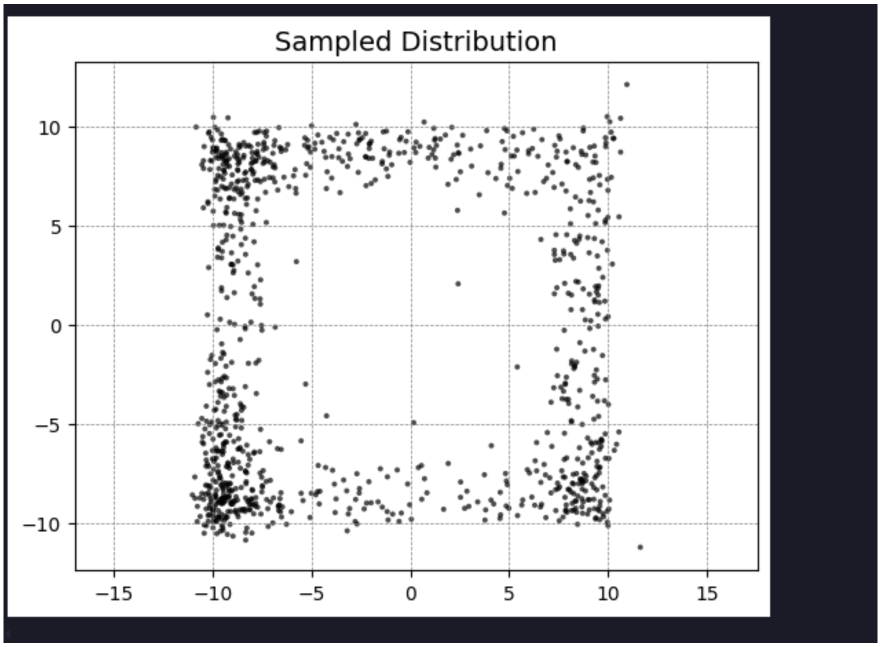
p_{D_1}



p_{comp}

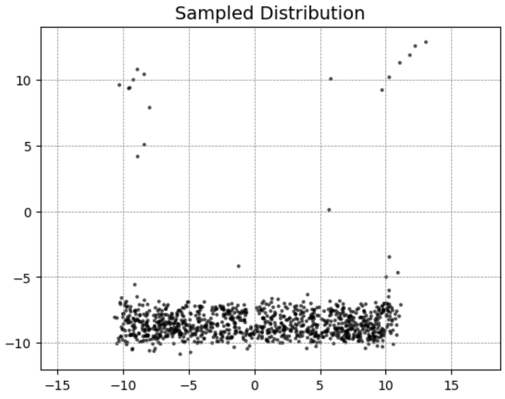
Other approached for this problem have been widely explored, in [2] they, instead of working with independently trained models work with Diffusion guidance and certain pre-specified concepts $(c_1, ..., c_n)$ they which to 'compose' i.e. they want to have both c_1 and c_2 present in an image, or they want c_1 NOT in the image etc. This was also discussed in [1] and [3].

They do this, by utilizing diffusion guidance and utilizing an 'unconditional' Model as the baseline, which in their case is just the union since they are inspired by guidance, in [3] and [1] they discuss why this approach could fail.
After first testing, we, for the simple four overlapping rectangles, see that (up to numerical errors?) the guidance works for getting the product of two concepts.

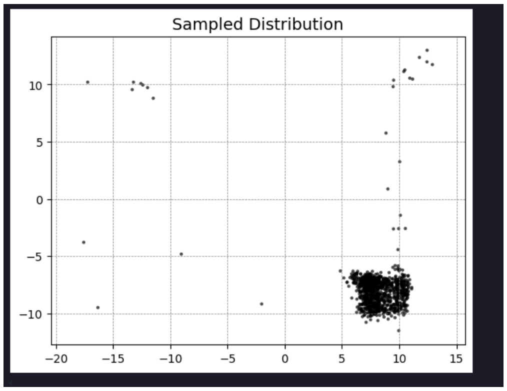


Unconditional

Now, when conditioning on only one rectangle, or two rectangles at the same time, we get.

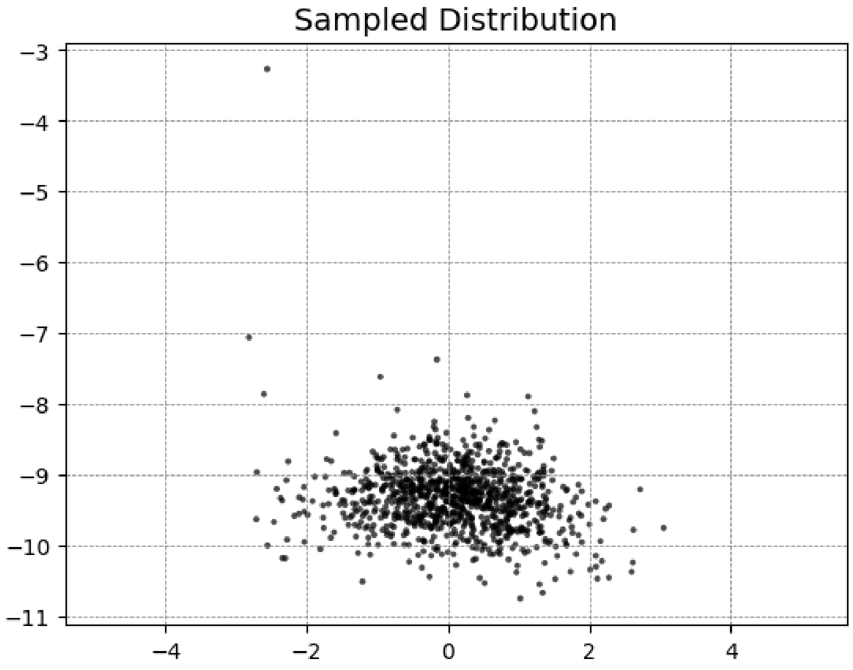


p_{D_1}



p_{comp}

But, even if not working perfectly, we can see that on these toy datasets, the guidance approach somewhat works in the above setting of rectangle and Mixture of Gaussians.



Guidance on one Mode and the Rectangle

metrics: KL, Wasserstein, https://en.wikipedia.org/wiki/Jensen

References

[1] Yilun Du et al. "Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc". In: *International conference on machine learning*. PMLR. 2023, pp. 8489–8510.

[2] Nan Liu et al. *Compositional Visual Generation with Composable Diffusion Models*. 2023. arXiv: `2206.01714 [cs.CV]`. URL: `https://arxiv.org/abs/2206.01714`.

[3] Arwen Bradley et al. *Mechanisms of Projective Composition of Diffusion Models*. 2025. arXiv: `2502.04549 [cs.LG]`. URL: `https://arxiv.org/abs/2502.04549`.