Compositional Spatial Planning: An Energy-Based Approach

Anonymous Authors[1]

Abstract

Error accumulation has long been an issue in the long-horizon planning tasks, such as robot manipulation, auto-driving, etc., and have been actively studied. However, most of the current methods are either domain-specific and cannot be generalized to the out-of-domain tasks, or augment pretrained planners with traditional search algorithms to compensate the inherited error-accumulation issue of the dynamic model. This project aims to tackle such problem by providing an energy-based perspective. Specifically, we propose *Compositional Spatial Planning*(CSP) – a novel training and inference paradigm to capture multiple dependencies including modality, temporal consistency, and goal-achieving within the planning trajectory by modeling multiple energy landscapes. We plan to show our paradigm's superiority on spatial planning tasks with incremental complexities, in aspects of less accumulative errors, better performance and better generalizability, when compared to other baselines.

1. Introduction

1.1. Background

Long-horizon planning has been actively explored. Recently, with advancement of Language Reasoning Models(LRMs) demonstrating superior capability in achieving long-Chain-of-Thoughts(long-CoT) tasks, such as multi-step reasoning and text-based agent planning, people expect similar planning capacity in the high-dimensional state space.

Several methods featuring with Video Language Models(VLMs) and Vision Language Activations(VLAs) attempts to formulate the task as first projecting the data with diverse modalities into a shared latent space and then utilize the comprehensibility of language models as the backbone to process the downstream tasks. This framework works well in terms of extracting high-level understandings from high-dimensional data, such as image caption, video understanding, etc., but is inefficient in terms of restricting the understanding and planning procedures to be performed in the language space. This could also suffer from the inherited hallucination issue of the language models.

Another attempts for long-horizon planning utilize Video Models(VMs) for dynamics simulation. Due to the diffusion nature, the VMs excels at generating continuous data with temporal and spatial consistency. However, due to autoregressive "frame-to-frame" generation and the information loss due to the global window size, VMs can only generate short-horizon trajectories before the accumulated errors explode. Many hierarchical planning methods compose VMs, VLMs and VLAs into a traditional search algorithm to eliminate the issue by external efforts, where the VMs function as short-term dynamics. Although achieving certain performances in the spatial planning, such methods are still inefficient and impractical due to the considerable overheads of underexplored trajectories generation and explicit dynamics simulation.

Therefore, a natural question to ask is whether we can mitigate the error accumulation in the long-horizon planning task through directly modeling the global and local dependencies. In this work, we introduce *Compositional Spatial Planning*(CSP) as an alternative energy-based perspective for eliminating this issue. Specifically, we learn the dependencies of multi-modality, temporal consistency, and goal-achieving by modeling the corresponding energy landscapes. For inference, we compose these learned landscapes together and use the engery gradients to guide the planning in an iterative manner. To facilitate the process, we adapt various auto-encoders to convert the input from different modalities(visual frames, textual query and robot motions) into their corresponding latent representations. After we have these latents learned and sampled from our landscapes, we decode them into their original model spaces.

1.2. Objective

We aim to show the superiority of our paradigm in terms of less accumulated errors, better goal achievement, and capability for test-time scaling, compositionability and gen-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

eralizability for OOD-tasks, comapared to current baselines on spatial planning tasks with increasing complexities. The ultimate deliveries include a conference submission,and codes and checkpoints open to public.

2. Related Works (Detailed Background)

2.1. Long-Horizon Planning and Error Accumulation

abracadabra

2.2. Energy-Based Models and Applications in Planning

abracadabra

3. Method (Methodology)

In this section, we introduce our paradigm in details. After formulating our problem setting in 3.1, we first introduce the encoders and decoders we used for converting the raw data of high-dimensions and multiple modalities into diverse but feasible corresponding latents (3.2). Then we illustrate how we define and model the three kinds of dependencies w.r.t. modality, time and goal-achieving in 3.3 with emphasis. Given all the key ingredients, we illustrate our diffusion-style training paradigm in 3.4 and our iterative sampling paradigm in 3.5. An overview of our paradigm is shown in Figure 1. The complete training and sampling pseudocodes are illustrated in Algorithm 1 and Algorithm 2.

3.1. Problem Formulation

Given a textual instruction of the planning goal $\mathbf{g}$ and a initial state represented by the image frame $\mathbf{x}_0$ at time $t = 0$, we want to generate the planning trajectory represented by a consecutive and iterative sequence of motions and resulted frames: $\{\mathbf{a}_1, \mathbf{x}_1, \mathbf{a}_2, \mathbf{x}_2, \ldots, \mathbf{a}_t, \mathbf{x}_t\}$, where each action $\mathbf{a}_i$ is the motion between the consecutive pair of frames $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$, and the final state $\mathbf{x}_t$ is expected to be the goal state.

3.2. Embed Data into Latent Space

For computational efficiency, we model the energy landscapes on the latent representations of the different modalities, instead of directly modeling on the raw data of high-dimensions. For all planning tasks involved in this paper, we assume three different modalities: textual planning instruction $\mathbf{g}$, vision frame $\mathbf{x}$, and motion $\mathbf{a}$. We adapted different encoders to convert them into their latent vectors of $\gamma_{\mathbf{g}} \in \mathbb{R}^a, \gamma_{\mathbf{x}} \in \mathbb{R}^b$ and $\gamma_{\mathbf{a}} \in \mathbb{R}^c$, where a, b and c are there corresponding latent dimensions, and recover them with decoders, correspondingly. Generally, We train our encoders and decoders with the Variational Autoencoder(VAE) setting. As for architecture, we choose BERT for processing the discrete textual query(each query is processed to a single latent $\gamma_{\mathbf{g}}$), and ViT for processing continuous image frames and robot motions.

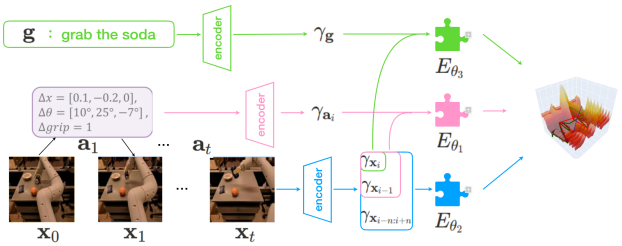

Figure 1. **Overview of our paradigm** – CSP models the latent dependencies of goal-achievement, multi-modality, and temporal consistency into three energy landscapes.

Our overall framework is illustrated in Figure 1.

3.3. Three Types of Dependencies

By intuition, we want to capture both the local and global planning dependencies through modeling different energy functions with latents representing diverse modalities, temporal information, and goal-achievement. Specifically, we define the following three energy functions to capture the corresponding dependencies:

Goal-achieving dependency. The first kind of dependency aims for capturing the global planning dependency.

$$E_{\theta_1}(\gamma_{x_0}, \gamma_{x_i}, \gamma_g) : \mathbb{R}^b \times \mathbb{R}^b \times \mathbb{R}^a \to \mathbb{R}, i = 1, \ldots, t \quad (1)$$

Intuitively, this energy function measures how well the current frame achieves the query goal, conditioned on the initial state in the latent space. Since γ_{x_0} and γ_g are fixed, we only need to model the landscape w.r.t. x_i, which we denote as y_1.

Multi-modal dependency. The second kind of dependency aims for measuring the dependency between different modalities at the same time stamp, which serves as a local planning dependency:

$$E_{\theta_2}(\gamma_{x_{i-1}}, \gamma_{x_i}, \gamma_{a_i}) : \mathbb{R}^b \times \mathbb{R}^b \times \mathbb{R}^c \to \mathbb{R}, i = 1, \ldots, t \quad (2)$$

Similarly, the $\gamma_{x_{i-1}}$ and γ_{x_i} are fixed during both training and sampling, so we only need the gradient of this energy landscape on γ_{a_i}, which we denote as y_2.

Temporal dependency. The last kind of dependency aims to measure the temporal consistency within a predefined context window of $2n$ frames, where the dependency captured is between the global and the local:

$$E_{\theta_3}(\gamma_{\mathbf{x}_{i-n:i+n}}) : \mathbb{R}^{b \times 2n} \to \mathbb{R}, i = n, \ldots, t - n \quad (3)$$

Unlike the previous two energy functions, the third energy landscape has the highest complexity since a sequence of consecutive latents $\gamma_{\mathbf{x}_{i-n:i+n}} := \{\gamma_{x_{i-n}}, \ldots, \gamma_{x_i}, \ldots, \gamma_{x_{i+n}}\}$ are subject to be tuned. However, this EBM can still be learned and inference with some trivial methods, which we will demonstrate in later sections. We denote the input latent sequence as $\mathbf{y}_3$.

Ultimately, we can simply sum up the three energy landscapes to obtain the final composed landscape:

$$E_\theta = E_{\theta_1} + E_{\theta_2} + E_{\theta_3}, \text{where } \theta = \theta_1 \cup \theta_2 \cup \theta_3 \quad (4)$$

3.4. Independent Diffusion-style Learning of Energy Landscapes

Following the traditional EBM training paradigm, we train the three EBMs in a continuous diffusion style. In our implementation, we adapt an inverse-cosine noise schedule on the three kinds of latent vectors. We iteratively add Gaussian noise to the latent, and after T iterations, the latent becomes random Gaussian noise:

$$\begin{aligned} \tilde{\gamma}^0 &= \gamma \\ \tilde{\gamma}^t &= \tilde{\gamma}^{t-1} + \epsilon, \epsilon \sim \mathcal{N}(0,1), k = 2, \ldots, T \\ \tilde{\gamma}^T &= \text{Gaussian} \end{aligned} \quad (5)$$

Then we learn the energy functions $E_{\theta_1}, E_{\theta_2}, E_{\theta_3}$ by supervising their gradients to denoise the noise-corrupted latent $y_1, y_2, \mathbf{y}_3$ at each diffusion step k and time stamp i using the L_2 objective:

$$\mathcal{L}_{\text{MSE},n}(\theta) = ||\nabla_{\mathbf{y}_n} E_\theta(\gamma_{x_0}, \gamma_g, \tilde{\gamma}^k_{\mathbf{x}_{i-n:i+n}}) - \epsilon_k||^2, n = 1, 2, 3 \quad (6)$$

3.5. Sampling with Iterative Refinement

Given the textual instruction $\mathbf{g}$ and the initial state $\mathbf{x}_0$, we can first use our textual and visual encoders to encode the latents γ_g and γ_{x_0}. Then we can use the three learned EBMs to generate the frame and motion trajectories in the latent space iteratively, and finally we use the motion and vision decoders to recover the final trajectory prediction $\{\mathbf{a}_1, \mathbf{x}_1, \ldots, \mathbf{a}_N, \mathbf{x}_N\}$.

Specifically, after we obtained the latent of the goal query γ_g and the initial state γ_{x_0}, we generate the latent trajectory in a diffusion style. Suppose we want to generate a trajectory $\{\gamma_{a_i}, \gamma_{x_i}\}_{i=1}^N$ of length N in T diffusion steps. By the end of each diffusion step $t = 1, \ldots, T$, a complete trajectory

Algorithm 1 Training EBMs

Require: train dataset $\mathcal{D}$, encoders f_x, f_a, f_g, EBMs $E_{\theta_1}, E_{\theta_2}, E_{\theta_3}$, sample frames t, noise schedules $\{\sigma_k\}$, diffusion timestep K, window size $2n$

while not converged **do**
  $(\mathbf{g}, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{a}_1, \ldots, \mathbf{a}_t) \sim \mathcal{D}$, $\epsilon \sim \mathcal{N}(0,1)$, $k \sim \{1, \ldots, K\}$
  Encode the raw data into latent space
  $\gamma_g \leftarrow f_g(\mathbf{g})$, $\gamma_{\mathbf{x}0:t} \leftarrow f_x(\mathbf{x}_{0:t})$, $\gamma_{\mathbf{a}1:t} \leftarrow f_a(\mathbf{a}_{1:t})$
  for $i = 1$ to t **do**
  Add noise to the latent inputs
    $\tilde{\gamma}_{\mathbf{x}i} \leftarrow \sqrt{1-\sigma_k^2}\,\gamma_{\mathbf{x}i} + \sigma_k \epsilon$
  Calculate the modal correlation, temporal correlaton, and goal-achievement
    $\mathcal{E}_i^{\text{mod}} \leftarrow E_{\theta_1}(\tilde{\gamma}_{\mathbf{x}i}, \gamma_{\mathbf{x}i-1}, \gamma_{\mathbf{a}i})$
    $\mathcal{E}_i^{\text{temp}} \leftarrow E_{\theta_2}(\gamma_{\mathbf{x}i-n:i+n}, \gamma_{\mathbf{a}i})$, $i+n < t$
    $\mathcal{E}_i^{\text{goal}} \leftarrow E_{\theta_3}(\gamma_{\mathbf{x}i}, \gamma_g)$
  Train with the compositional energy objective
    $\mathcal{E}_i \leftarrow \mathcal{E}_i^{\text{mod}} + \mathcal{E}_i^{\text{temp}} + \mathcal{E}_i^{\text{goal}}$
    $\Delta\theta_1 \leftarrow \nabla_{\theta_1}\left(\|\nabla_{\gamma_{\mathbf{x}i}} \mathcal{E}_i - \epsilon\|^2\right)$
    $\Delta\theta_2 \leftarrow \nabla_{\theta_2}\left(\|\nabla_{\gamma_{\mathbf{x}i-n:i+n}} \mathcal{E}_i - \epsilon\|^2\right)$
    $\Delta\theta_3 \leftarrow \nabla_{\theta_3}\left(\|\nabla_{\gamma_{\mathbf{x}i}, \gamma_g} \mathcal{E}_i - \epsilon\|^2\right)$
  end for
  Update $\theta_1, \theta_2, \theta_3$ on $\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$ using Adam optimizer
end while

$\{\gamma_{a_i}^t, \gamma_{x_i}^t\}_{i=1}^N$ with a noise degree of t is generated. And within each diffusion step t, we also denoise and refine the latents near the current state iteratively at each time stamp $i = 1, \ldots, N$ with an increasing context of observables:

First, the goal-achieving EBM E_{θ_1} denoises the current state γ_{x_i} conditioned on the goal γ_g and the initial state γ_{x_0} in T steps.

$$\gamma_{x_i}^{t+1} \leftarrow \gamma_{x_i}^t - \lambda_1 \nabla_{\gamma_{x_i}^t} E_{\theta_1}(\gamma_g, \gamma_{x_0}, \gamma_{x_i}^t) \quad (7)$$

where $\lambda_i, i = 1, 2, 3$ are gradient steps for the three EBMs. In our implementation,we choose to initialize the current frame latent with Gaussian noise: $\gamma_{x_i}^0 \sim \mathcal{N}(0, I_x)$

Secondly, the multi-modal EBM E_{θ_2} denoises the latest motion γ_{a_i} which results in the predicted current frame, conditioned on the previous and the current frames $\gamma_{x_{i-1}}, \gamma_{x_i}$. In our implementation, We incorporate E_{θ_1} together with E_{θ_2} to impose a stronger guidance on the sampling:

$$\gamma_{a_i}^{t+1} \leftarrow \gamma_{a_i}^t - \lambda_2 \nabla_{\gamma_{a_i}^t} E_{\theta_1 \cup \theta_2}(\gamma_g, \gamma_{x_0}, \gamma^t_{\mathbf{x}_{i-1:i}}) \quad (8)$$

Here we also choose to randomly initialize the initial motion latent: $\gamma_{a_i}^0 \sim \mathcal{N}(0, I_a)$.

Finally, if the currently denoised latents are enough to form a complete context window of size $2n$, the temporal-consistency EBM E_{θ_3} refines each frame latent iteratively, conditioned on other latent inside the window. Similarly,

Algorithm 2 Planning via Sampling Latents

Models: trained encoders $f_{\text{vision}}, f_{\text{goal}}, f_{\text{motion}}$; trained decoders $g_{\text{vision}}, g_{\text{motion}}$; trained EBMs $E_{\theta_1}(\gamma_{\mathbf{x}0}, \gamma_{\mathbf{x}i}, \gamma_g)$, $E_{\theta_2}(\gamma_{\mathbf{x}i-1:i}, \gamma_{\mathbf{a}i})$, $E_{\theta_3}(\gamma_{\mathbf{x}i-n:i+n})$
Input: initial frame $\mathbf{x}_0$, goal $\mathbf{g}$, frames N, gram size $2n$, sampling steps T
Initialize latent buffers $B_{\mathbf{x}}, B_{\mathbf{a}}$
Initialize compositional energy $\mathcal{E} \leftarrow \infty$
Initialize latents $\gamma_{\mathbf{x}1:N}$, $\gamma_{\mathbf{a}1:N} \sim \mathcal{N}(0, I)$
$\gamma_g \leftarrow f_{\text{goal}}(\mathbf{g})$, $\gamma_{\mathbf{x}0} \leftarrow f_{\text{vision}}(\mathbf{x}_0)$
for $t = 1$ to T **do**
  for $i = 1$ to N **do**
    first: denoise frame latent at i
    $\gamma'_{\mathbf{x}i} \leftarrow \gamma_{\mathbf{x}i} - \nabla_{\gamma_{\mathbf{x}i}} E_{\theta_1}(\gamma_{\mathbf{x}0}, \gamma_{\mathbf{x}i}, \gamma_g)$
    second: denoise motion latent at i
    $\gamma'_{\mathbf{a}i} \leftarrow \gamma_{\mathbf{a}i} - \nabla_{\gamma_{\mathbf{a}i}} E_{\theta_1+\theta_2}(\gamma_{\mathbf{x}i-1}, \gamma_{\mathbf{x}i}, \gamma_{\mathbf{a}i}, \gamma_g)$
    third: joint update of frame latent at i
    if $n \leq i \leq N - n$ **then**
      $\gamma''_{\mathbf{x}i} \leftarrow \gamma'_{\mathbf{x}i} - \nabla_{\gamma'_{\mathbf{x}i}} E_{\theta_1+\theta_2+\theta_3}(\gamma'_{\mathbf{x}i-n:i+n}, \gamma'_{\mathbf{a}i-n:i+n}, \gamma_g)$
      $\gamma''_{\mathbf{a}i} \leftarrow \gamma'_{\mathbf{a}i} - \nabla_{\gamma'_{\mathbf{a}i}} E_{\theta_1+\theta_2+\theta_3}(\gamma''_{\mathbf{x}i-n:i+n}, \gamma'_{\mathbf{a}i-n:i+n}, \gamma_g)$
      if $E_\theta(\gamma_{\mathbf{x}0:n}, \gamma''_{\mathbf{x}i-n:i+n}, \gamma_g) < E_\theta(\gamma_{\mathbf{x}0:n}, \gamma_{\mathbf{x}i-n:i+n}, \gamma_g)$ **then**
        $\gamma_{\mathbf{x}i-n:i+n} \leftarrow \gamma''_{\mathbf{x}i-n:i+n}$, $\gamma_{\mathbf{a}i-n:i+n} \leftarrow \gamma''_{\mathbf{a}i-n:i+n}$
      end if
    end if
  end for
end for
return $\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{a}_1, \ldots, \mathbf{a}_N$

we incorporate E_{θ_3} together with the other two EBMs for better sampling:

$$\gamma_{x_j}^{t+1} \leftarrow \gamma_{x_j}^{t} - \lambda_3 \nabla_{\gamma_{x_j}^t} E_{\theta_1 \cup \theta_2 \cup \theta_3}(\gamma_g, \gamma_{x_0}, \gamma_{\mathbf{x}_{i-n:i+n}}^t) \quad (9)$$

where $j \in [i-n, i+n]$.

An illustration of the decoding procedure within each diffusion time stamp is shown in Figure 2.

4. Experiments (Schedule & Milestones)

Please refer to the tentative schedule of my project in Table 1.

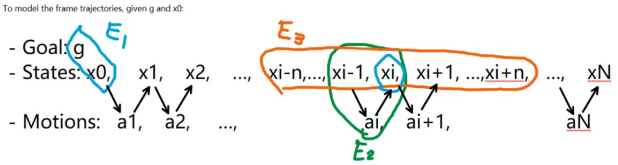

Figure 2. **Relationships of the latent and EBMs** – Different EBMs with their inputs are represented in different colors: the blue ones are for goal-achievement, the green ones are for multi-modality and the orange ones are for temporal-consistency. For inference, we incrementally expand the context observables and incorporates the EBMs for sampling, with order: $E_{\theta_1} \rightarrow E_{\theta_1 \cup \theta_2} \rightarrow E_{\theta_1 \cup \theta_2 \cup \theta_3}$.

Date	To-Do's	Event
Oct 1-31, 2025	• Revise the method • Test in Particle Environment • Test in Maze Environment • Test Sawyer Arm	Simple Trials
Nov 1 - Dec 31, 2025	• Revise the method • Test in Libero Environment • Test in Meta-World Environment • Test THOR Visual Navigation • Test in Calvin Environment	Harder Trials
Jan 1-12, 2026	• Run baselines • Prepare for presentation	RES
Jan 12–17, 2026	—	First presentation
Jan 25, 2026	• Preliminary implementation • Detailed interim report	Second Deliverable
Feb 1 - Mar 31, 2026	• Ablation Study • Paperwork	Complete Experiments
Apr 1-18, 2026	• prepare final report and website • prepare final presentation	RES
Apr 19, 2026	• Finalised tested implementation • Final report • Project web page (final version)	Final Deliverable
Apr 20–25, 2026	—	Final presentation
Apr 22, 2026	1-min video	Poster exhibition

Table 1. **FYP timeline: deliverables and events** - The majority work is expected to be done in 2025(with some tail-in works left to 2026), since my coursework in semester 2 will be quite heavy.