Energy-based Models are Zero-Shot Planners for Compositional Scene Rearrangement

Nikolaos Gkanatsios†, Ayush Jain†, Zhou Xian, Yunchu Zhang, Chris Atkeson, Katerina Fragkiadaki
Carnegie Mellon University
{ngkanats,ayushj2,xianz1,yunchuz,cga}@andrew.cmu.edu, katef@cs.cmu.edu

We consider the scene arrangement task shown in Figure 1. Given a visual scene and an instruction regarding object spatial relations, the robot is tasked to rearrange the objects to their instructed configuration. Our focus is on strong generalization to longer instructions with novel predicate compositions, as well as to scene arrangements that involve novel objects and backgrounds. This is an abstract of our RSS 2023 paper.

We propose generating goal scene configurations corresponding to language instructions by minimizing a composition of energy functions over object spatial locations, where each energy function corresponds to a language concept (predicate) in the instruction. We represent each language concept as an n-ary energy function over relative object poses and other static attributes, such as object size. We train these predicate energy functions to optimize object poses starting from randomly sampled object arrangements through Langevin dynamics minimization, using a handful of examples of visual scenes paired with single predicate captions. Energy functions can be binary for two-object concepts such as *left of* and *in front of*, or multi-ary for concepts that describe arrangements for sets of objects, such as *line* or *circle*. We show that gradient descent on the sum of predicate energy functions, each one involving different subsets of objects, generates a configuration that jointly satisfies all predicates, if this configuration exists (Figure 1).

We propose a robot learning framework that harnesses minimization of compositions of energy functions to generate instruction-compatible object configurations for scene rearrangement. A neural semantic parser maps the input instruction to a set of predicates and corresponding energy functions, and an open-vocabulary detector model grounds their arguments to objects in the scene. Gradient descent on the sum of energies with respect to the objects' spatial coordinates computes the final object locations that best satisfy the set of spatial constraints expressed in the instruction. Then, we use vision-based pick-and-place policies that condition on the visual patch around the predicted pick and place locations to rearrange the objects. We call our method Scene Rearrangement via Energy Minimization (SREM).

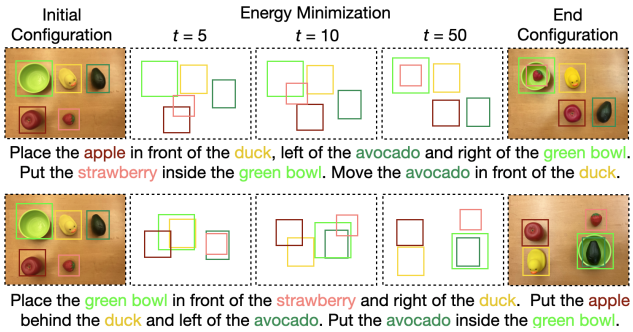

Figure 1. We represent language concepts with energy functions over object locations and sizes. Gradient descent on the sum of energy functions, one per predicate in the instruction, iteratively updates the object spatial coordinates and generates a goal scene configuration that satisfies the instruction, if one exists.

We test SREM in scene rearrangement of tabletop environments on existing simulation benchmarks, as well as on new ones we contribute that involve compositional instructions. We curate multiple train and test splits to test out-of-distribution generalization with respect to (i) longer instructions with more predicates, (ii) novel objects and (iii) novel background colors. We show SREM generalizes zero-shot to complex predicate compositions, such as *"put all red blocks in a circle in the plate"* **while trained from single predicate examples**, such as *"an apple inside the plate"* and *"a circle of blocks"*. We show SREM generalizes to real-world scene rearrangement without any fine-tuning, thanks to the object abstractions it operates on. We compare our model against state-of-the-art language-to-action policies as well as Large Language Model planners and show it dramatically outperforms both, especially for long complicated instructions. We ablate each component of our model and evaluate contributions of perception, semantic parsing, goal generation and low-level policy modules to performance. The full paper, simulation and real-world robot execution videos, as well as our code are publicly available on our website: https://ebmplanner.github.io.