# Concept Learning with Energy-Based Models

**Igor Mordatch**
OpenAI
San Francisco, CA
mordatch@openai.com

## Abstract

Many hallmarks of human intelligence, such as generalizing from limited experience, abstract reasoning and planning, analogical reasoning, creative problem solving, and capacity for language require the ability to consolidate experience into *concepts*, which act as basic building blocks of understanding and reasoning. We present a framework that defines a concept by an energy function over events in the environment, as well as an attention mask over entities participating in the event. Given few demonstration events, our method uses inference-time optimization procedure to generate events involving similar concepts or identify entities involved in the concept. We evaluate our framework on learning visual, quantitative, relational, temporal concepts from demonstration events in an unsupervised manner. Our approach is able to successfully generate and identify concepts in a few-shot setting and resulting learned concepts can be reused across environments. Example videos of our results are available at **sites.google.com/site/energyconceptmodels**

## 1 Introduction

Many hallmarks of human intelligence, such as generalizing from limited experience, abstract reasoning and planning, analogical reasoning, creative problem solving, and capacity for language and explanation are still lacking in the artificial intelligent agents. We, as others [25, 22, 21] believe what enables these abilities is the capacity to consolidate experience into *concepts*, which act as basic building blocks of understanding and reasoning.

Examples of concepts include visual (*"red"* or *"square"*), spatial (*"inside"*, *"on top of"*), temporal (*"slow"*, *"after"*), social (*"aggressive"*, *"helpful"*) among many others [22]. These concepts can be either identified or generated - one can not only find a square in the scene, but also create a square, either physical or imaginary. Importantly, humans also have a largely unique ability to combine concepts compositionally (*"red square"*) and recursively (*"move inside moving square"*) - abilities reflected in the human language. This allows expressing an exponentially large number of concepts, and acquisition of new concepts in terms of others. We believe the operations of identification, generation, composition over concepts are the tools with which intelligent agents can understand and communicate existing experiences and reason about new ones.

Crucially, these operations must be performed on the fly throughout the agent's execution, rather than merely being a static product of an offline training process. Execution-time optimization, as in recent work on meta-learning [6] plays a key role in this. We pose the problem of parsing experiences into an arrangement of concepts as well as the problems of identifying and generating concepts as optimizations performed during execution lifetime of the agent. The meta-level training is performed by taking into account such processes in the inner level.

Specifically, a concept in our work is defined by an energy function taking as input an event configuration (represented as trajectories of entities in the current work), as well as an attention mask over entities in the event. Zero-energy event and attention configurations imply that event entities selected by the attention mask satisfy the concept. Compositions of concepts can then be created by simply summing energies of constituent concepts. Given a particular event, optimization can be used to identify entities belonging to a concept by solving for attention mask that leads to zero-energy configuration. Similarly, an example of a concept can be generated by optimizing for a zero-energy event configuration. See Figure 1 for examples of these two processes.

The energy function defines a family of concepts, from which a particular concept is selected with a specific concept code. Encoding of event and attention configurations can be achieved by execution-time optimization over concept codes. Once an event is encoded, the resulting concept code structure can be used to re-enact the event under different initial configurations (task of imitation learning), recognize similar events, or concisely communicate the nature of the event. We believe there is a strong link between concept codes and language, but leave it unexplored in this work.

At the meta level, the energy function is the only entity that needs to be learned. This is different from generative model or inverse reinforcement learning approaches, which typically also learn an explicit generator/policy function, whereas we define it implicitly via optimization. Our advantage as that the learned energy function can be reused in other domains, for example using a robot platform to re-enact concepts in the physical world. Such transfer is not possible with an explicit generation/policy function, as it is domain-specific.

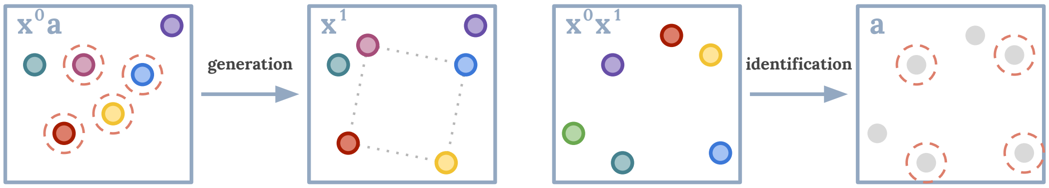

Figure 1: Examples of generation and identification processes for a *"square"* concept. a) Given initial state $\mathbf{x}^0$ and attention mask $\mathbf{a}$, square consisting of entities in $\mathbf{a}$ is formed via optimization over $\mathbf{x}^1$. b) Given states $\mathbf{x}$, entities comprising a square are found by optimization over attention mask $\mathbf{a}$.

## 2 Related Work

We draw upon several areas for inspiration in our work, including energy-based models, concept learning, inverse reinforcement learning and meta-learning.

Energy-based modelling approaches have a long history in machine learning, commonly used for density modeling [4, 16, 26, 9]. These approaches typically aim to learn a function that assigns low energy values to inputs in the data distribution and high energy values to other inputs. The resulting models can then be used to either discriminate whether or not a query input comes from the data distribution, or to generate new samples from the data distribution. One common choice for sampling procedure is Markov Chain Monte Carlo (MCMC), however it suffers from slow mixing and often requires many iterative steps to generate samples [26]. Another choice is to train a separate network to generate samples [18]. Generative Adversarial Networks [11] can be thought of as instances of this approach [7]. The key difficulty in training energy-based models lies in estimating their partition function, with many approaches relying on the sampling procedure to estimate it or further approximations [16]. Our approach avoids both the slow mixing of MCMC and the need to train a separate network. We use sampling procedure based on gradient of the energy (which mixes much faster than gradient-free MCMC), while training the energy function to have a gradient field that produces good samples.

The problem of learning and reasoning over concepts or other abstract representations has long been of interest in machine learning (see [21, 3] for review). Approaches based on Bayesian reasoning have notably been applied for numerical concepts [32]. A recent framework of [15] focuses on visual concepts such as color and shape, where concepts are defined in terms of distributions over latent variables produced by a variational autoencoder. Instead of focusing solely on visual concepts from pixel input, our work explores learning of concepts that involve complex interaction of multiple entities.

Our method aims to learn concepts from demonstration events in the environment. A similar problem is tackled by inverse reinforcement learning (IRL) approaches, which aim to infer an underlying cost function that gives rise to demonstrated events. Our method's concept energy functions are analogous to the cost or negative of value functions recovered by IRL approaches. Under this view, multiple concepts can easily be composed simply by summing their energy functions. Concepts are then enacted in our approach via energy minimization, mirroring the application a forward reinforcement learning step in IRL methods. Max entropy [36] is a common IRL formulation, and our method closely resembles recent instantiations of it [8, 5].

Our method relies on performing inference-time optimization processes for concept generation and identification, as well as for determining which concepts are involved in an event. The training is performed by taking behavior of these inner optimization processes into account, similar to the meta-learning formulations of [6]. Relatedly, iterative processes have been explored in the context of control [30, 29, 34, 14, 28] and image generation [13].

## 3 Energy-Based Concept Models

Concepts operate over events, which in this work is a trajectory of $T$ states $\mathbf{x} = \left[\mathbf{x}^0, ..., \mathbf{x}^T\right]$. Each state contains a collection of $N$ entities $\mathbf{x}^t = [\mathbf{x}_0, ..., \mathbf{x}_N]$ and each entity $\mathbf{x}_i^t$ can contain information such as position and color of the entity. Considering entire trajectories and entities allows us to model temporal or relational concepts, unlike work that focuses on visual concepts [15]. Attention over entities in the event is specified by a mask $\mathbf{a} \in \mathbb{R}^N$ over each of the entities.

Existence of a particular concept is given by energy function $E(\mathbf{x}, \mathbf{a}, \mathbf{w}) \in \mathbb{R}^+$, where parameter vector $\mathbf{w}$ specifies a particular concept from a family. The interpretation of $\mathbf{w}$ is similar to that of a code in an autoencoder. $E(\mathbf{x}, \mathbf{a}, \mathbf{w}) = 0$ when state trajectory $\mathbf{x}$ under attention mask $\mathbf{a}$ over entities satisfies the concept $\mathbf{w}$. Otherwise, $E(\mathbf{x}, \mathbf{a}, \mathbf{w}) > 0$. The conditional probabilities of a particular event configuration belonging to a concept and a particular attention mask identifying a concept are given by the Boltzmann distributions:

$$p(\mathbf{x}|\mathbf{a}, \mathbf{w}) \propto \exp\left\{-E(\mathbf{x}, \mathbf{a}, \mathbf{w})\right\} \qquad p(\mathbf{a}|\mathbf{x}, \mathbf{w}) \propto \exp\left\{-E(\mathbf{x}, \mathbf{a}, \mathbf{w})\right\} \tag{1}$$

Given concept code $\mathbf{w}$, the energy function can be used for both generation and identification of a concept implicitly via optimization (see Figure 1):

$$\mathbf{x}(\mathbf{a}) = \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{a}, \mathbf{w}) \qquad \mathbf{a}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{a}} E(\mathbf{x}, \mathbf{a}, \mathbf{w}) \tag{2}$$

Samples from distributions in (1) can be generated via stochastic gradient Langevin dynamics, effectively performing stochastic minimization in (2):

$$\begin{aligned} \tilde{\mathbf{x}} &\sim \pi_x\left(\,\cdot\mid\mathbf{a}, \mathbf{w}\right) = \mathbf{x}^K,\ \ \mathbf{x}^k = \mathbf{x}^{k-1} + \frac{\alpha}{2}\nabla_{\mathbf{x}} E(\mathbf{x}, \mathbf{a}, \mathbf{w}) + \omega^k \\ \tilde{\mathbf{a}} &\sim \pi_a\left(\,\cdot\mid\mathbf{x}, \mathbf{w}\right) = \mathbf{a}^K,\ \ \mathbf{a}^k = \mathbf{a}^{k-1} + \frac{\alpha}{2}\nabla_{\mathbf{a}} E(\mathbf{x}, \mathbf{a}, \mathbf{w}) + \omega^k,\quad \omega^k \sim \mathcal{N}(0, \alpha) \end{aligned} \tag{3}$$

This stochastic optimization procedure is performed during execution time of the algorithm and is reminiscent of the Monte Carlo sampling procedures in prior work on energy-based models [16, 26, 9]. The procedure differs from approaches that use explicit generator functions [4, 20, 18] or explicit attention mechanisms, such as dot product attention [24].

It is shown in [35] that $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{a}}$ will approach samples from posterior distributions $p$ as $K \mapsto \infty$ and $\alpha \mapsto 0$. However in practice it is only possible to execute the dynamics for a finite number of steps (we use $K = 10$ in all our experiments). This truncated procedure results in samples drawn from a biased distribution, which we call $\pi$ and which may not be equal to $p$. Similar issues of slow mixing are also present in prior work, which typically uses non-differentiable sampling procedures. In our case, the sampling procedure in equation (3) can be differentiated and can be trained to produce samples close to true distribution $p$.

There are many possible choices for the energy function as long as it is non-negative. The specific form we use in this work is based on relation network architecture [27] for its ability to easily capture interactions between pairs of entities

$$E_\theta(\mathbf{x}, \mathbf{a}, \mathbf{w}) = f_\theta(\sum_{t,i,j} \sigma(\mathbf{a}_i)\sigma(\mathbf{a}_j) \cdot g_\theta(\mathbf{x}_i^t, \mathbf{x}_j^t,\ \mathbf{w}),\ \mathbf{w})^2 \tag{4}$$

Where $f$ and $g$ are multi-layer neural networks that each take concept code as part of their input. $\sigma$ is the sigmoid function and is used to gate the entity pairs by their attention masks.

# 4 Learning Concepts from Events

To learn concepts from experience grounded in events, we pose a few-shot prediction task. Given a few demonstration examples $X^{\text{demo}}$ containing tuples $(\mathbf{x}, \mathbf{a})$ and initial state $\mathbf{x}^0$ for a new event in $X^{\text{train}}$, the task is to predict attention $\mathbf{a}$ and the future state trajectory $\mathbf{x}^{1:T}$ of the new event. The new event may contain a different configuration or number of entities, so it is not possible to directly transfer attention mask, for instance. To simplify notation, we consider prediction of only one future state $\mathbf{x}^1$, although predicting more states is straightforward. The procedure is depicted in Figure 2.

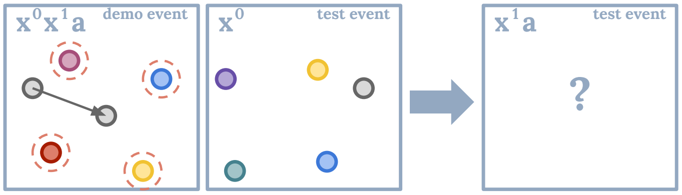

Figure 2: Example of a few-shot prediction task we use to learn concept energy functions.

We follow the maximum entropy inverse reinforcement learning formulation [36] and assume demonstrations are samples from the distributions given by the energy function $E$. Given an inferred concept code $\mathbf{w}$ (details discussed below), finding energy function parameters $\theta$ is posed as as maximum likelihood estimation problem over future state and attention given initial state. The resulting loss for a particular dataset $X$ is

$$\mathcal{L}_p^{\text{ML}}(X, \mathbf{w}) = \mathbb{E}_{(\mathbf{x},\mathbf{a})\sim X}\left[-\log p\left(\mathbf{x}^1, \mathbf{a} \mid \mathbf{x}^0, \mathbf{w}\right)\right] \tag{5}$$

Where the joint probability can be decomposed in terms of probabilities in (1) as

$$\log p\left(\mathbf{x}^1, \mathbf{a} \mid \mathbf{x}^0, \mathbf{w}\right) = \log p\left(\mathbf{x}^1 \mid \mathbf{a}, \mathbf{w}_x\right) + \log p\left(\mathbf{a} \mid \mathbf{x}^0, \mathbf{w}_a\right), \quad \mathbf{w} = [\mathbf{w}_x, \mathbf{w}_a] \tag{6}$$

We use two concept codes, $\mathbf{w}_x$ and $\mathbf{w}_a$ to specify the joint probability. The interpretation is that $\mathbf{w}_x$ specifies the concept of the action that happens in the event (i.e. *"be in center of"*) while $\mathbf{w}_a$ specifies the argument the action happens over (i.e. *"square"*). This is a concept structure or syntax that describes the event. The concept codes are interchangeable and same concept code can be used either as action or as an argument because the energy function defining the concept can either be used for generation or identification. This importantly allows concepts to be understood from their usage under multiple contexts.

Conditioned on the two codes concatenated as $\mathbf{w}$, the two log-likelihood terms in (6) can be approximated as (see Appendix for the derivation)

$$\begin{aligned} \log p\left(\mathbf{x}^1 \mid \mathbf{a}, \mathbf{w}_x\right) &\approx -\left[E(\mathbf{x}^1, \mathbf{a}, \mathbf{w}_x) - E(\tilde{\mathbf{x}}, \mathbf{a}, \mathbf{w}_x)\right]_+ \qquad &\tilde{\mathbf{x}} \sim \pi_x\left(\,\cdot \mid \mathbf{a}, \mathbf{w}_x\right) \\ \log p\left(\mathbf{a} \mid \mathbf{x}^0, \mathbf{w}_a\right) &\approx -\left[E(\mathbf{x}^0, \mathbf{a}, \mathbf{w}_a) - E(\mathbf{x}^0, \tilde{\mathbf{a}}, \mathbf{w}_a)\right]_+ \qquad &\tilde{\mathbf{a}} \sim \pi_a\left(\,\cdot \mid \mathbf{x}^0, \mathbf{w}_a\right) \end{aligned} \tag{7}$$

Where $[\cdot]_+ = \log(1 + \exp(\cdot))$ is the softplus operator. This form is similar to contrastive divergence [16] and structured SVM forms [2] and is a special case of guided cost learning formulation [8]. The approximation comes from sample-based estimates of the partition functions for $p(\mathbf{x})$ and $p(\mathbf{a})$.

The above equations make use of truncated and biased gradient-based sampling distributions $\pi_x$ and $\pi_a$ in (3) to estimate the respective partition functions. Following [8], the approximation error in these estimates is minimal when KL divergence between biased distribution $\pi$ and true distribution $\exp\{-E\}/Z$ is minimized:

$$\begin{aligned} \mathcal{L}_\pi^{\text{KL}}(X, \mathbf{w}) &= \text{KL}\left(\pi_x \| \, p_x\right) + \text{KL}\left(\pi_a \| \, p_a\right) \\ &= \mathbb{E}_{(\mathbf{x},\mathbf{a})\sim X}\left[E(\tilde{\mathbf{x}}, \mathbf{a}, \mathbf{w}_x) + E(\mathbf{x}^0, \tilde{\mathbf{a}}, \mathbf{w}_a)\right] + \text{H}\left[\pi_x\right] + \text{H}\left[\pi_a\right] \\ &\tilde{\mathbf{x}} \sim \pi_x\left(\,\cdot \mid \mathbf{a}, \mathbf{w}_x\right), \;\; \tilde{\mathbf{a}} \sim \pi_a\left(\,\cdot \mid \mathbf{x}^0, \mathbf{w}_a\right) \end{aligned}$$

The above equation intuitively encourages sampling distributions $\pi$ to generate samples from low-energy regions.

**Execution-Time Inference of Concepts** Given a set of example events $X$, the concept codes can be inferred at execution-time via finding codes $\mathbf{w}$ that minimize $\mathcal{L}^{\text{ML}}$ and $\mathcal{L}^{\text{KL}}$. Similar to [12], in this work we only consider positive examples when adapting $\mathbf{w}$ and ignore the effect that changing $\mathbf{w}$ has on the sampling distribution $\pi$. The result is simply minimizing the energy functions wrt $\mathbf{w}$ over the concept example events

$$\mathbf{w}^*_\theta(X) = \operatorname*{argmin}_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},\mathbf{a})\sim X}\left[E_\theta(\mathbf{x}^1,\mathbf{a},\mathbf{w}_x) + E_\theta(\mathbf{x}^0,\mathbf{a},\mathbf{w}_a)\right] \tag{8}$$

This minimization is similar to execution-time parameter adaptation and the inner update of meta-learning approaches [6]. We perform the optimization with stochastic gradient updates similar to equation (3). This approach again implicitly infers codes at execution time via meta-learning using only the energy model as opposed to incorporating additional explicit inference networks.

**Meta-Level Parameter Optimization** We seek probability density functions $p$ that maximize the likelihood of training data $X$ via $\mathcal{L}^{\text{ML}}_p$ and simultaneously we seek sampling distributions $\pi$ that generate samples from $p$ via $\mathcal{L}^{\text{KL}}_\pi$. In inverse reinforcement learning setting of [8] and [10], these two objectives correspond to cost and policy function optimization are treated as separate alternating optimization problems because they operate over two different functions. However, in our case both $p$ and $\pi$ are implicitly a functions of the energy model and its parameters $\theta$, a dependence which we denote as $p(\theta)$ and $\pi(\theta)$. Consequently we can pose the problem as a single joint optimization

$$\min_\theta \mathcal{L}^{\text{ML}}_{p(\theta)}(X^{\text{train}}, \mathbf{w}^*_\theta(X^{\text{demo}})) + \mathcal{L}^{\text{KL}}_{\pi(\theta)}(X^{\text{train}}, \mathbf{w}^*_\theta(X^{\text{demo}})) \tag{9}$$

We solve the above optimization problem via end-to-end backpropagation, differentiation through gradient-based sampling procedures. See Figure 3 for an overview of our procedure and appendix for a detailed algorithm description.

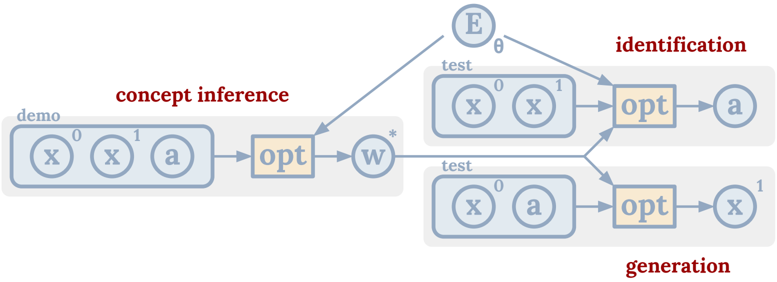

Figure 3: Execution-time inference in our method and the resulting optimization problems.

## 5 Experiments

The main purpose of our experiments is to investigate 1) whether our single model is able to learn understanding of wide variety of concepts under multiple contexts, 2) the utility of iterative optimization-based inference processes, and 3) ability to reuse learned concepts on different environments and actuation platforms.

### 5.1 Evaluation Environment and Tasks

We wish to evaluate understanding of concepts under multiple contexts - generation and identification. To the best of our knowledge we are not aware of any existing datasets or environments that simultaneously test both contexts for a wide range of concepts. For example, [17] tests understanding via question answering, while [15] focuses on visual concepts. Thus to evaluate our method, we introduce a new simulated environment and tasks which extend the work in [23]. The environment is a two-dimensional scene consisting of a varying collection of entities, each processing position, color, and shape properties. We wanted environment and tasks to be simple enough to facilitate ease of analysis, yet complex enough to lead to formation of a variety of concepts. In this work we focus on position-based environment representation, but a pixel-based representation and generation would be an exciting avenue for future work.

The task in this environment is, given $N$ demonstration events that involve (we use $N = 5$) that involve identical attention and state changes under different situations, perform analogous behavior under $N$ novel test situations (by attending to analogous entities and performing analogous state changes). Such behavior is not unique and these may be multiple possible solutions. Because our energy model architecture in section 3 processes all entities independently, the number of entities can vary between events. See Appendix for the description of events we consider in our dataset and **sites.google.com/site/energyconceptmodels** for video results of our model learning on these events.

### 5.2 Understanding Concepts in Multiple Contexts

An important property of our model is ability to learn from and apply it in both generation and identification contexts. We qualitatively observe that the model performs sensible behavior in both contexts. For example, we considered events with proximity relations *"closest"* and *"farthest"* and found model able to both attend to entities that are closest or furthest to another entity, and to move an entity to be closest or furthest to another entity as shown in figure 4. There are multiple admissible solutions which can be generated, as shown by the energy heatmap overlaid. We also wish to understand several other properties of this formulation, which we discuss below.

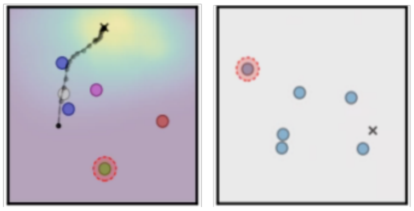

Figure 4: Outcomes of generation (left) and identification (right) for the concept of being farthest to cross-shaped entity. Path in left image is the optimization trajectory for the cross entity with the energy heatmap is overlaid.

**Transfer of learning between generation and identification contexts:** When our model trained on both contexts it shares experience between contexts. Knowing how to act out a concept should help in identifying it and vice versa - an effect observed in humans and other animals [1]. To evaluate the efficacy of transfer, we perform an experiment where we train the energy model only in identification context and test the model's performance in generation context (and conversely and second experiment where we train in generation context and test on identification context). Since it is difficult to quantitatively evaluate generative models which have multiple admissible solutions, we have collected a set of events that only involve the task of moving to an absolute location which have unique answer that allows quantitative evaluation. The results of transfer between contexts on this subset of events are reported in figure 5.

We observe that even without explicitly being trained on the appropriate context, the networks perform much better than baseline of two independently-trained networks, though not as effectively as networks that were trained on both contexts. This transfer is advantageous because in many situations demonstrations from only one type of context may be available, which our framework would still be able to integrate.

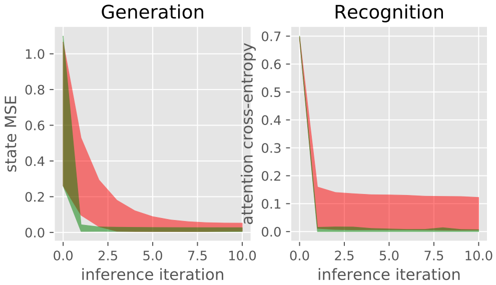

Figure 5: Accuracy of transfer between contexts for an absolute position concept. **Red** is error of the model trained only in one context (generation or identification) evaluated on the opposite context. **Green** is error of the model trained in both contexts.

**Sharing of concept codes across contexts:** Another property of our model is that codes $\mathbf{w}_x$ and $\mathbf{w}_a$ for identifying concepts are interchangeable and can be shared between generation and identification contexts. For example, either turning an entity red would or identifying all red entities in the scene would ideally use the same concept of *"red"*. We indeed observe that events which involve recognizing entities of a particular color, the codes $\mathbf{w}_a$ match the codes $\mathbf{w}_x$ for setting entities to that color (see Figure 6 for the PCA projection of these codes). We find similar correlation in the other events as well. Thus we see evidence that a concept code is reused across contexts, similarly to how words in a language are used in multiple contexts. This property presents exciting opportunities in applying our model to grounded language understanding.

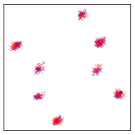

Figure 6: Projected concept codes for color events. **red** are generation codes $\mathbf{w}_x$ and **blue** are attention codes $\mathbf{w}_a$.

### 5.3 Optimization-Based Inference

Another important property of our model is that inference processes are based on iterative stochastic optimization dynamics that build up output over time and may involve non-trivial feedback corrections. In figure 7 (left), we show examples of the optimization trajectories for generation of different shape concepts. We see that the multi-step processes consist of a number of non-trivial feedback corrections to achieve the appropriate joint arrangement of entities. On the other hand, a single-step processes must achieve the arrangement through a single very precise step. While this can be adequate for simple shapes such as a line, is it problematic for more complex shapes such as the square.

In optimization trajectories of attention vectors for identification, we observe a mix of outcomes - in some cases attention vector is settled on early in the optimization process, but in other cases optimization involves non-trivial feedback corrections as shown in figure 7 (top right). In optimization of concept codes, we also observe that desired energy landscape forms only after multiple optimization iterations as from in figure 7 (bottom right).

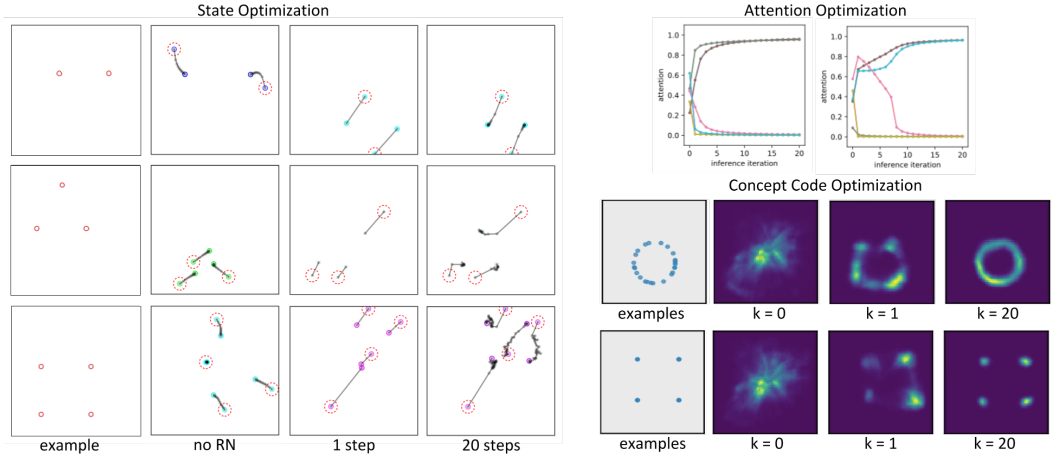

Figure 7: Trajectories of our execution-time inference processes. **Left:** Generation trajectories for shape concepts (given example shape) trained and executed with 1 optimization step and 20 steps. **Top Right:** Examples of two attention vector optimization trajectories for identification of a *line* concept. **Bottom Right:** Examples of energy landscape as $\mathbf{w}$ optimization progresses.

**Effect of a Relational Architecture** We find that using a relational architecture in our model as in equation 4 complements the optimization-based inference. We considered an alternative energy function that independently operates over individual entities rather than pair of entities, such as $E_\theta(\mathbf{x},\mathbf{a},\mathbf{w}) = f_\theta(\sum_{t,i}\sigma(\mathbf{a}_i)\cdot g_\theta(\mathbf{x}_i^t,\ \mathbf{w}),\ \mathbf{w})^2$. We find that concepts that involve a single entity, such as positioning or color concepts are able to be generated and identified without the use of relation network architecture. However, for concepts that involve coordination of multiple entities such as shape or temporal concepts we observe that not using relation network results in poor samples as shown in *"no RN"* case of figure 7.

**Effect of Training with $\mathcal{L}^{\mathrm{KL}}$ Objective** We wish to understand whether it is necessary to explicitly encourage inference process to produce good samples via $\mathcal{L}^{\mathrm{KL}}$ objective in equation (8) as it involves a computationally expensive back-propagation through the optimization procedure. Given enough steps, stochastic gradient Langevin dynamics could in theory generate samples from the energy model's distribution without this explicit objective.

However, in our experiments we observe that training without $\mathcal{L}^{\mathrm{KL}}$ objective, the gradient-based inference process of equations in (3) is not able to produce good samples in a small number of steps. Sample negative log-likelihoods are significantly higher when training without $\mathcal{L}^{\mathrm{KL}}$ objective. In figure 8 we see that while the energy network learns to discriminate between true example events (plotted in green) and sampled and random events (plotted in red and black, respectively) as a result of objective $\mathcal{L}^{\mathrm{ML}}$. However, the network is unable to produce sample events that match energy of examples. On the other hand, the network trained with both objectives $\mathcal{L}^{\mathrm{ML}}$ and $\mathcal{L}^{\mathrm{KL}}$ is able to generate samples that match the energies of examples while still being able to discriminate between true example and random events.

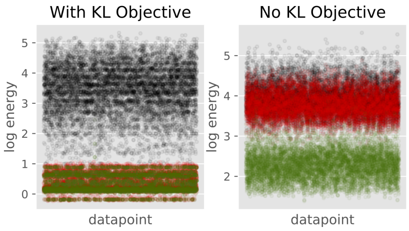

Figure 8: Energy values from models trained with and without KL objective. **Green** and positive example events, **red** are sample events generated by our run-time inference process, and **black** are random events from the initial distribution.

### 5.4 Reuse of Concepts between Environments

When a concept is learned implicitly (as opposed to generated in a feed-forward manner by a generator function or a policy), it allows the possibility to reuse the concept model under different environments and actuation platforms, provided there can be a mapping between the representations of the two environments.

To test generation of concepts in a different environment, we generated similar scenes in a three-dimensional physically-based mujoco environment [33] where the actuation mechanism is joint torques of a robotic arm rather than direct changes to environment state. To generate a concept, we used model-predictive control [31] as the optimization mechanism and used energy function learned in original environment as a cost for this optimization. Figure 9 shows results of reusing the concepts to reenact behavior of original demonstration to move into a location between two blue entities. Note that we manually defined a correspondence between representations of two environments and do not claim to tackle automatic transfer of representations - our aim is to show that learned energy functions are robust to being used under different dynamics, actuation mechanism and control algorithm.

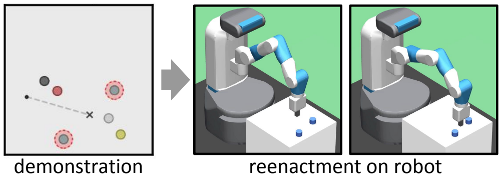

Figure 9: Energy function of reaching between blue objects learned from demonstration in 2D particle environment reused in a 3D robot simulator under a novel arrangement.

## 6 Conclusion

We believe that execution-time optimization plays a crucial role in acquisition and generalization of knowledge, planning and abstract reasoning, and communication. In this preliminary work, we proposed energy-based concept models as basic building blocks over which such optimization procedures can fruitfully operate. In the current work we used a simple concept structure, but more complex structure with multiple arguments or recursion would be interesting to investigate in the future. It would also be interesting to test compositionality of concepts, which is very suited to our model as compositions corresponds to the summation of the constituent energy functions.

# References

[1] S. Acharya and S. Shukla. Mirror neurons: enigma of the metaphysical modular brain. *Journal of natural science, biology, and medicine*, 3(2):118, 2012.

[2] D. Belanger, B. Yang, and A. McCallum. End-to-end learning for structured prediction energy networks. *arXiv preprint arXiv:1703.05667*, 2017.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[5] K. Dvijotham and E. Todorov. Inverse optimal control with linearly-solvable mdps. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 335–342, 2010.

[6] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[7] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

[8] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58, 2016.

[9] K. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.

[10] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] E. Grant, C. Finn, J. Peterson, J. Abbott, S. Levine, T. Griffiths, and T. Darrell. Concept acquisition via meta-learning: Few-shot learning from positive examples. In *Proceedings of the NIPS 2017 Workshop on "Cognitively Informed Artificial Intelligence"*, 2017.

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

[14] J. B. Hamrick, A. J. Ballard, R. Pascanu, O. Vinyals, N. Heess, and P. W. Battaglia. Metacontrol for adaptive imagination-based optimization. *arXiv preprint arXiv:1705.02670*, 2017.

[15] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Botvinick, D. Hassabis, and A. Lerchner. Scan: Learning abstract hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017.

[16] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Training*, 14(8), 2006.

[17] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

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

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–101, 2016.

[22] G. Lakoff and M. Johnson. The metaphorical structure of the human conceptual system. *Cognitive science*, 4(2):195–208, 1980.

[23] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6382–6393, 2017.

[24] C. Olah and S. Carter. Attention and augmented recurrent neural networks. *Distill*, 1(9):e1, 2016.

[25] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-braem. Basic objects in natural categories. *COGNITIVE PSYCHOLOGY*, 1976.

[26] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[27] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4974–4983, 2017.

[28] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[29] D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dulac-Arnold, D. Reichert, N. Rabinowitz, A. Barreto, et al. The predictron: End-to-end learning and planning. *arXiv preprint arXiv:1612.08810*, 2016.

[30] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, pages 2154–2162, 2016.

[31] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE, 2012.

[32] J. B. Tenenbaum. Bayesian modeling of human concept learning. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 59–68. MIT Press, 1999.

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

[34] T. Weber, S. Racanière, D. P. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.

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

[36] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

## Appendix

### Derivation of Joint Log-Likelihood Approximation

The derivation is similar to guided cost learning [8] and cost learning in linearly-solvable MDP [5] formulations. Joint negative log-likelihood of observing tuple $\left(\mathbf{x}^0, \mathbf{x}^1, \mathbf{a}\right)$ is $-\log p\left(\mathbf{x}^1, \mathbf{a} \mid \mathbf{x}^0, \mathbf{w}\right)$

$$= -\log\left(p\left(\mathbf{x}^1 \mid \mathbf{a}, \mathbf{w}_x\right) p\left(\mathbf{a} \mid \mathbf{x}^0, \mathbf{w}_a\right)\right) \tag{10}$$

$$= -\log \frac{\exp\left\{-E(\mathbf{x}^1, \mathbf{a}, \mathbf{w}_x)\right\}}{\int_{\tilde{\mathbf{x}}} \exp\left\{-E(\tilde{\mathbf{x}}, \mathbf{a}, \mathbf{w}_x)\right\}} - \log \frac{\exp\left\{-E(\mathbf{x}^0, \mathbf{a}, \mathbf{w}_a)\right\}}{\int_{\tilde{\mathbf{a}}} \exp\left\{-E(\mathbf{x}^0, \tilde{\mathbf{a}}, \mathbf{w}_a)\right\}} \tag{11}$$

Consider a more general form of the two individual terms above with non-negative function $f$

$$-\log \frac{\exp\left\{-f(\mathbf{x})\right\}}{\int_{\tilde{\mathbf{x}}} \exp\left\{-f(\tilde{\mathbf{x}})\right\}} = f(\mathbf{x}) + \log \mathbb{E}_{\tilde{\mathbf{x}} \sim q}\left[\frac{\exp\left\{-f(\tilde{\mathbf{x}})\right\}}{q(\tilde{\mathbf{x}})}\right] \tag{12}$$

The equality follows due to importance sampling under distribution $q$. There are a number of choices for sampling distribution $q$, but a choice that simplifies the above expression and we found to give stable results in practice is $q(X) = \frac{1}{2}\mathbb{I}[X = \mathbf{x}] + \frac{1}{2}\mathbb{I}[X = \tilde{\mathbf{x}}]$ where $\tilde{\mathbf{x}} \sim \pi$ and $\pi$ is a distribution that minimizes $\mathrm{KL}(\pi(X) || \ \exp\{-f(X)\}/Z)$.

In this case, sample-based approximation of equation (12) leads to

$$f(\mathbf{x}) + \log \mathbb{E}_{\tilde{\mathbf{x}}\sim q}\left[\frac{\exp\{-f(\tilde{\mathbf{x}})\}}{q(\tilde{\mathbf{x}})}\right] \approx f(\mathbf{x}) + \log\left(\exp\{-f(\mathbf{x})\} + \exp\{-f(\tilde{\mathbf{x}})\}\right) \tag{13}$$

$$= \log\left(1 + \exp\{f(\mathbf{x}) - f(\tilde{\mathbf{x}})\}\right) = [f(\mathbf{x}) - f(\tilde{\mathbf{x}})]_+ \tag{14}$$

Using the above approximation in equation (11), gives the desired result $-\log p\left(\mathbf{x}^1, \mathbf{a} \mid \mathbf{x}^0, \mathbf{w}\right)$

$$\approx \left[E(\mathbf{x}^1, \mathbf{a}, \mathbf{w}_x) - E(\tilde{\mathbf{x}}, \mathbf{a}, \mathbf{w}_x)\right]_+ + \left[E(\mathbf{x}^0, \mathbf{a}, \mathbf{w}_a) - E(\mathbf{x}^0, \tilde{\mathbf{a}}, \mathbf{w}_a)\right]_+ \tag{15}$$

$$\text{where } \tilde{\mathbf{x}} \sim \pi_x\left(\,\cdot \mid \mathbf{a}, \mathbf{w}_x\right), \quad \tilde{\mathbf{a}} \sim \pi_a\left(\,\cdot \mid \mathbf{x}^0, \mathbf{w}_a\right)$$

### Environment Event Descriptions

We created a dataset containing a variety of events in our environment in order to evaluate learning of a variety of concept in both generation and identification contexts. This dataset is not meant to be an exhaustive list of all possible concepts, but meant to represent a varied sampling of possible interesting concepts. The events involve relations described below, which are either generated or attended to:

- Changing color of entities or attending to entities of a particular color.
- Regional placement of entities in a particular spatial area - either a point, horizontal or vertical line, circle, or corners of a square.
- Placement relations of a one entity either north, south, east, west of another entity or between two entities.
- Shape relations between entities of either joining together, or forming a line, triangle, or square shapes.
- Proximity relations bring attention to the entity closest or furthest to another entity or to bring that entity to be closest or furthest to the attended entity.
- Quantity relations bring attention to any one, two, three, or more than three entities.
- Temporal relations bring an one entity to another entity only after the other entity starts moving.

See **sites.google.com/site/energyconceptmodels** for video results of our model learning on these events.

### Experiment and Training Details

In all our experiments, $f$ and $g$ in the relation network in section 3 are multi-layer neural networks with two hidden layers and 128 hidden units each. We use 5 demonstration examples in $X^{\text{demo}}$ and $X^{\text{train}}$ for each concept and use a batch size of 1024. We use $K = 10$ gradient descent steps for concept inference and sampling. We trained our experiments for 10000 timesteps and used Adam [19] optimizer with learning rate $10^{-3}$.

### Additional Results

We provide quantitative results for the transfer of learning between generation and identification contexts described in Section 5.2.

| **Condition** | **Generation Error** | **Attention Error** |
|---|---|---|
| Untrained Network | 0.621 | 0.698 |
| Trained on Generation | 0.006 | 0.061 |
| Trained on Identification | 0.016 | 0.001 |
| Trained on Both | 0.007 | 0.001 |

## Algorithm Details

For completeness, we provide the algorithm of our method below.

**Algorithm 1:** Energy-based model learning from demonstration events

```
Initialize energy model parameters $\theta$
for events $X^{\text{train}}$ and $X^{\text{demo}}$ sampled from the same concept do
    Randomly sample event $\big(\mathbf{x}^0, \mathbf{x}^1, \mathbf{a}\big)$ from $X^{\text{demo}}$
    Initialize $\mathbf{w}_x$ and $\mathbf{w}_a$ from unit Gaussian
    for sampling iteration $k = 1$ to $K$ do
        Update samples $\mathbf{w}_x$ and $\mathbf{w}_a$ via stochastic gradient Langevin dynamics step:
            $\mathbf{w}_x(\theta) \leftarrow \mathbf{w}_x(\theta) + \frac{\alpha}{2}\nabla_{\mathbf{w}} E_\theta(\mathbf{x}^0, \mathbf{a}, \mathbf{w}_x(\theta)) + \omega^k$
            $\mathbf{w}_a(\theta) \leftarrow \mathbf{w}_a(\theta) + \frac{\alpha}{2}\nabla_{\mathbf{w}} E_\theta(\mathbf{x}^1, \mathbf{a}, \mathbf{w}_a(\theta)) + \omega^k$
    end for
    Randomly sample event $\big(\mathbf{x}^0, \mathbf{x}^1, \mathbf{a}\big)$ from $X^{\text{train}}$
    Initialize $\tilde{\mathbf{x}}$ to $\mathbf{x}^0$ and initialize $\tilde{\mathbf{a}}$ from unit Gaussian
    for sampling iteration $k = 1$ to $K$ do
        Update samples $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{a}}$ via stochastic gradient Langevin dynamics step:
            $\tilde{\mathbf{x}}(\theta) \leftarrow \tilde{\mathbf{x}}(\theta) + \frac{\alpha}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}(\theta), \mathbf{a}, \mathbf{w}_x(\theta)) + \omega^k$
            $\tilde{\mathbf{a}}(\theta) \leftarrow \tilde{\mathbf{a}}(\theta) + \frac{\alpha}{2}\nabla_{\mathbf{a}} E_\theta(\mathbf{x}^0, \tilde{\mathbf{a}}(\theta), \mathbf{w}_a(\theta)) + \omega^k$
    end for
    Set $\bar{\mathbf{x}}$ be the result of applying gradient stopping operator on $\tilde{\mathbf{x}}$ (and similarly for $\bar{\mathbf{a}}$, $\bar{\mathbf{w}}$ and $\bar{E}$)
    Formulate two losses operating over $p$ and $\pi$ holding other function fixed:
    $\mathcal{L}^{\text{ML}}(\theta) = \big[E_\theta(\mathbf{x}^1, \mathbf{a}, \mathbf{w}_x(\theta)) - E_\theta(\bar{\mathbf{x}}, \mathbf{a}, \mathbf{w}_x(\theta))\big]_+ + \big[E_\theta(\mathbf{x}^0, \mathbf{a}, \mathbf{w}_a(\theta)) - E_\theta(\mathbf{x}^0, \bar{\mathbf{a}}, \mathbf{w}_a(\theta))\big]_+$
    $\mathcal{L}^{\text{KL}}(\theta) = \bar{E}(\tilde{\mathbf{x}}(\theta), \mathbf{a}, \bar{\mathbf{w}}_x) + \bar{E}(\mathbf{x}^0, \tilde{\mathbf{a}}(\theta), \bar{\mathbf{w}}_a)$
    Update $\theta$ based on gradient $\nabla_\theta(\mathcal{L}^{\text{ML}}(\theta) + \mathcal{L}^{\text{KL}}(\theta))$ via Adam optimizer
end for
```