Position: Compositional Generative Modeling
A Single Model is Not All You Need

Yilun Du [1] Leslie Kaelbling [1]

Abstract

Large monolithic generative models trained on massive amounts of data have become an increasingly dominant approach in AI research. We argue that we should instead construct large generative systems by composing smaller generative models together. We show how such a compositional generative approach enables us to learn distributions in a more data-efficient manner, enabling generalization to parts of the data distribution unseen at training time. We further show how this enables us to program and construct new generative models for tasks completely unseen at training. Finally, we show that in many cases, we can discover compositional components from data.

1. Introduction

In the past two years, increasingly large generative models have become a dominant force in AI research, with compelling results in natural language (Brown et al., 2020), computer vision (Rombach et al., 2022) and decision-making (Reed et al., 2022). Much of the AI research field has now focused on scaling and constructing increasingly large generative models (Hoffmann et al., 2022), developing tools to build even larger models (Dao et al., 2022; Kwon et al., 2023), and studying how properties emerge as these models scale in size (Lu et al., 2023; Schaeffer et al., 2023).

Despite significant scaling in generative models, existing models remain far from intelligent, exhibiting poor reasoning ability (Tamkin et al., 2021), extensive hallucinations (Zhang et al., 2023b), and poor understanding of commonsense relationships in images (Figure 2) (Majumdar et al., 2023). Despite this, large models have already been trained on most of the existing data on the Internet and have reached the limits of modern computational hardware, costing hundreds of millions of dollars to train (Figure 1).

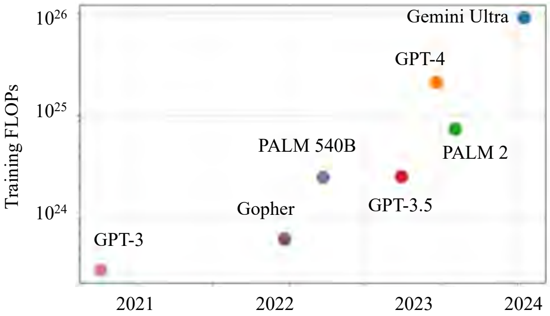


Figure 1. **Rising Size and Cost of Models.** While much of AI research has focused on constructing increasingly larger monolithic models, training costs are exponentially rising by a factor of 3 every year with current models already costing several hundred million dollars per training run. Data from (Epoch, 2023).

Inference costs of such gigantic models are also prohibitive, requiring large computational clusters and a cost of several dollars for longer queries and answers (OpenAI).

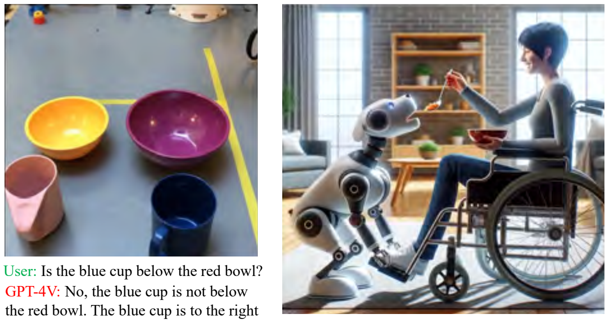
In addition, adapting such large models to new task distributions is difficult. Directly fine-tuning larger models is often prohibitively expensive, requiring a large computation cluster and an often difficult-to-acquire fine-tuning dataset. Other works have explored leveraging language and a set of in-context examples to teach models new distributions, but such adaptation is limited to settings that are well expressed using a set of language instructions that are further roughly similar to the distributions already seen during training (Yadlowsky et al., 2023).

In this paper, we argue that as an alternative to studying how to scale and construct increasingly large monolithic generative models, **we should instead construct complex generative models compositionally from simpler models**. Each constituent model captures the probability distribution of a subset of variables of the distribution of interest, which are combined to model the more complex full distribution. Individual distributions are therefore much simpler and computationally modeled with both fewer parameters and learnable from less data. Furthermore, the combined model can generalize to unseen portions of the data distribution as long as each constituent dimension is locally in distribution.

Such compositional generative modeling enables us to effectively represent the sparsity and symmetry naturally found

[1] MIT. Correspondence to: Yilun Du <yilundu@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



User: Is the blue cup below the red bowl?
GPT-4V: No, the blue cup is not below the red bowl. The blue cup is to the right side of the picture, and the red bowl is not visible in the image

DALL-E 3 Prompt: Create a picture of a Spot robot feeding a person.

Figure 2. **Limited Compositionality in Multimodal Models.** Existing large multimodal models such as GPT-4V and DALL-E 3 still struggle with simple textual queries, often falling back to biases in data.

in nature. Sparsity of interactions, for instance between an agent and external environment dynamics can be encoded by representing each with separate generative models. Sources of symmetry can be captured using multiple instances of the same independent generative component to represent each occurrence of the symmetry, for instance by tiling patch-level generative model over the patches in an image. Compositional structure is widely used in existing work, to tractably represent high dimensional distributions in Probablistic Graphical Models (PGMs) (Koller & Friedman, 2009), and even in existing generative models, *i.e.* autoregressive models which factorize distributions into a set of conditional probability distributions (represented by a single model).

Compositional generative modeling further enables us to effectively program and construct new generative systems for unseen task distributions. Individual generative models can be composed in new ways, with each model specifying a set of constraints, and probabilistic composition seen as a communication language among models, ensuring a distribution is a constructed so that all constraints are satisfied to form the task distribution of interest. Such programming further requires no *explicit training* or *data*, enabling generalization in inference even on distributions with no previously seen data. We illustrate how such recombination enables generalization to new task distributions in decision making, image and video synthesis.

The underlying compositional components in generative modeling can in many cases be directly inferred and discovered in an unsupervised manner from data, representing compositional structure such as objects and relations. Such discovered components can then be similarly recombined to form new distributions – for instance, objects components discovered by one generative model on one dataset can be combined with components discovered by a separate generative model on another dataset to form hybrid scenes with objects in both datasets. We illustrate the efficacy of such discovered compositional structure across domains in images and trajectory dynamics.

Overall, in this paper, we advocate for the idea that we should construct complex generative systems by representing them as a compositional system of simpler components and illustrate its benefits across various domains.

2. Data Efficient Generative Modeling

The predominant paradigm for training generative models has been to construct increasingly larger monolithic models trained with greater amounts of data and computational power. While language models have demonstrated significant improvements with increased scale (albeit still with difficulty in compositionality (Dziri et al., 2023)), current multimodal models such as DALL-E 3 and GPT-4V remain unable to take advantage of even simple forms of compositionality (Figure 2). Such models may be unable to accurately generate images given combinations of relations rarely seen in training data, or fail to understand simple spatial relations in images, despite being trained on a very significant portion of the existing Internet.

One difficulty is that the underlying sample complexity of learning generative models over joint distributions of variables increases dramatically with the number of variables. As an example, consider learning probability distributions by maximizing log-likelihood over a set of random variables A, B, C, D, each of which can take a set of K values. Directly learning a distribution over a single variable A by requires $O(K)$ values (Canonne, 2020). The data required to learn distributions over a joint set of variables generally increases exponentially – so that learning a joint distribution $p(A, B, C, D)$ requires $O(K^4)$ samples (Canonne, 2020).

Constructing large multimodal generative models such as GPT-4V or DALL-E 3 falls into the same difficulty – as the number of modalities jointly modeled increases, the combination of samples required to see and learn the entire data distribution exponentially increases. This is particularly challenging in the multimodal setting as the existing data on the Internet used to train these models is often highly non-uniform, with many combinations of natural language and images unseen.

One approach to significantly reduce the data necessary to learn generative models over complex joint distributions is factorization – if we know that a distribution exhibits an independence structure between variables such as

$$p(A, B, C, D) \propto p(A)p(B)p(C, D),$$

we can substantially reduce the data requirements by only needing to learn these factors, composing them together to form a more complex distribution. This also enables our learned joint distribution to generalize to unseen combinations of variables so long as each local variable combination is in distribution (illustrated in Figure 3). Even in settings where distributions are not accurately modeled as a product

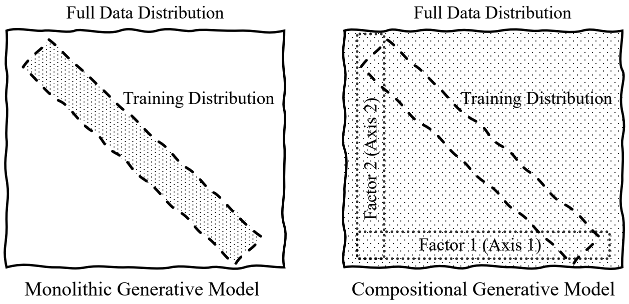


Figure 3. **Generalizing Outside Training Data.** Given a narrow slice of training data, we can learn generative models that generalize outside the data through composition. We learn separate generative models to model each axis of the data – the composition of models can then cover the entire data space.

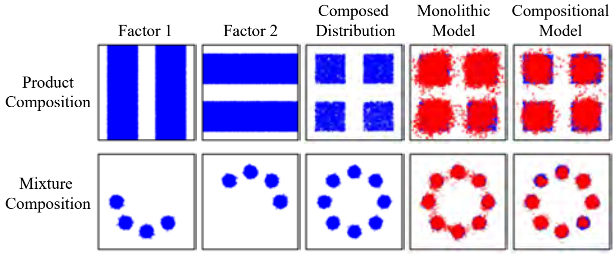


Figure 4. **Distribution Composition** – When modeling simple product (top) or mixture (bottom) compositions, learning two compositional models on the factors is more data efficient than learning a single monolithic model on the product distribution. The monolithic model is trained on twice as much data as individual factors.

of independent factors, such a factorization can still lead to a better models given limited data by reducing the hypothesis space (Murphy, 2022). This idea of factorizing probability distributions has led to substantial work in probabilistic graphical models (PGMs) (Koller & Friedman, 2009).

Below, we illustrate across four settings how representing a target distribution $p(x)$ in a factorized manner can substantially improve generative modeling performance from a limited amount of data:

Simple Distribution Composition. In Figure 4, we consider modeling a distribution $p(x)$ that is a product $p(x) \propto p_1(x)p_2(x)$ or mixture $p(x) \propto p_1(x) + p_2(x)$ of two factors $p_1(x)$ and $p_2(x)$. We compare training either a single model on $p(x)$ or learning two generative models on the factors $p_1(x)$ and $p_2(x)$. We find that training compositional models leads to a more accurate distribution modeling if the same amount of data is used to learn $p(x)$ as is used to learn both $p_1(x)$ and $p_2(x)$. Even when modeling simple distributions, the data complexity of modeling each factor is simpler than representing the joint distribution.

Trajectory Modeling. Next, we consider modeling a probability distribution $p(\tau)$ over trajectories $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$, which many recent works have typically modeled using a single joint distribution $p(s_0, a_0, \ldots, s_T, a_T)$ (Janner et al., 2022; Ajay et al., 2022). In contrast to a monolithic generative distribution, given structural knowledge of the environment – i.e., that it is a Markov Decision Process, a more factorized generative model to represent the distribution is as a product

$$p(\tau) \propto \prod_i p(s_i \mid s_{i-1}, a).$$

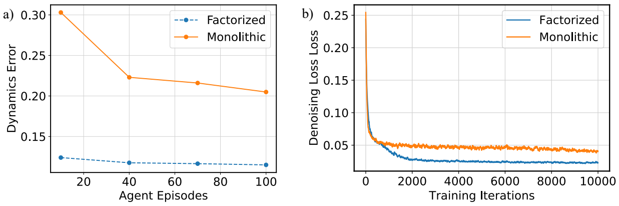


Figure 5. **Compositional Trajectory Generation** – By factorizing a trajectory generative model into a set of components, models are able to more accurately simulate dynamics from limited trajectories (a) and train in fewer training iterations (b).

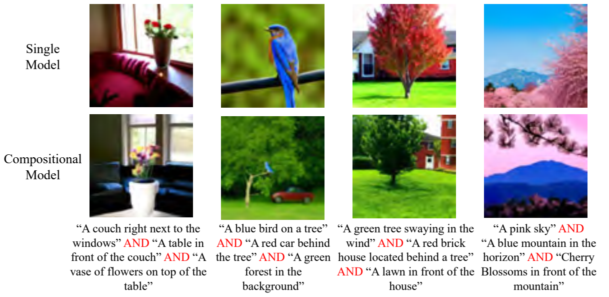

Single Model
Compositional Model
"A couch right next to the windows" AND "A table in front of the couch" AND "A vase of flowers on top of the table"
"A blue bird on a tree" AND "A red car behind the tree" AND "A green forest in the background"
"A green tree swaying in the wind" AND "A red brick house located behind a tree" AND "A lawn in front of the house"
"A pink sky" AND "A blue mountain in the horizon" AND "Cherry Blossoms in front of the mountain"

Figure 6. **Compositional Visual Synthesis.** By composing a set of generative models modeling conditional image distributions given a sentence description, we can more accurately synthesize images given paragraph-level text descriptions. Figure adapted from (Liu et al., 2022)

In Figure 5 , we explore the efficacy of compositional and monolithic models in characterizing trajectories in Maze2D, which consists of a 4D state space (2D position and velocity) and 2D action space (2D forces), using the model in (Janner et al., 2022) (with the compositional model representing trajectory chunksize 8 to ensure compatibility with the architecture). We plot the accuracy of generated trajectories at unseen start states as the function of the number of agent episodes used to train models, where each episode has length of approximately 10000 timesteps. As seen in the Figure 5(a), given only a very limited number of agent episodes in an environment, a factorized model can more accurately simulate trajectory dynamics. In addition, we found that training a single joint generative model also took a substantially larger number of iterations to train than the factorized model as illustrated in Figure 5(b).

Compositional Visual Generation. We further consider modeling a probability distribution $p(x \mid T)$ in text-to-image synthesis, where x is an image and T is a complex text description. While this distribution is usually characterized by a single generative model, we can factor the generation as a product of distributions (Liu et al., 2022) given sentences t_1, t_2, and t_3 in the description T

$$p(x \mid T) \propto p(x \mid t_1)p(x \mid t_2)p(x \mid t_3).$$

This representation of the distribution is more data efficient: we only need to see the full distribution of images given single sentences. In addition, it enables us to generalize to unseen regions of $p(x \mid T)$ such as unseen combinations of sentences and longer text descriptions. In Figure 6, we illustrate the efficacy of such an approach.

Composing Language Models. Finally, we consider modeling a probability distribution $p(x)$ over a language sequence x. Similar to the previous examples, we can represent the likelihood as a composition $p(x) \propto \prod_i p_i(x)$, where each distribution $p_i(x)$ is parameterized by a separate language model. However, directly sampling from such a composition of language models is difficult as it requires intermediate access to the output logits of each model, which are often unavailable for proprietary models. One approach to avoid this issue is to combine outputs of individual language models $p_i(x)$ in the language space and use the result as context for representing the final distribution $p(x)$ over language sequences (Du et al., 2023b).

In Du et al. (2023b), this compositional approach is found to effectively improve the performance of base language models. For instance, on the MATH dataset (Hendrycks et al., 2021), by composing 5 instances of a GPT-3.5 model, we can obtain a final accuracy of $58.0 \pm 2.8\%$, even outperforming a much larger and expensive GPT-4 model, which obtains a performance of $55.0 \pm 2.9\%$.

3. Generalization to New Distributions

In the previous section, we've illustrated how composition can enable us to effectively model a distribution $p(x)$, including areas we have not seen any data in. In this section, we further illustrate how composition enables generalization, allowing us to re-purpose a generative model $p(x)$ to solve a new task by constructing a new generative model $q(x)$.

Consider the task of planning, where we wish to construct a generative model $q(\tau)$ which samples plans that reach a goal state g starting from a start state s. Given a generative model $p(\tau)$, which sample legal, but otherwise unconstrained, state sequences in an environment, we can construct an additional generative model $r(\tau, s, g)$ which has high likelihood when τ has start state s and goal state g and low likelihood everywhere else. By composing the two distributions

$$q(\tau) \propto p(\tau)r(\tau, s, g), \tag{1}$$

we can construct our desired planning distribution $q(\tau)$, exploiting the fact that probability can be treated as a "currency" to combine models, enabling us to selectively choose trajectories that satisfy the constraints in both distributions.

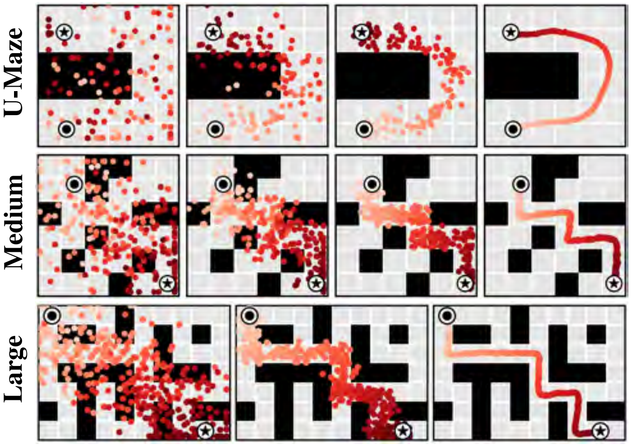
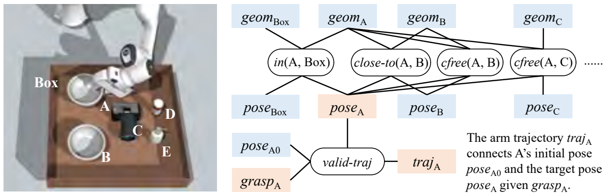


Figure 7. **Planning through Probability Composition.** By composing a probability density trained on modeling dynamics in an environment $p_{traj}(\tau)$ with a probability density $p_{goal}(\tau, g)$ which specifies a specific goal state, we can sample plans from specified start ◉ to a goal ✪ condition. Figure from (Janner et al., 2022), where the horizontal axis illustrates progression of sampling.



(a) Visualization of the environment while placing object A.

(b) Visualization of the constraint graphs associated with the object placement. There are three decision variables.

Figure 8. **Manipulation through Constraint Composition.** New object manipulation problems can be converted into a graph of constraints between variables. Each constraint can be represented as a low-dimensional factor of the joint distribution, with sampling from the composition of distributions corresponding to solving the arrangement problem. Figure adapted from (Yang et al., 2023b).

Below, we illustrate a set of applications where we can construct new compositional generative models $q(x)$ to solve tasks in planning, constraint satisfaction, hierarchical decision-making, and image and video generation.

Planning with Trajectory Composition. We first consider constructing $q(\tau)$ representing planning as described in Equation 1. In Figure 7 we illustrate how sampling from this composed distribution enables successful planning from start to goal states. Quantatively, this approach performs well also as illustrated in (Janner et al., 2022).

Manipulation through Constraint Satisfaction. We next illustrate how we can construct a generative model $q(V)$ to solve a variety of robotic object arrangement tasks. As illustrated in Figure 8, many object arrangement tasks can be formulated as *continuous constraint satisfaction problems* consisting of a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{U}, \mathcal{C} \rangle$, where $v \in \mathcal{V}$ is a

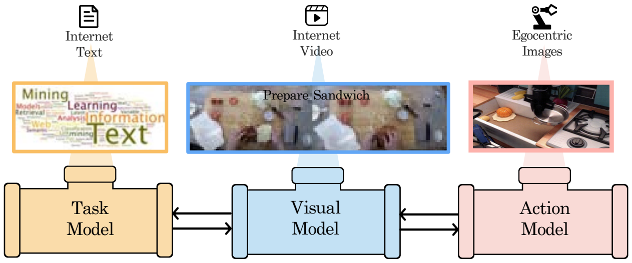


Figure 9. **Hierarchical Planning through Composition.** By composing a set of foundation models trained on Internet data (language, videos, action), we can zero-shot construct a hierarchical planning system. Figure adapted from (Ajay et al., 2023).

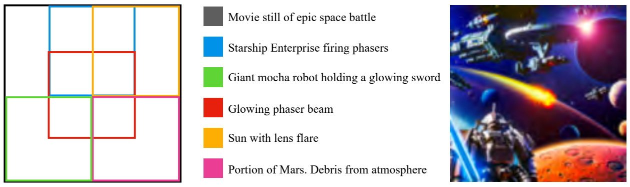

Movie still of epic space battle
Starship Enterprise firing phasers
Giant mocha robot holding a glowing sword
Glowing phaser beam
Sun with lens flare
Portion of Mars. Debris from atmosphere

Figure 10. **Image Tapestries through Composition.** By composing a set of probability distributions defined over different spatial regions in an image, we can construct detailed image tapestries. Figure adapted from (Du et al., 2023a).

decision variable (such as the pose of an object), while each $u \in \mathcal{U}$ is a conditioning variable (such as the geometry of an object) and $c \in \mathcal{C}$ is a constraint such as collision-free. Given such a specification, we can solve the robotics tasks by sampling from the composed distribution

$$q(V) \propto \prod_{c \in \mathcal{C}} p_c(\mathcal{V}^c \mid \mathcal{U}^c),$$

corresponding to solving the constraint satisfaction problem. Such an approach enables effective generalization to new problems (Yang et al., 2023b), to temporally extended plans (Mishra et al., 2023), and the combination of heterogeneous policies (Wang et al., 2024).

Hierarchical Planning with Foundation Models. We further illustrate how we can construct a generative model that functions as a hierarchical planner for long-horizon tasks. We construct $q(\tau_{\text{text}}, \tau_{\text{image}}, \tau_{\text{action}})$, which jointly models the distribution over a text plan τ_{text}, image plan τ_{image}, and action plan τ_{action} given a natural language goal g and image observation o, by combining pre-existing foundation models trained on Internet knowledge. We formulate $q(\tau_{\text{text}}, \tau_{\text{image}}, \tau_{\text{action}})$ through the composition

$$p_{\text{LLM}}(\tau_{\text{text}}, g) p_{\text{Video}}(\tau_{\text{image}}, \tau_{\text{text}}, o) p_{\text{Action}}(\tau_{\text{action}}, \tau_{\text{image}}).$$

This distribution assigns a high likelihood to sequences of natural-language instructions τ_{text} that are plausible ways to reach a final goal g (leveraging textual knowledge embedded in an LLM) which are consistent with visual plans τ_{image} starting from image o (leveraging visual dynamics information embedded in a video model), which are further consistent with execution with actions τ_{action} (leveraging action information in a large action model). Sampling from this distribution then corresponds to finding sequences $\tau_{\text{text}}, \tau_{\text{image}}, \tau_{\text{action}}$ that are mutually consistent with all constraints, and thus constitute successful hierarchical plans to accomplish the task. We provide an illustration of this composition in Figure 9 with efficacy of this approach demonstrated in (Ajay et al., 2023).

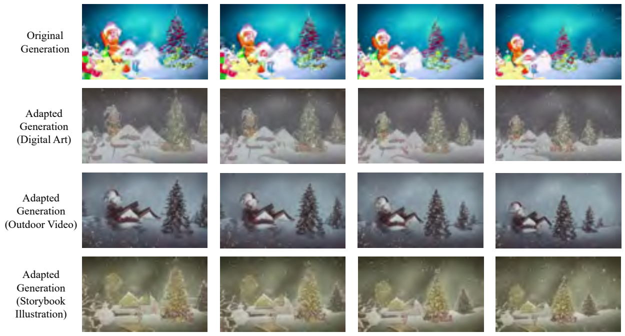

Original Generation
Adapted Generation (Digital Art)
Adapted Generation (Outdoor Video)
Adapted Generation (Storybook Illustration)

Figure 11. **Video Stylization through Composition.** By composing one video model with a model specifying style, we can stylize video generations. Figure adapted from (Yang et al., 2023a).

Controllable Image Synthesis. Composition can also allows us to construct a generative model $q(x \mid D)$ to generate images x from a detailed scene description D consisting of text and bounding-box descriptions $\{\text{text}_i, \text{bbox}_i\}_{i=1:N}$. This compositional distribution is

$$q(x|D) \propto \prod_{i \in \{1,\dots,N\}} p(x_{\text{bbox}_i} \mid \text{text}_i),$$

where each distribution is defined over bounding boxes in an image. In Figure 10, we illustrate the efficacy of this approach for constructing complex images. This approach enables the synthesis of image tapestries (Du et al., 2023a) and collages (Zhang et al., 2023a).

Style Adaptation of Video Models. Finally, composition can be used to construct a generative model $q(\tau)$ that synthesizes video in new styles. Given a pretrained video model $p_{\text{pretrained}}(\tau \mid \text{text})$ and a small video model of a particular style $p_{\text{adapt}}(\tau \mid \text{text})$, we can sample videos τ from the compositional distribution

$$p_{\text{pretrained}}(\tau \mid \text{text}) p_{\text{adapt}}(\tau \mid \text{text})$$

to generate new videos in different specified styles. The efficacy of using composition to adapt the style of a video model is illustrated in (Yang et al., 2023a).

4. Generative Modeling with Learned Compositional Structure

A limitation of compositional generative modeling discussed in the earlier sections is that it requires a priori knowledge about the independence structure of the distribution we wish to model. However, these compositional components can also be discovered jointly while learning a probability

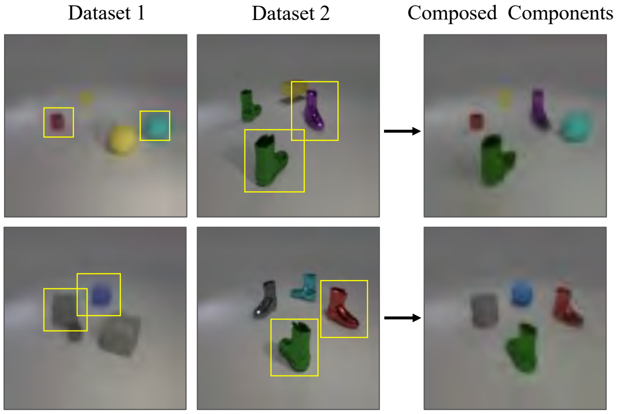


Figure 12. **Composition of Discovered Objects.** Probabilistic components corresponding to individual objects in a scene are discovered unsupervised in two datasets using two separate models. Discovered components (illustrated with yellow boxes) can be multiplied together to form new scenes with a hybrid composition of objects. Figure adapted from (Su et al., 2024).

distribution by formulating maximum likelihood estimation as maximizing the likelihood of the factorized distribution

$$p_\theta(x) \propto \prod_i p_\theta^i(x).$$

Similar to the previous two sections, the discovery of the learned components $p_\theta^i(x)$ enables more data-efficient learning of the generative model as well as the ability to generate samples from new task distributions. Here, we illustrate three examples of how different factors can be discovered in an unsupervised manner.

Discovering Factors from an Input Image. Given an input image x of a scene, we can parameterize a probability distribution over the pixel values of the image as a product of the compositional generative models

$$p_\theta(x) \propto \prod_i p_\theta(x \mid \text{Enc}_i(x)),$$

where $\text{Enc}(\cdot)$ is a learned neural encoder with low-dimensional latent output to encourage each component to capture distinct regions of an image. By training models to autoencode images with this likelihood expression, each component distribution $p_\theta(x \mid \text{Enc}_i(x))$ finds interpretable decomposition of images corresponding to individual objects in a scene as well global factors of variation in the scene such as lighting (Du et al., 2021; Su et al., 2024). In Figure 12, we illustrate how these discovered components, $p_\theta(x \mid z_1)$ and $p_\theta(x \mid z_2)$ from a model trained on cubes and spheres, $p_\phi(x \mid z_3)$ and $p_\phi(x \mid z_4)$ from a separate model trained on trucks and boots can be composed together to form the distribution

$$p_\theta(x \mid z_1)p_\theta(x \mid z_2)p_\phi(x \mid z_3)p_\phi(x \mid z_4),$$

to construct hybrid scenes with objects from both datasets.

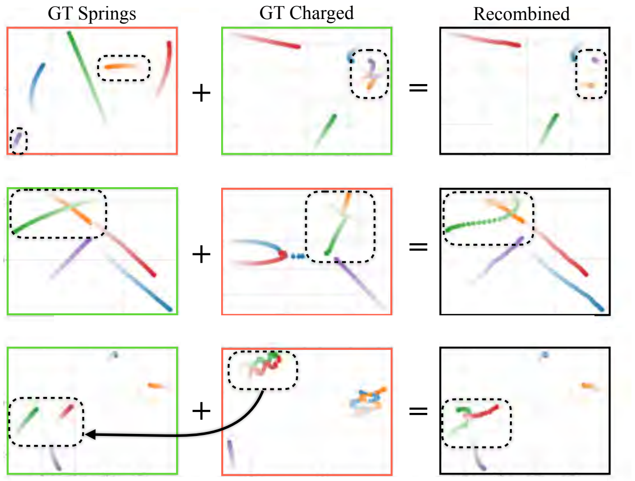


Figure 13. **Composition of Discovered Relation Potentials** In a particle dataset, particles exhibit potentials corresponding to invisible springs between particles (Col. 1) or charges between particles (Col. 2). By swapping discovered probabilistic components between each pair of objects between particle systems, we can recombine trajectories framed in green but with a pair of edge potentials from trajectories formed in red in Col. 3. Figure adapted from (Comas et al., 2023)

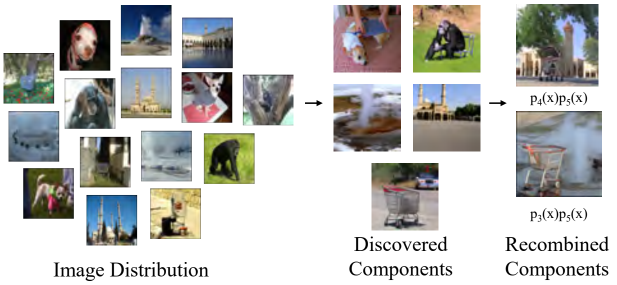


Figure 14. **Discovering Image Classes.** Given a distribution of images drawn from 5 image classes in ImageNet, discovered components correspond to each image class. Components can further be composed together to form new images. Figure adapted from (Liu et al., 2023).

Discovering Relational Potentials. Given a trajectory τ of N particles, we can similarly parameterize a probability distribution over the reconstruction of the particle system as a product of components defined over each pairwise interaction between particles

$$p_\theta(\tau) \propto \prod_{i,j \forall j \neq i} p_\theta(\tau \mid \text{Enc}_{ij}(\tau)),$$

where $\text{Enc}_{ij}(\tau)$ corresponds to latent encoding interactions between particle i and j. In Figure 13, we illustrate how these discovered relational potentials on one particle system can be composed with relational potentials discovered on a separate set of forces to simulate those forces on the particle system.

Discovering Object Classes From Image Distributions. Given a distribution of images $p(x)$ representing images drawn from different classes in Imagenet, we can model the likelihood of the distribution as a composition

$$p_\theta(x) \propto p_\phi(w \mid x) \prod_i p_\theta^i(x)^{w_i},$$

where w_i refers to the weighting coefficient for each component. In Figure 14, we illustrate that the discovered components in this setting represent each of the original Imagenet classes in the input distribution of images. We further illustrate how these discovered components to be composed together to generate images with multiple classes of objects.

5. Implementing Compositional Generation

In this section, we discuss some challenges with implementing compositional sampling with common generative model parameterizations and discuss a generative model parameterization that enables effective compositional generation. We then present some practical implementations of compositional sampling in both continuous and discrete domains.

5.1. Challenges With Sampling from Compositional Distributions

Given two probability densities $p_1(x)$ and $p_2(x)$, it is often difficult to directly sample from the product density $p_1(x)p_2(x)$. Existing generative models typically represent probability distributions in a factorized manner to enable efficient learning and sampling, such as at the token level in autoregressive models (Van Den Oord et al., 2016) or across various noise levels in diffusion models (Sohl-Dickstein et al., 2015). However, depending on the form of the factorization, the models may not be straightforward to compose.

For instance, consider two learned autoregressive factorizations $p_1(x_i|x_{0:i-1})$ and $p_2(x_i|x_{0:i-1})$ over sequences $x_{0:T}$. The autogressive factorization of the product distribution $p_{\text{product}}(x) \propto p_1(x)p_2(x)$ corresponds to

$$p_{\text{product}}(x_i|x_{0:i-1}) = \sum_{x_{i+1:T}} p_1(x_{i+1:T}|x_{0:i})p_1(x_i|x_{0:i-1}) \\ p_2(x_{i+1:T}|x_{0:i})p_2(x_i|x_{0:i-1}),$$

where we need to marginalize over all possible future values of $x_{i+1:T}$. Since this marginalization is different dependent on the value of x_i, $p_{\text{product}}(x_i|x_{0:i-1})$ is not equivalent to $p_1(x_i|x_{0:i-1})p_2(x_i|x_{0:i-1})$ and therefore autoregressive factorizations are not directly compositional. Similarly, two learned score functions from diffusion models are not directly composable as they do not correspond to the noisy gradient of the product distribution (Du et al., 2023a).

While it is often difficult to combine generative models, representing the probability density explicitly enables us to combine models by manipulating the density. One such approach is to represent probability density as an Energy-Based Model, $p_i(x) \propto e^{-E_i(x)}$ (Hinton, 2002; Du & Mordatch, 2019). Under this factorization by definition, we can construct the product density corresponding to

$$e^{-(E_1(x)+E_2(x))} \propto e^{-E_1(x)}e^{-E_2(x)}, \tag{2}$$

corresponding to a new EBM $E_1(x) + E_2(x)$. It is important to observe that EBMs generally represent probability densities in an unnormalized manner, and the product of two normalized probability densities $p_1(x)$ and $p_2(x)$ will be an unnormalized probability density as well (where the normalization constant is intractable to compute as it requires marginalization over the sample space). Additional operations between probability densities such as mixtures and inversions of distributions can also be expressed as combinations of energy functions (Du et al., 2020a).

To generate samples from any EBM distribution, it is necessary to run Markov Chain Monte Carlo (MCMC) to iteratively refine a starting sample to one that is high likelihood (low energy) under the EBM. We present practical MCMC algorithms for sampling from composed distributions in continuous spaces in Section 5.2 and discrete spaces in Section 5.3 with EBMs. Recently, new methods for implementing compositional sampling using separately trained classifiers to efficiently specify each conditioned factor have been developed (Garipov et al., 2023), which we encourage the reader to also read.

5.2. Effective Compositional Sampling on Continuous Distributions

Given a composed distribution represented as EBM $E(x)$ defined over inputs $x \in \mathbb{R}^D$, directly finding a low energy sample through MCMC becomes increasingly inefficient as the data dimension D rises. To more effectively find low-energy samples in EBMs in high-dimensional continuous spaces, we can use the gradient of the energy function to help guide sampling. In Du & Mordatch (2019), Langevin dynamics is used to implement efficient sampling, where a sample can be repeatedly optimized using the expression

$$x_t = x_{t-1} - \lambda \nabla_x E(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma),$$

where x_0 is initialized from uniform noise. By converting different operations such as products, mixtures, and inversions of probability distributions into composite energy functions, the above sampling procedure allows us to effectively compositionally sample from composed distributions (Du et al., 2020a).

There has been a substantial body of recent work on improving learning in EBMs (Du & Mordatch, 2019; Nijkamp et al., 2019; Grathwohl et al., 2019; Du et al., 2020b; Grathwohl et al., 2021) but EBMs still lag behind other generative approaches in efficiency and scalability of training. By leveraging the close connection of diffusion models with EBMs in (Song & Ermon, 2019) we can also directly implement the compositional operations with EBMs with diffusion models (Du et al., 2023a), which we briefly describe below.

Given a diffusion model representing a distribution $p(x)$, we can interpret the T learned denoising functions $\epsilon(x, t)$ of the diffusion model as representing T separate EBM distributions, $e^{-E(x,t)}$, where $\nabla_x E(x, t) = \epsilon(x, t)$. This sequence of EBM distributions transition from $e^{-E(x,T)}$ representing the Gaussian distribution $\mathcal{N}(0, 1)$ to $e^{-E(x,0)}$ representing the target distribution $p_i(x)$. We can draw samples from this sequence of EBMs using annealed importance sampling (Du et al., 2023a), where we initialize a sample from Gaussian noise and sequentially run several steps of MCMC on each EBM distribution, starting at $e^{-E(x,T)}$ and ending at $e^{-E(x,0)}$.

This EBM interpretation of diffusion models allows them to be composed using operations such as Equation 2 by applying the operation to each intermediate EBM corresponding to the component diffusion distributions, for instance $e^{-(E_1(x,k)+E_2(x,k))}$. We can then use an annealed importance sampling procedure on this sequence of composite EBMs. Note that this annealed importance procedure is necessary for accurate compositional sampling – using the reverse diffusion process directly on this composed score does not sample from the composed distribution (Du et al., 2023a).

A variety of different MCMC samplers such as ULA, MALA, U-HMC, and HMC can be used as intermediate MCMC samplers for this sequence of EBM distributions. One easy-to-implement MCMC transition kernel that is easy to understand is the standard diffusion reverse sampling kernel at a fixed noise level. We illustrate in Appendix A that this is equivalent to running a ULA MCMC sampling step. This allows compositional sampling in diffusion models to be easily implemented by simply constructing the score function corresponding to the composite distribution we wish to sample from and then using the standard diffusion sampling procedure, but with the diffusion reverse step applied multiple times at each noise level.

5.3. Effective Compositional Sampling on Discrete Distributions

Given an EBM representing a composed distribution $E(x)$ on a high dimensional discrete landscape, we can use Gibbs sampling to sample from the resultant distribution, where we repeatedly resample values of individual dimensions of x using the marginal energy function $E(x_i \mid x_{-i})$. However, this process is increasingly inefficient as the underlying dimensionality of the data increases.

The use of a gradient of the energy function $E(x)$ to accelerate sampling in the discrete landscape is difficult, as the gradient operation is not well defined in discrete space (though there are also promising discrete analogs of gradient samplers (Grathwohl et al., 2021)). However, we can leverage our learned generative distributions to accelerate sampling, by using one generative model as a proposal distribution and the remaining energy functions to implement a Metropolis-Hastings step (Li et al., 2022; Verkuil et al., 2022).

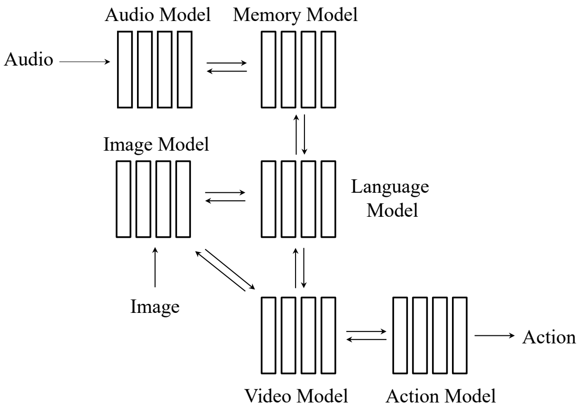


Figure 15. **Decentralized Decision Making.** By composing generative models operating over various modalities we can construct decentralized architectures for intelligent agents. Communication between models is induced by inference over the joint distribution.

As an example, to sample from an energy function $E(x) = E_1(x) + E_2(x)$, given an initial MCMC sample x_t, we can draw a new sample x_{t+1} by sampling from the learned distribution $e^{-E_1(x)}$, and accept the new sample x_{t+1} with a Metropolis acceptance rate

$$a(x_{t+1}) = \text{clip}(e^{E_2(x_t) - E_2(x_{t+1})}, 0, 1).$$

This procedure allows us to leverage $e^{-E_1(x)}$ to guide sampling from $e^{-E(x)}$.

6. Discussion and Future Directions

Most recent research on building generative models has focused on increasing the computational scale and data on which models are trained. We have presented an orthogonal direction to constructing complex generative systems, by building systems *compositionally*, combining simpler generative models to form more complex ones. We have illustrated how this can be more data and computation-efficient to learn, enable flexible reprogramming, and how such components can be discovered from raw data.

Such compositional systems have additional benefits in terms of both buildability and interpretability. As individual models are responsible for independent subsets of data, each model can be built separately and modularly by different institutions. Simultaneously, at execution time, it is significantly easier to understand and monitor the execution of each simpler constituent model than a single large monolithic model.

In addition, such compositional systems can be more environmentally friendly and easier to deploy than large monolithic models. Since individual models are substantially smaller, they can run efficiently using small amounts of computation. In addition, it is more straightforward to deploy separate models across separate computational machines.

In the setting of constructing an artificially intelligent agent, such a compositional architecture may look like a decentralized decision-making system in Figure 15. In this system, separate generative models are responsible for processing each modality an agent receives and other models responsible for decision-making. Sampling composed generative distribution of models corresponds to message passing between models, inducing cross-communication between models similar to a set of daemons communicating with each other (Selfridge, 1988). Individual generative models in this architecture can be substituted with existing models such as LLMs for proposing plausible plans for actions and text-to-video presenting future world states.

Finally, while we have provided a few promising results on applications of compositional generative modeling, there are many limitations to address in future work. First, the current work on compositional modeling assumes a fixed prespecified structure through which models are composed, limiting generalization to new distributions. To flexibly apply compositional models across new tasks, it would be important to construct systems that can instead automatically discover the correct compositional structure between models as well as the appropriate per-model weighting.

Second, current work on discovering compositional structure assumes that data is naturally factorized into an independent product of components. In many real-world settings, gathered data will often exhibit spurious correlations that violate such independence assumptions, causing existing algorithms to fail to discover the correct structure. Exploring more robust approaches to discovering compositional structure such as through prior knowledge or active intervention in the environment are rich directions for future work.

Lastly, while our focus in this position paper has been on combining separately trained generative models, it would be interesting to theoretically characterize compositional generalization in such systems as well as alternative approaches to improve such generalization. Past theoretical work has characterized compositional generalization in additive models (Wiedemer et al., 2024; Lachapelle et al., 2024), and it would be interesting to extend such analysis to compositional generative modeling. Furthermore, it would be interesting to explore adding explicit compositional structure to individual models to improve compositional generalization (Misino et al., 2022; Sehgal et al., 2023).

Acknowledgements

We acknowledge support from NSF grant 2214177; from AFOSR grant FA9550-22-1-0249; from ONR MURI grant N00014-22-1-2740; and from ARO grant W911NF-23-1-0034. Yilun Du is supported by a NSF Graduate Fellowship.

Impact Statement

In this paper, we argue that generative models should be built compositionally, from simpler individual parts and illustrate how this enables from data-efficient generative modeling. As generative models become increasingly deployed into production, we believe that such an approach can significantly broaden the impact of such models, enabling them to be deployed in a variety of domains with limited data.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Ajay, A., Han, S., Du, Y., Li, S., Gupta, A., Jaakkola, T., Tenenbaum, J., Kaelbling, L., Srivastava, A., and Agrawal, P. Compositional foundation models for hierarchical planning. *arXiv preprint arXiv:2309.08587*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Canonne, C. L. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020.

Comas, A., Du, Y., Lopez, C. F., Ghimire, S., Sznaier, M., Tenenbaum, J. B., and Camps, O. Inferring relational potentials in interacting systems. In *International Conference on Machine Learning*, pp. 6364–6383. PMLR, 2023.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Du, Y., Li, S., and Mordatch, I. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020a.

Du, Y., Li, S., Tenenbaum, J., and Mordatch, I. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020b.

Du, Y., Li, S., Sharma, Y., Tenenbaum, B. J., and Mordatch, I. Unsupervised learning of compositional energy concepts. In *Advances in Neural Information Processing Systems*, 2021.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pp. 8489–8510. PMLR, 2023a.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023b.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jian, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D., et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

Epoch. Key trends and figures in machine learning, 2023. URL https://epochai.org/trends. Accessed: 2024-01-23.

Garipov, T., De Peuter, S., Yang, G., Garg, V., Kaski, S., and Jaakkola, T. Compositional sculpting of iterative generative processes. *arXiv preprint arXiv:2309.16115*, 2023.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. Oops i took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pp. 3831–3841. PMLR, 2021.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Lachapelle, S., Mahajan, D., Mitliagkas, I., and Lacoste-Julien, S. Additive decoders for latent variables identification and cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36, 2024.

Li, S., Du, Y., Tenenbaum, J. B., Torralba, A., and Mordatch, I. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Liu, N., Du, Y., Li, S., Tenenbaum, J. B., and Torralba, A. Unsupervised compositional concepts discovery with text-to-image generative models. *arXiv preprint arXiv:2306.05357*, 2023.

Lu, S., Bigoulaeva, I., Sachdeva, R., Madabushi, H. T., and Gurevych, I. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*, 2023.

Majumdar, A., Ajay, A., Zhang, X., Putta, P., Yenamandra, S., Henaff, M., Silwal, S., Mcvay, P., Maksymets, O., Arnaud, S., Yadav, K., Li, Q., Newman, B., Sharma, M., Berges, V., Zhang, S., Agrawal, P., Bisk, Y., Batra, D., Kalakrishnan, M., Meier, F., Paxton, C., Sax, S., and Rajeswaran, A. Openeqa: Embodied question answering in the era of foundation models. 2023.

Mishra, U. A., Xue, S., Chen, Y., and Xu, D. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

Misino, E., Marra, G., and Sansone, E. Vael: Bridging variational autoencoders and probabilistic logic programming. *Advances in Neural Information Processing Systems*, 35: 4667–4679, 2022.

Murphy, K. P. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019.

OpenAI. URL https://openai.com/pricing.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.

Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

Sehgal, A., Grayeli, A., Sun, J. J., and Chaudhuri, S. Neurosymbolic grounding for compositional world models. *arXiv preprint arXiv:2310.12690*, 2023.

Selfridge, O. G. Pandemonium: A paradigm for learning. In *Neurocomputing: Foundations of research*, pp. 115–122. 1988.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Su, J., Liu, N., Tenenbaum, J. B., and Du, Y. Compositional image decomposition with diffusion models, 2024. URL https://openreview.net/forum?id=88FcNOwNvM.

Tamkin, A., Brundage, M., Clark, J., and Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. *bioRxiv*, pp. 2022–12, 2022.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, L., Zhao, J., Du, Y., Adelson, E. H., and Tedrake, R. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36, 2024.

Yadlowsky, S., Doshi, L., and Tripuraneni, N. Pretraining data mixtures enable narrow model selection capabilities in transformer models. *arXiv preprint arXiv:2311.00871*, 2023.

Yang, M., Du, Y., Dai, B., Schuurmans, D., Tenenbaum, J. B., and Abbeel, P. Probabilistic adaptation of text-to-video models. *arXiv preprint arXiv:2306.01872*, 2023a.

Yang, Z., Mao, J., Du, Y., Wu, J., Tenenbaum, J. B., Lozano-Pérez, T., and Kaelbling, L. P. Compositional diffusion-based continuous constraint solvers. *arXiv preprint arXiv:2309.00966*, 2023b.

Zhang, Q., Song, J., Huang, X., Chen, Y., and Liu, M.-Y. Diffcollage: Parallel generation of large content with diffusion models. *arXiv preprint arXiv:2303.17076*, 2023a.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023b.

Appendix

A. Implementing ULA Transitions as Multiple Reverse Diffusion Steps

We illustrate how a step of reverse sampling on a diffusion model at a fixed noise level is equivalent to ULA MCMC sampling at the same fixed noise level. We use the α_t and β_t formulation from (Ho et al., 2020). The reverse sampling step on an input x_t at a fixed noise level at timestep t is given by a Gaussian with a mean

$$\mu_\theta(x_t, t) = x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t).$$

with the variance of β_t (using the variance small noise schedule in (Ho et al., 2020)). This corresponds to a sampling update,

$$x_{t+1} = x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) + \beta_t \xi, \quad \xi \sim \mathcal{N}(0, 1).$$

Note that the expression $\frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}$ corresponds to the score $\nabla_x p_t(x)$, through the denoising score matching objective (Vincent, 2011), where the EBM $p_t(x)$ corresponds to the data distribution perturbed with t steps of noise. The reverse sampling step can be equivalently written as

$$x_{t+1} = x_t - \beta_t \nabla_x p_t(x) + \beta_t \xi, \quad \xi \sim \mathcal{N}(0, 1). \tag{A1}$$

The ULA sampler draws an MCMC sample from the EBM probability distribution $p_t(x)$ using the expression

$$x_{t+1} = x_t - \eta \nabla_x p_t(x) + \sqrt{2}\eta \xi, \quad \xi \sim \mathcal{N}(0, 1), \tag{A2}$$

where η is the step size of sampling.

By substituting $\eta = \beta_t$ in the ULA sampler, the sampler becomes

$$x_{t+1} = x_t - \beta_t \nabla_x p_t(x) + \sqrt{2}\beta_t \xi, \quad \xi \sim \mathcal{N}(0, 1). \tag{A3}$$

Note the similarity of ULA sampling in Eqn A3 and the reverse sampling procedure in Eqn A1, where there is a factor of $\sqrt{2}$ scaling of the added Gaussian noise in the ULA sampling procedures. This means that we can implement the ULA sampling by running the standard reverse process, but by scaling the noise added in each timestep by a factor of $\sqrt{2}$. Alternatively, we can directly we can directly use the reverse sampling procedure in Eqn A1 to run ULA, where this then corresponds to sampling a tempered variant of $p_t(x)$ with temperature $\frac{1}{\sqrt{2}}$ (corresponding to less stochastic samples from the composed probability distribution).