Compositional Visual Generation and Inference with Energy Based Models

Yilun Du [1] **Shuang Li** [1] **Igor Mordatch** [2]

Abstract

A vital aspect of human intelligence is the ability to compose increasingly complex concepts out of simpler ideas, enabling both rapid learning and adaptation of knowledge. In this paper we show that energy-based models can exhibit this ability by directly combining probability distributions. Samples from the combined distribution correspond to compositions of concepts. For example, given a distribution for smiling faces, and another for male faces, we can combine them to generate smiling male faces. This allows us to generate natural images that simultaneously satisfy conjunctions, disjunctions, and negations of concepts. We evaluate compositional generation abilities of our model on the CelebA dataset of natural faces and synthetic 3D scene images. We also demonstrate other unique advantages of our model, such as the ability to continually learn and incorporate new concepts, or infer compositions of concept properties underlying an image.

1 Introduction

Humans are able to rapidly learn new concepts and continuously integrate them among prior knowledge. The core component in enabling this is the ability to compose increasingly complex concepts out of simpler ones as well as recombining and reusing concepts in novel ways (Fodor & Lepore, 2002). By combining a finite number of primitive components, humans can create an exponential number of new concepts, and use them to rapidly explain current and past experiences (Lake et al., 2017). We are interested in enabling such capabilities in machine learning systems, particularly in the context of generative modeling.

Past efforts have attempted to enable compositionality in several ways. One approach decomposes data into disentangled factors of variation and situate each datapoint in the resulting - typically continuous - factor vector space (Vedantam et al., 2018; Higgins et al., 2018). The factors can either be explicitly provided or learned in an unsupervised manner. In both cases, however, the dimensionality of the factor vector space is fixed and defined prior to training. This makes it difficult to introduce new factors of variation, which may be necessary to explain new data, or to taxonomize past data in new ways. Another approach to incorporate the compositionality is to spatially decompose an image into a collection of objects, each object slot occupying some pixels of the image defined by a segmentation mask (van Steenkiste et al., 2018; Greff et al., 2019). Such approaches can generate visual scenes with multiple objects, but may have difficulty in generating interactions between objects. These two incorporations of compositionality are considered distinct, with very different underlying implementations.

In this work, we propose to implement the compositionality via energy based models (EBMs). Instead of an explicit vector of factors that is input to a generator function, or object slots that are blended to form an image, our unified treatment defines factors of variation and object slots via energy functions. Each factor is represented by an individual scalar energy function that takes as input an image and outputs a low energy value if the factor is exhibited in the image. Images that exhibit the factor can then be generated implicitly through an Markov Chain Monte Carlo (MCMC) sampling process that minimizes the energy. Importantly, it is also possible to run MCMC process on some *combination* of energy functions to generate images that exhibit multiple factors or multiple objects, in a globally coherent manner.

There are several ways to combine energy functions. One can add or multiply distributions as in mixtures (Shazeer et al., 2017; Greff et al., 2019) or products (Hinton, 2002) of experts. We view these as probabilistic instances of logical operators over concepts. Instead of using only one, we consider three operators: logical conjunction, disjunction, and negation (illustrated in Figure 1). We can then flexibly and recursively combine multiple energy functions via these operators. More complex operators (such as implication) can be formed out of our base operators.

EBMs with such composition operations enable a unique continual learning capability. Our formulation defines concepts or factors implicitly via examples, rather than predeclaring an explicit latent space ahead of time. For example, we can create an EBM for concept "black hair" from

[1]MIT CSAIL [2]Google Brain. Correspondence to: Yilun Du <yilundu@mit.edu>, Igor Mordatch <imordatch@google.com>.

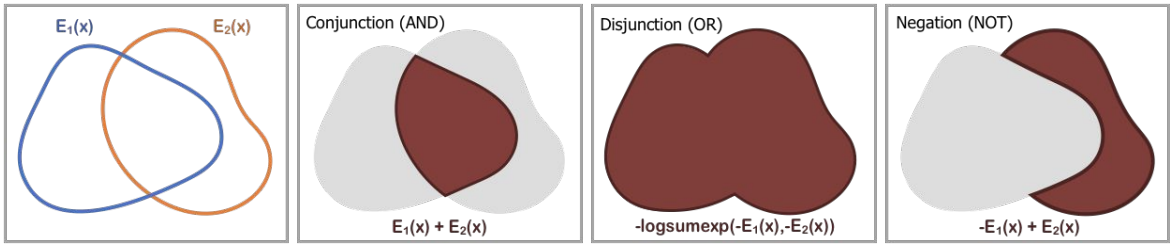


Figure 1: Illustration of logical composition operators over energy functions E_1 and E_2 (drawn as level sets).

a dataset of face images that share this concept. New concepts (or factors), such as hair color can be learned by simply adding a new energy function and can then be combined with energies for previously trained concepts. This process can repeat continually. This view of few-shot concept learning and generation is similar to work of (Reed et al., 2017), with the distinction that instead of learning to generate holistic images from few examples, we learn *factors* from examples, which can be composed with other factors. A related advantage is that finely controllable image generation can be achieved by specifying the desired image via a collection of logical clauses, with applications to neural scene rendering (Eslami et al., 2018).

Our contributions are as follows: first, while composition of energy-based models has been proposed in abstract settings before (Hinton, 2002), we show that it can be used to generate plausible natural images. Second, we propose to combine energy models based on logical operators which can be chained recursively, allowing controllable generation based on a collection of logical clauses. Third, we demonstrate unique advantages of such an approach, such as extrapolation to concept combinations, continual incorporation of new energy functions, and the ability to infer concept properties.

2 Method

In this section, we first give an overview of the Energy-Based Model formulation we use and introduce three logical operators over these models.

2.1 Energy Based Models

EBMs represent data by learning an unnormalized probability distribution across the data. For each data point $\mathbf{x}$, an energy function $E_\theta(\mathbf{x})$, parameterized by a neural network, outputs a scalar real energy such that

$$p_\theta(x) \propto e^{-E_\theta(x)}. \tag{1}$$

To train an EBM on a data distribution p_D, we follow the methodology defined in (Du & Mordatch, 2019), where a Monte Carlo estimate (Equation 2) of maximum likelihood $\mathcal{L}$ is minimized with the following gradient

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{x^+ \sim p_D} E_\theta(x^+) - \mathbb{E}_{x^- \sim p_\theta} E_\theta(x^-). \tag{2}$$

To sample x^- from p_θ for both training and generation, we use MCMC based off Langevin dynamics (Welling & Teh, 2011). Samples are initialized from uniform random noise and are iteratively refined following Equation 3

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k-1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda), \tag{3}$$

where k is the k^{th} iteration step and λ is the step size. We refer to each iteration of Langevin dynamics as a negative sampling step. We note that this form of sampling allows us to use the gradient of the combined distribution to generate samples from distributions composed of p_θ and the other distributions. We use this ability to generate from multiple different compositions of distributions.

2.2 Composition of Energy-Based Models

We next present different ways that EBMs can compose. We consider a set of independently trained EBMs, $E(\mathbf{x}|c_1), E(\mathbf{x}|c_2), \ldots, E(\mathbf{x}|c_n)$, which are learned conditional distributions on underlying latent codes c_i. Latent codes we consider include position, size, color, gender, hair style, and age, which we also refer to as concepts. Figure 2 shows three concepts and their combinations on the CelebA face dataset and attributes.

Concept Conjunction In concept conjunction, given separate independent concepts (such as a particular gender, hair style, or facial expression), we wish to construct an output with the specified gender, hair style, and facial expression – the combination of each concept. Since the likelihood of an output given a set of specific concepts is equal to the product of the likelihood of each individual concept, we have Equation 4, which is also known as the product of experts (Hinton, 2002):

$$p(x|c_1 \text{ and } c_2, \ldots, \text{ and } c_i) = \prod_i p(x|c_i) \propto e^{-\sum_i E(x|c_i)}. \tag{4}$$

We can thus apply Equation 3 to the distribution that is the sum of the energies of each concept to obtain Equation 5 to sample from the joint concept space with $\omega^k \sim \mathcal{N}(0, \lambda)$.

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} \sum_i E_\theta(\tilde{\mathbf{x}}^{k-1}|c_i) + \omega^k. \tag{5}$$

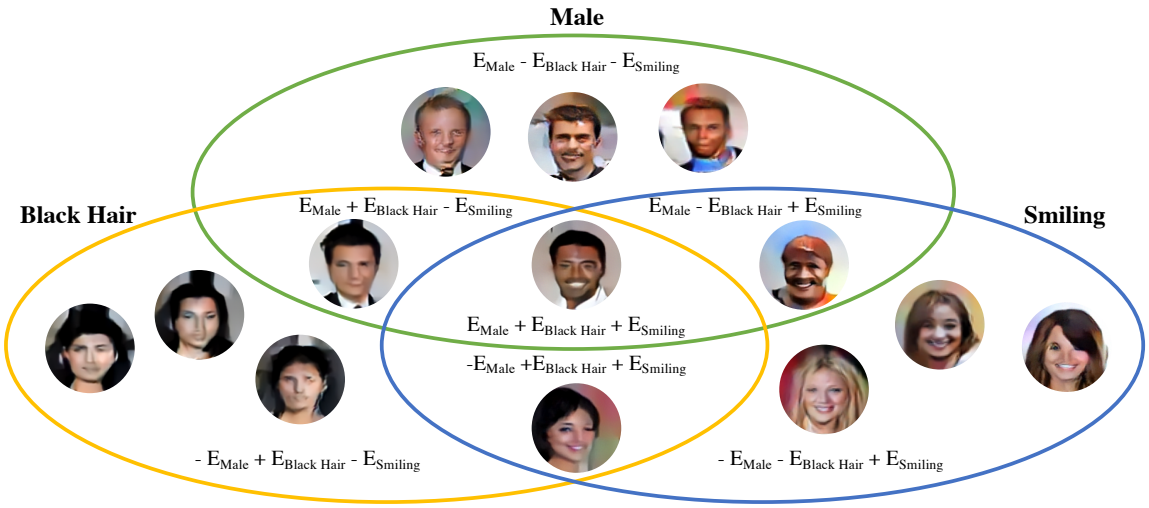


Figure 2: Concept conjunction and negation. All the images are generated through the conjunction and negation of energy functions. For example, the image in the central part is the conjunction of male, black hair, and smiling energy functions.

Concept Disjunction In concept disjunction, given separate concepts such as the colors red and blue, we wish to construct an output that is either red or blue. We wish to construct a new distribution that has probability mass when any chosen concept is true. A natural choice of such a distribution is the sum of the likelihood of each concept:

$$p(x|c_1 \text{ or } c_2, \dots \text{ or } c_i) \propto \sum_i p(x|c_i)/Z(c_i). \quad (6)$$

where $Z(c_i)$ denotes the partition function for each concept. If we assume all partition functions $Z(c_i)$ to be equal, this simplifies to

$$\sum_i p(x|c_i) \propto \sum_i e^{-E(x|c_i)} = e^{\text{logsumexp}(-E(x|c_i))}, \quad (7)$$

where $\text{logsumexp}(f_1, \dots, f_N) = \log \sum_i \exp(f_i)$. We can thus apply Equation 3 to the distribution that is a negative smooth minimum of the energies of each concept to obtain Equation 8 to sample from the disjunction concept space:

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}}\text{logsumexp}(-E(x|c_i)) + \omega^k, \quad (8)$$

where $\omega^k \sim \mathcal{N}(0, \lambda)$. In our experiments, we empirically found the partition function $Z(c_i)$ estimates to be similar across concepts (see Appendix), justifying the simplification of Equation 7.

Concept Negation In concept negation, we wish to generate an output that does not contain the concept. Given a color red, we want an output that is of a different color, such as blue. Thus, we want to construct a distribution that places high likelihood to data that is outside a given concept. One choice is a distribution inversely proportional to the concept. Importantly, negation must be defined with respect to another concept to be useful. The opposite of alive may be dead, but not inanimate. Negation without a data distribution is not integrable and leads to a generation of chaotic textures which, while satisfying absence of a concept, is not desirable. Thus in our experiments with negation we combine it with another concept to ground the negation and obtain an integrable distribution:

$$p(x|\text{not}(c_1), c_2) \propto \frac{p(x|c_2)}{p(x|c_1)^\alpha} \propto e^{\alpha E(x|c_1) - E(x|c_2)}. \quad (9)$$

We found relative smoothing parameter α to be a useful regularizer (when $\alpha = 0$ we arrive at uniform distribution) and we use $\alpha = 0.01$ in our experiments. The above equation allows us to apply Langevin dynamics to obtain Equation 10 to sample concept negations.

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}}(\alpha E(x|c_1) - E(x|c_2)) + \omega^k, \quad (10)$$

where $\omega^k \sim \mathcal{N}(0, \lambda)$. We note that the combinations of conjunctions, disjunctions, and negations allow us to specify more complex logical operators such as implication, but we leave exploration of this to future work.

Concept Inference Our formulation allows us to easily infer the latent concept parameters through which a given input is generated. Given several example inputs of an underlying concept, we wish to combine the data to make an informed estimation of the underlying concept. Assuming each input is independent of each other, the overall likelihood of the inputs is equivalent to the product of likelihood of each input under a concept, and thus is the conjunction of likelihood for each individual data point

$$p(x_1, x_2, \dots, x_n|c) \propto e^{-\sum_i E(x_i|c)}. \quad (11)$$

We can then obtain maximum a posteriori (MAP) estimates of concept parameters by minimizing the logarithm of the

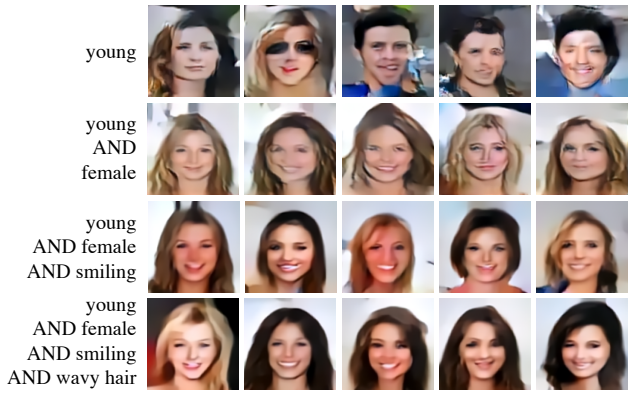


Figure 3: Combinations of different attributes on CelebA via concept conjunction. Each row adds an additional energy function. Images on the first row are only conditioned on young, while images on the last row are conditioned on young, female, smiling, and wavy hair.

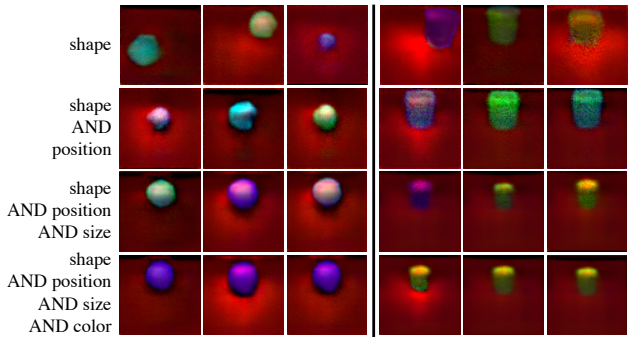


Figure 4: Combinations of different attributes on MuJoCo via concept conjunction. Each row adds an additional energy function. Images on the first row are only conditioned on shape, while images on the last row are conditioned on shape, position, size, and color. The left part is the generation of a sphere shape and the right is a cylinder.

above expression, again assuming that partition functions are equal (justified in the appendix)

$$c(x_1, x_2, \ldots, x_n) = \arg\min_c \sum_i E(x_i|c). \tag{12}$$

3 Experiments

We perform empirical studies to answer the following questions: (1) Can EBMs exhibit concept compositionality (such as concept negation, conjunction, and disjunction) in generating images? (2) Can we take advantage of concept combinations to learn new concepts in a continual manner? (3) Does explicit factor decomposition enable generalization to novel combinations of factors? (4) Can we perform concept inference across multiple inputs?

3.1 Setup

We perform experiments on 64x64 object scenes rendered in MuJoCo (Todorov et al., 2012) (MuJoCo Scenes) and the 128x128 CelebA dataset. For MuJoCo Scene images, we generate a central object of shape either sphere, cylinder, or box of varying size and color at different positions, with some number of (specified) additional background objects. Images are generated with varying lighting and objects.

Table 1: Quantitative evaluation of conjunction, disjunction and negation generations on the Mujoco Scenes dataset using an EBM. Each individual attribute (Color or Position) generation is a individual EBM. (Acc: accuracy)

Model	**Position Acc**	**Color Acc**
Color	0.128	0.997
Position	0.984	0.201
Conjunction(Position, Color)	0.801	0.8125
Conjunction(Position, Negation(Color))	0.872	0.096
Conjunction(Negation(Position), Color)	0.033	0.971
Model	**Position 1 Acc**	**Position 2 Acc**
Position 1	0.875	0.0
Position 2	0.0	0.817
Disjunction (Position 1, Position 2)	0.432	0.413

We use the ImageNet32x32 architecture and ImageNet128x128 architecture from (Du & Mordatch, 2019) with the Swish activation (Ramachandran et al., 2017) on MuJoCo and CelebA datasets. Models are trained on MuJoCo datasets for up to 1 day on 1 GPU and for 1 day on 8 GPUs for CelebA. More training details and model architecture can be found in the appendix.

3.2 Compositional Generation

Quantitative evaluation. We first evaluate compositionality operations of EBMs from Section 2.2. To quantitatively evaluate generation, we use the MuJoCo Scenes dataset. We train a supervised classifier to predict position and color on the MuJoCo Scenes dataset, 99.3% for position and 99.9% for color on the test set. We also train seperate conditional EBMs on the concepts of position and color. For a given positional generation then, if the predicted position (obtained from a supervised classifier on generated images) and original conditioned generation position is smaller than 0.4, then a generation is consider correct. A color generation is correct if the predicted color is the same as the conditioned generation color.

In Table 1, we quantitatively evaluate the quality of generated images given combinations of conjunction, disjunction, and negation on the color and position concepts. When using either Color or Position EBMs, the respective accuracy is high. Conjunction(Position, Color) has high position and color accuracies which demonstrates that an EBM can combine different concepts. Under Conjunction(Position, Negation(Color)), the color accuracy drops to below that of Color EBM. This means negating a concept reduces the likelihood of the concept. The same conclusion follows for Conjunction(Negation(Position), Color).

To evaluate disjunction, we set Position 1 to be a random point in bottom left corner of a grid and Position 2 to be a

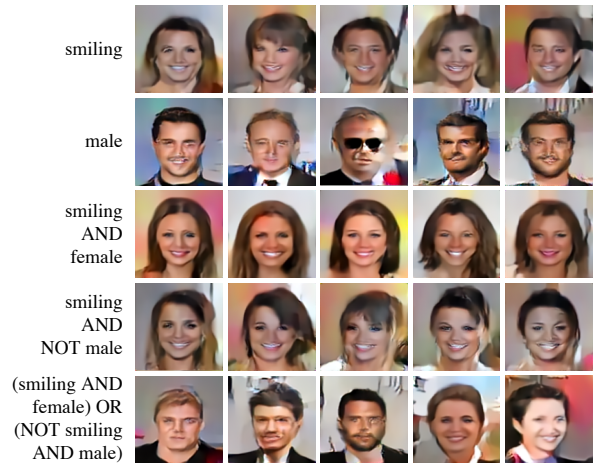


Figure 5: Examples of recursive compositions of disjunction, conjunction, and negation on the CelebA dataset.

random point in the top right corner of a grid. Averages over 1000 generated images are reported in Table 1. Position 1 EBM or Position 2 EBM is able to obtain high accuracy in predicting their own positions. Disjunction(Position 1, Position 2) EBM can generate images that are roughly evenly distributed between Position 1 and Position 2, indicating the disjunction is able to combine concepts additively (generate images that are either concept A or concept B).

Qualitative evaluation. We further provide qualitative visualizations of conjunction, disjunction, and negation operations on both MuJoCo Scenes and CelebA datasets.

Concept Conjunction: In Figure 3, we show the conjunction of EBMs are able to combine multiple independent concepts, such as age, gender, smile, and wavy hair, and get more precise generations when combining more energy models of different concepts. Similarily, EBMs can combine independent concepts of shape, position, size, and color to get more precise generations in Figure 4. We also show results of conjunction with other logical operators in Figure 5.

Concept Negation: In Figure 5, row 4 shows images that are opposite to the trained concept using negation operation. Since concept negation operation should accompany with another concept as described in Section 2.2, we use "smiling" as the second concept. The images in row 4 shows the negation of male AND smiling is smiling female. This can further be combined with disjunction in the row 5 to make either "non-smiling male" or "smiling female".

Concept Disjunction: The last row of Figure 5 shows EBMs can combine concepts additively (generate images that are concept A or concept B). By constructing sampling using logsumexp, EBMs can sample an image that is "not smiling male" or "smiling female", where both "not smiling male" and "smiling female" are specified through the conjunction of energy models of the two different concepts.

Multiple object combination: We show that our composition operations not only combine object concepts or attributes, but also on the object level. To verify this, we constructed a dataset with one green cube and a large amount background clutter objects (which are not green) in the scene. We train a conditional EBM (conditioned on position) on the dataset. Figure 6 "cube 1" and "cube 2" are the generated images conditioned on different positions. We perform the conjunction operation on the EBMs of "cube 1" and "cube 2" and use the combined energy model to generate images (row 3). We find that adding two conditional EBMs allows us to selectively generate two different cubes. Furthermore, such generation satisfies the constraints of the dataset. For example, when two conditional cubes are too close, the conditionals EBMs are able to default and just generate one cube like the last image in row 3.

3.3 Continual Learning

We evaluate to what extent compositionality in EBMs enables continual learning of new concepts and their combination with previously learned concepts. If we create an EBM for a novel concept, can it be combined with previous EBMs that have never observed this concept in their training data? And can we continually repeat this process? To evaluate this, we use the following methodology on MuJoCo dataset:

1. We first train a position EBM on a dataset of varying positions, but a fixed color and a fixed shape. In experiment, we use shape "cube" and color "purple". The position EBM allows us generate a purple cube at various positions. (Figure 7 row 1).

2. Next we train a shape EBM by training the model in combination with the position EBM to generate images of different shapes at different positions. But we do not train position EBM. As shown in Figure 7 row 2, after combining the position and shape EBMs, the "sphere" is placed in the same position as "cubes" in row 1 even these "sphere" positions never be seen during training.

3. Finally we train a color EBM in combination with both position and shape EBMs to generate images of different shapes at different positions and colors. Again we fix both position and shape EBMs, and only train the color model. In Figure 7 row 3, the objects with different color have the same position as row 1 and same shape as row 2 which shows the EBM can continual learning different concepts and extrapolate new concepts in combination with previously learned concepts to generate new images.

In Table 2, we quantitatively evaluate the continuous learning ability of our EBM and GAN (Radford et al., 2015). Similar to the quantitative evaluation in Section 2.2, we a train three classifiers for position, shape, color respectively. For fair comparison, the GAN model is also trained sequentially on the position, shape, and color datasets (with the

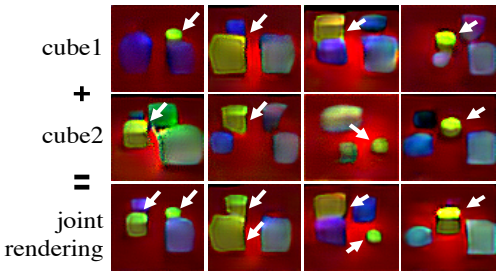


Figure 6: Multi-object compositionality with EBMs. An EBM is trained to generate a green cube of specified size and shape in a scene alongside other objects. At test time, we sample from the conjunction of two EBMs conditioned on different positions and sizes (cube 1 and cube 2) and generates cubes at both locations. Two cubes are merged into one if they are too close (last column).

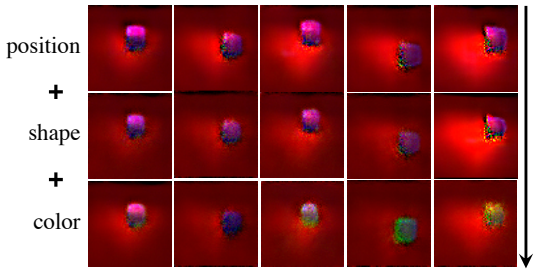


Figure 7: Continual learning of concepts. A position EBM is first trained on one shape (cube) of one color (purple) at different positions. A shape EBM is then trained on different shapes of one fixed color (purple). Finally, a color EBM is trained on shapes of many colors. EBMs can continually learn to generate many shapes (cube, sphere) with different colors at different positions.

Table 2: Quantitative evaluation of continual learning. A position EBM is first trained on "purple" "cubes" at different positions. A shape EBM is then trained on different "purple" shapes. Finally, a color EBM is trained on shapes of many colors with Earlier EBMs are fixed and combined with new EBMs. We compare with a GAN model (Radford et al., 2015) which is also trained on the same position, shape and color dataset. EBMs is better at continually learning new concepts and remember the old concepts. (Acc: accuracy)

Model	Position Acc	Shape Acc	Color Acc
EBM (Position)	0.901	-	-
EBM (Position + Shape)	0.813	0.743	-
EBM (Position + Shape + Color)	0.781	0.703	0.521
GAN (Position)	0.941	-	-
GAN (Position + Shape)	0.111	0.977	-
GAN (Position + Shape + Color)	0.117	0.476	0.984

corresponding position, shape, color available and other attributes set to random to match the training in EBMs).

The position accuracy of EBM doesn't drop significantly when continually learning new concepts (shape and color) which shows our EBM is able to extrapolate earlier learned concepts by combining them with newly learned concepts. In contrast, while the GAN model is able to learn the attributes of position, shape and color models given the corresponding dataset. We find the accuracies of position and shape drops significantly after learning color. The bad performance of previously learned concepts upon learning new concept shows that GANs cannot combine the newly learned attributes with the previous attributes.

3.4 Cross Product Extrapolation

Humans are endowed with the ability to extrapolate novel concept combinations when only a limited number of combinations were originally observed. For example, despite never having seen a "purple cube", a human can compose what it looks like based on the previously observation of "red cube" and "purple sphere".

We evaluate the extrapolation ability of EBMs. We construct a dataset of MuJoCo scene images with spheres of all possible sizes appearing only in the top right corner of the scene and spheres of only large size appearing in the remaining positions. The left figure in Figure 8 shows how does the scene looks like. For the spheres only in the top right corner of the scene, we design different settings. For example, 1% meaning only 1% of positions starting from top right corner with all sphere sizes are used for training. At test time, we evaluate the generation of spheres of all sizes at the positions not seen during the training time. Similar to 1%, 10% and 100% mean the spheres of all sizes appears only in the top right 10% and 100% of the scene. The task is to test the quality of generated objects with unseen size and position combinations. This requires the model to extrapolate the learned position and size concepts in novel combinations.

We train two EBMs on this dataset. One is conditioned on the position latent and trained only on large sizes and another is conditioned on the size latent and trained at the aforementioned percentage of positions. Conjunction of the two EBMs is fine-tuned on this dataset using the gradient descent in Equation 2. We compare this composed model with a baseline holistic model conditioned on both position and size jointly. The baseline is trained on the same position and size combinations and optimized directly from the Mean Squared Error of the generated image and real image. Both models use the same architecture and number of parameters (described in Appendix).

We qualitatively compare the EBM and baseline in Figure 8. When sphere of all sizes are only distributed in the 1% of possible locations, both the EBM and baseline have bad performance. This is because the very few size and position combinations make both of them fail on extrapolation. For the 10% setting, our EBM is better than baseline. EBM is possible to combine concepts to form images from few combination examples by learning an independent model for each concept factor. Both EBM and baseline models generate accurate images when given examples of all combinations of sizes and positions (100% setting), but our EBM is more close to ground truth than baseline.

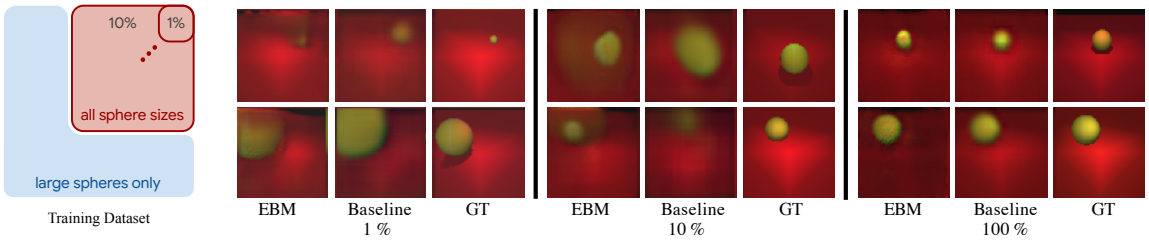


Figure 8: Cross product extrapolation. Left: the spheres of all sizes only appear in the top right corner (1%, 10%, . . .) of the scene and the remaining positions only have large size spheres. Right: generated images of novel size and position combinations using EBM and the baseline model (left).

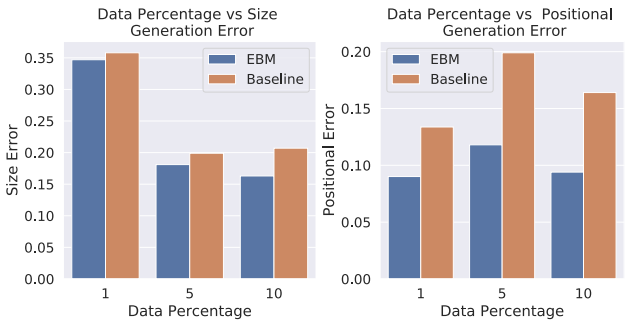


Figure 9: Cross product extrapolation results with respect to the percentages of areas on the top right corner. EBM has lower size and position errors which means EBM is able to extrapolate better with less data than the baseline model.

In Figure 9, we quantitatively evaluate the extrapolation ability of EBM and the baseline. We train a regression model that outputs both the position and size of a generated sphere image. We compute the error between the predicted size and ground truth size and report it in the first image of Figure 9. The position error is in the second image. EBMs are able to extrapolate both position and size better than the baseline model with smaller errors. The size errors go down given larger percentage area of all sphere sizes. For position error, both EBM and the baseline model have smaller errors at 1% data than 5% or 10% data. This result is due to the make-up of the data – with 1% data, only 1% of the rightmost sphere positions have different size annotations, so the models generate large spheres at the conditioned position which are closer to the ground truth position since most positions (99%) are large spheres.

3.5 Concept Inference

In addition to generation, EBMs can be used to infer the underlying concepts given an image. For example, maximum likelihood concept is inferred by minimizing Equation 12. We evaluate inference on an EBM trained on object position, which takes an image and an object position (x,y in 2D) as input and outputs an energy. By iterating densely over all positions (20 by 20 grid positions), we can select the position with the minimal energy as our inference result.

Table 3: Position error on different test datasets. "Test" has the same data distribution with training set. Other datasets change one environmental parameter, e.g. color, size, type, and light, which are unseen in the training set. "Avg" is the average error of "Color", "Light", "Size", and "Type". "Steps"indicates the number of sampling steps used to train the EBM. EBMs are able to generalize better on unseen datasets. Larger number of sampling steps significantly decrease overall EBM error.

Model	Steps	Color	Light	Size	Type	Avg	Test
EBM	200	10.899	6.307	8.431	6.304	7.985	3.903
EBM	400	**4.084**	**4.033**	**6.853**	**3.694**	**4.666**	**2.917**
Resnet	-	20.002	5.881	10.378	6.310	10.643	3.635
PixelCNN	-	60.607	58.589	33.889	48.138	50.306	43.460

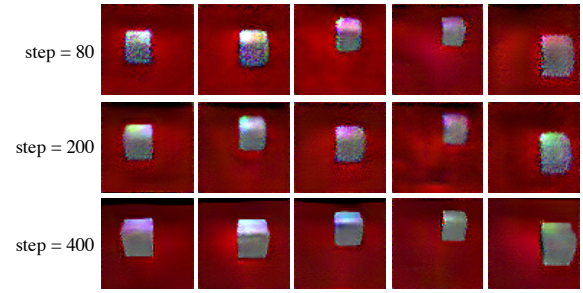


Figure 10: Examples of generated images with varying number of Langevin Dynamics sampling steps. Larger number of steps leads to more realistic images.

We evaluate this result, comparing the predicted position and ground truth object position and compute the inference error using Mean Absolute Error.

We generate a new MuJoCo Scene dataset for training. Each scene has varying lighting conditions with one object, either sphere or cube, at all possible positions and some sizes. To evaluate the performance and generalization ability of EBMs on concept inference, we build several different datasets. The easiest one is *"Test"* which has the same data distribution with the training dataset. The *"Size"* test dataset contains objects twice the size of training objects. *"Color"* dataset has object colors never been seen during training. *"Light"* is a test dataset with different light sources and *"Type"* dataset consists of cylinder images while the training images are only spheres or cubes.

We compare EBMs with two baseline models, ResNet model

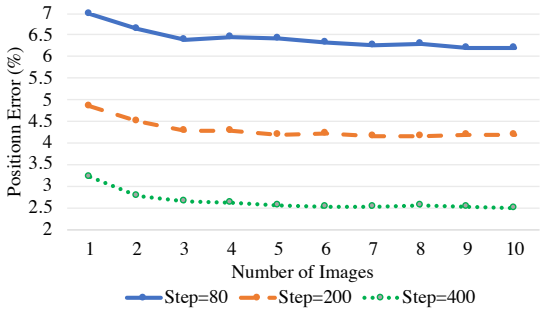


Figure 11: Concept inference from multiple observations. Multiple images are generated under different size, shape, camera view points, and lighting conditions. The position prediction error decreases when the number of input images increases with different Langevin Dynamics sampling steps.

(He et al., 2016) (with the same architecture as EBM) and PixelCNN (Oord et al., 2016). Table 3 shows the comparison results using different number of Langevin Dynamics sampling steps (k in Equation 3). Larger Langevin sampling steps have better performance. Figure 10 shows larger Langevin sampling steps generate better images. We find EBM (step=400) performs better than ResNet and PixelCNN baselines on both the "Test" set and other generalization datasets.

Concept Inference from Multiple Observations The composition rules in Section 2.2 apply directly to inference. When given several different views of an object at a particular position with different size, shape, camera view points, and lighting conditions, we can formulate concept inference as inference over a conjunction of multiple positional EBMs. Each positional EBM takes a different view as input we minimize energy value over positions across the sum of the energies. We use the same metric used above, i.e. Mean Absolute Error, in position inference and find the error in regressing positions goes down when successively giving more images in Figure 11.

Concept Inference of Unseen Scene with Multiple Objects We also investigate the inherent compositionality that emerges from inference on a single EBM generalizing to multiple objects. Given EBMs trained on images of a single object, we test on images with multiple objects (not seen in training). In Figure 12, we plot the input RGB image and the generated energy maps over all positions in the scene. The "Two Cubes" scenes are never seen during training, but the output energy map is still make scene with the bimodality energy distribution. The generated energy map of "Two Cubes" is also close to the summation of energy maps of "Cube 1" and "Cube 2" which shows the EBM is able to infer concepts, such as position, on unseen scene with multiple objects.

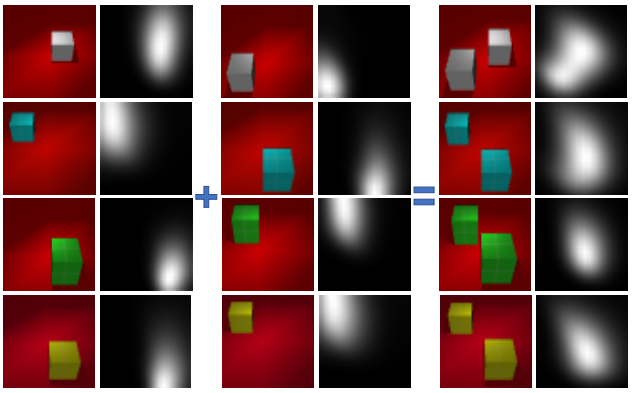


Figure 12: Concept inference of multiple objects with EBM trained on single cubes and tested on two cubes. The color image is the input and in grayscale is the output energy map over all positions. The energy map of two cubes correctly shows the bimodality which is close to the summation of the front two energy maps.

4 Related Work

Our work draws on results in energy based models - see (LeCun et al., 2006) for a comprehensive review. A number of methods have been used for inference and sampling in EBMs, from Gibbs Sampling (Hinton et al., 2006), Langevin Dynamics (Du & Mordatch, 2019), Path Integral methods (Du et al., 2019) and learned samplers (Kim & Bengio, 2016). In this work, we show that MCMC sampling on EBMs through Langevin Dynamics can be used to compositionally generate realistic images.

Compositionality has been incorporated in representation learning (see (Andreas, 2019) for a summary) and in generative modeling. One approach to compositionality has focused on learning disentangled factors of variation (Higgins et al., 2017; Kulkarni et al., 2015; Vedantam et al., 2018). Such an approach allows the combinatorial specification of outputs, but does not allow the addition of new factors. A different approach to compositionality includes learning various different pixel/segmentation masks for each concept (Greff et al., 2019; Gregor et al., 2015). However such a factorization may have difficulty capturing the global structure of an image, and in many cases different concepts can not be explicitly factored as attention masks.

In contrast, our approach towards compositionality focuses on composing separate learned probability distribution of concepts. Such an approach allows viewing factors of variation as constraints (Mnih & Hinton, 2005). (Hinton, 1999) shows product of EBMs allows for conjunction of different concepts. In our work we show additional logical compositions and corresponding performance on realistic datasets.

Our work is motivated by the goal of continual lifelong learning - see (Parisi et al., 2018) for a thorough review. Many methods are focused on how to overcome catash-

tophic forgetting (Kirkpatrick et al., 2017; Li & Hoiem, 2017), but do not support dynamically growing capacity. Progressive growing of the models (Rusu et al., 2016) has been considered, but is implemented at the level of the model architecture, whereas our method is agnostic to the models. Meta and few-shot learning (Reed et al., 2017; Bartunov & Vetrov, 2018) is another approach, but focuses on learning to model images rather than factors.

5 Conclusion

In this paper, we demonstrate the potential of EBMs for both compositional generation and inference. We show that EBMs support composition on both the factor and object level, unifying different perspectives of compositionality and can recursively combine with each other. We further showcase how this composition can be applied to both continually learn and compositionally infer underlying concepts. We hope our results inspire future work in this direction.

6 Acknowledgement

We should like to thank Jiayuan Mao for reading and providing feedback on the paper and both Josh Tenenbaum and Jiayuan Mao for helpful feedback on the paper.

References

Andreas, J. Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*, 2019.

Bartunov, S. and Vetrov, D. Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 670–678, 2018.

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

Du, Y., Lin, T., and Mordatch, I. Model based planning with energy based models. *CoRL*, 2019.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Fodor, J. A. and Lepore, E. *The compositionality papers*. Oxford University Press, 2002.

Greff, K., Kaufmann, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. Beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Higgins, I., Sonnerat, N., Matthey, L., Pal, A., Burgess, C. P., Bosnjak, M., Shanahan, M., Botvinick, M., Hassabis, D., and Lerchner, A. Scan: Learning hierarchical compositional visual concepts. *ICLR*, 2018.

Hinton, G. E. Products of experts. *International Conference on Artificial Neural Networks*, 1999.

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

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7): 1527–1554, 2006.

Kim, T. and Bengio, Y. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. Deep convolutional inverse graphics network. In *NIPS*, 2015.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

LeCun, Y., Chopra, S., and Hadsell, R. A tutorial on energy-based learning. 2006.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Mnih, A. and Hinton, G. Learning nonlinear constraints with contrastive backpropagation. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 1302–1307. IEEE, 2005.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *CoRR*, abs/1802.07569, 2018. URL http://arxiv.org/abs/1802.07569.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Reed, S., Chen, Y., Paine, T., Oord, A. v. d., Eslami, S., Rezende, D., Vinyals, O., and de Freitas, N. Few-shot autoregressive density estimation: Towards learning to learn distributions. *arXiv preprint arXiv:1710.10304*, 2017.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

van Steenkiste, S., Kurach, K., and Gelly, S. A case for object compositionality in deep generative models of images. *arXiv preprint arXiv:1810.10340*, 2018.

Vedantam, R., Fischer, I., Huang, J., and Murphy, K. Generative models of visually grounded imagination. In *ICLR*, 2018.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011.

A Appendix

A.1 Approximating the Magnitude of Partition Function

We estimate the magnitude of the partition function of an EBM by evaluating the energy it assigns to all data points it is trained on, and plotting the resultant histogram of energies. Figure 13 shows that the EBMs we train have similar histograms due to a combination of L2 normalization and spectral normalization. Each EBM we evaluate have different architectures but still have similar histograms.

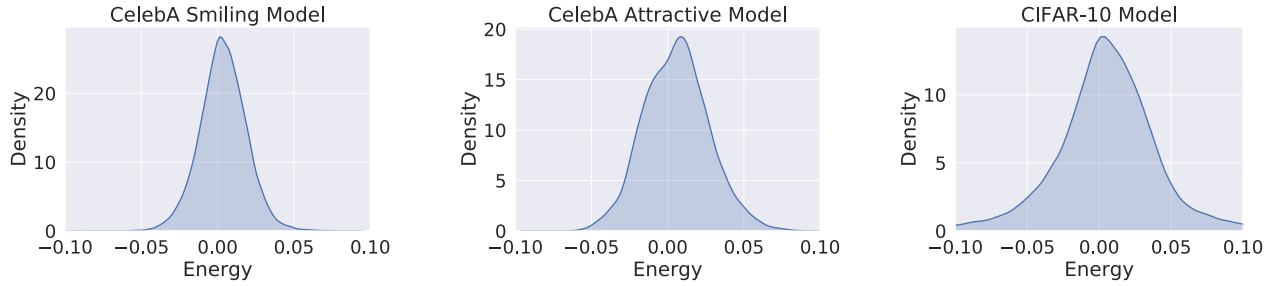


Figure 13: Energy histogram of model trained on CelebA smiling (left), CelebA attractive (middle) and pretrained CIFAR-10 model from (Du & Mordatch, 2019) (right). Each EBM we evaluate have different architectures but still have similar histograms.

Specifically, in Figure 13, we compare the energy histogram of a CelebA model trained on either smiling or attractive histograms as well as the CIFAR-10 model from (Du & Mordatch, 2019). We find that all energy histograms are similar, exhibiting minimum and maximum energies between -0.01 and 0.01. This is true even for the CIFAR-10 model which uses a significantly different dataset and architecture. Given EBMs exhibit similar partition functions at the same temperature, we obtain equation 7 and 11 in the paper.

A.2 Additional Compositionality Results

We present the composition of old, male, smiling, and non-wavy hair trained on CelebA in Figure 14.

Figure 14: Generated images from the conjunction of an EBM trained on old, male, smiling and non-wavy hair. Our model is able to generate images that are unlikely to be found in the training dataset.

We also consider combining EBMs from different domains together in Figure 15. We combine a conditional EBM trained on the attribute smiling on CelebA with an EBM trained on Mujoco scenes on different different plane colors. EBMs are trained on separate datasets with separate architectures, but are still able to make somewhat successfull combinations.

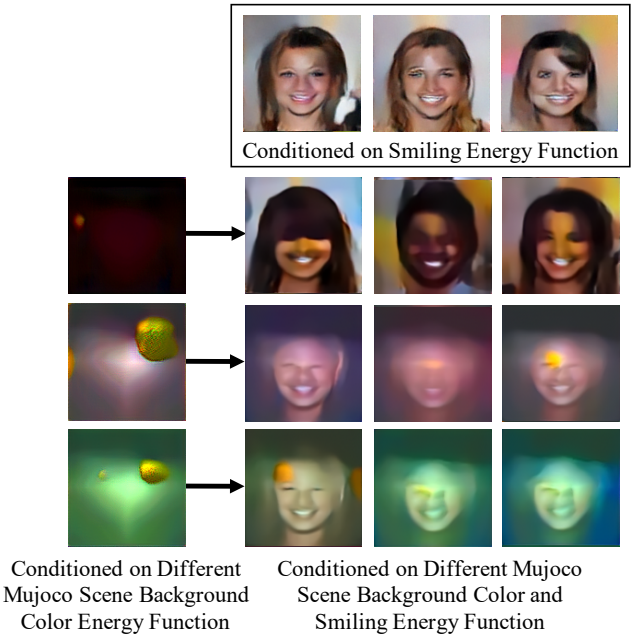
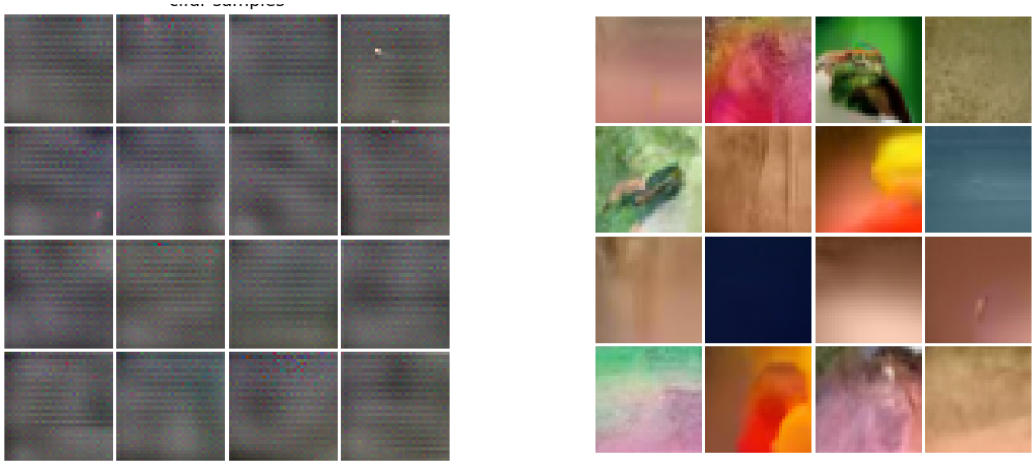


Figure 15: Generated images from EBMs trained on different domains. One EBM is conditioned on the attribute of smiling from the CelebA dataset, while the other EBM is conditioned on the color of the plane from a Mujoco Scenes dataset. our models are able to generalize to unseen combinations in training.

A.3 Discussion on Other Generative Models

To sample from the conjunction/disjunction/negation of seperate probability distributions, MCMC must be run. Other generative models, such as autoregressive models, can also support MCMC, but we find that in practice other generative models do not sample well under gradient based MCMC.



(a) Samples Generated from Langevin Sampling on PixelCNN++ model from (Salimans et al., 2017)

(b) Samples Generated from Autoregressive Sampling on PixelCNN++ model from (Salimans et al., 2017)

Figure 16: Comparison on samples generated from different sampling scenes on PixelCNN++ model from (Salimans et al., 2017). We note that Langevin sampling, while not making realistic samples, generate **higher** likelihood samples than those from autoregressive sampling

We considered Langevin based sampling on the pretrained CIFAR-10 unconditional PixelCNN++ model (Salimans et al., 2017) in Figure 16. While both sampling schemes generate images with similar likelihoods (with Langevin sampling

creating higher likelihood samples), we find images generated from Langevin sampling are significantly poorer than those generated from autoregressive sampling. We believe that when using MCMC sampling on generative models, it best to use EBMs since they are trained with MCMC inference, while other models are not trained in such a manner, and may have modes easily found through sampling that are not realistic as noted by (Nalisnick et al., 2018).

A.4 Models

3x3 conv2d, 64
ResBlock down 64
ResBlock down 128
ResBlock down 128
ResBlock down 256
Global Mean Pooling
Dense $\rightarrow$ 1

(a) EBM Model Architecture used on the Mujoco Scenes Dataset

Dense $\rightarrow$ 4096
Reshape $\rightarrow$ 256x4x4
ResBlock up 256
ResBlock up 128
ResBlock up 64
ResBlock up 64
3x3 conv2d, 3

(b) Baseline Model for Joint Generation (section 3.4)

3x3 conv2d, 64
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock down 1024
ResBlock 1024
Global Sum Pooling
dense $\rightarrow$ 1

(c) EBM Model Architecture used on the CelebA Dataset

We detail the EBM architectures used for the Mujoco Scenes images in Figure 17a and for the Celeba 128x128 images in Figure 17c. The baseline model used in comparison for section 3.4 is in Figure 17b.

A.5 Training Details/Hyperparameters/Source Code

We include an anonymous zip to code used in our experiments in the supplement.

Models trained on Mujoco Scenes and CelebA datasets use the Adam optimizer with a learning rate 3e-4 with first order moment 0.0 and second order moment 0.999. A batch size of 128 to train models, with a replay buffer of size 50000, and a 5% replacement rate. Spectral normalization is applied across models with a step size of 100 for each Langevin dynamics step. Sixty steps of Langevin sampling per training iteration for CelebA dataset and eighty steps of Langevin sampling per training iteration for the Mujoco Scenes dataset. We use the Swish activation to train our models (as noted in (Du & Mordatch, 2019)), and find that it greatly stabilizes and speed up training of models.