# A Minimal GPT-Style Model for Learning and Visualizing Latent Representations of Simple Physical Dynamics

Your Name

**Abstract**

This document presents a minimal prototype of a GPT-style Transformer model trained to predict the dynamics of a simple 1D physical system. We outline the physics, data generation, model architecture, and a method to visualize how the model's latent representations vary as the underlying physics parameter changes. This prototype serves as an illustrative tool for understanding how large language models internally represent dynamics and structure in a fixed-dimensional latent space.

## 1 Introduction

Transformers operate by mapping an input sequence into a sequence of vectors in a fixed-dimensional latent space (e.g., $\mathbb{R}^{4096}$). Although the dimensionality is fixed, the actual *latent state* depends entirely on the input context and evolves layer-by-layer.

To understand how changes in a physical system influence changes in a model's latent space, we construct a small GPT-style model that predicts the next state of a simple dynamical system. We then visualize the hidden representations using PCA or t-SNE.

## 2 Simple Physics: 1D Constant-Acceleration System

Consider a 1D particle with position $x_k$, velocity $v_k$, and constant acceleration $a$. Using a fixed time step $\Delta t$, the dynamics are:

$$v_{k+1} = v_k + a\Delta t, \tag{1}$$
$$x_{k+1} = x_k + v_k\Delta t. \tag{2}$$

For each trajectory, we randomly sample:

- initial state $(x_0, v_0)$,
- physical parameter $a \in \{-1.0, -0.5, 0.5, 1.0\}$,
- horizon length $T$ (e.g., $T = 16$).

The model receives as input the sequence:

$$s_k = [x_k, v_k, a],$$

and is trained to predict the next state $[x_{k+1}, v_{k+1}]$ at each time step.

## 3 Data Generation

A single trajectory is generated as:

$$v_{k+1} = v_k + a\Delta t, \tag{3}$$
$$x_{k+1} = x_k + v_k\Delta t, \tag{4}$$

for $k = 0, \ldots, T-1$, where $a$ is fixed for the entire trajectory.

The dataset consists of many such trajectories, each labeled by the underlying physics parameter $a$.

## 4 GPT-Style Model Architecture

Let $d_{\text{model}}$ be the Transformer hidden dimension (e.g., $d_{\text{model}} = 64$). The architecture contains:

- An input projection $W_{\text{in}} : \mathbb{R}^3 \to \mathbb{R}^{d_{\text{model}}}$,
- Learned positional embeddings $\mathbf{p}_k \in \mathbb{R}^{d_{\text{model}}}$,

- A stack of $L$ Transformer encoder layers with causal masking,
- An output head $W_{\text{out}} : \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^2$ to predict $[x_{k+1}, v_{k+1}]$.

Given an input sequence $X \in \mathbb{R}^{T\times 3}$, the model computes:

$$H_0 = W_{\text{in}} X + P, \tag{5}$$

$$H_\ell = \text{TransformerLayer}(H_{\ell-1}), \quad \ell = 1, \ldots, L, \tag{6}$$

$$\hat{Y} = W_{\text{out}} H_L, \tag{7}$$

where $P$ contains the positional embeddings.

The hidden tensor $H_L \in \mathbb{R}^{T\times d_{\text{model}}}$ is the final latent representation for all time steps.

## 5 Training Objective

We train the model with mean squared error (MSE):

$$\mathcal{L} = \frac{1}{BT} \sum_{b=1}^{B} \sum_{k=1}^{T} \left\| \hat{y}_{b,k} - y_{b,k} \right\|^2, \tag{8}$$

where $B$ is batch size and $T$ is sequence length.

## 6 Extracting Latent States

After training, we freeze the model and extract hidden states $H_L$ for many trajectories under different physical parameters $a$. Let each hidden state for time step $k$ be denoted:

$$h_k \in \mathbb{R}^{d_{\text{model}}}.$$

Collect all hidden states for all trajectories into a matrix:

$$\mathbf{H} \in \mathbb{R}^{N\times d_{\text{model}}},$$

where $N$ is the total number of collected time-step embeddings.

## 7 Latent Space Visualization

To visualize how different physics settings produce different latent representations, we apply PCA or t-SNE:

$$\mathbf{Z} = \text{PCA}(\mathbf{H}), \qquad \mathbf{Z} \in \mathbb{R}^{N\times 2}. \tag{9}$$

We then plot $\mathbf{Z}$, coloring each point by:

- the underlying physics parameter $a$, or
- the time step $k$, or
- the layer index (if collecting multi-layer states).

Empirically, one typically observes:

- clustering or distinct manifolds corresponding to different physics parameters $a$,
- temporal structure forming smooth trajectories in the latent space,
- deeper layers exhibit stronger separation and structure.

## 8 Figure Placeholders

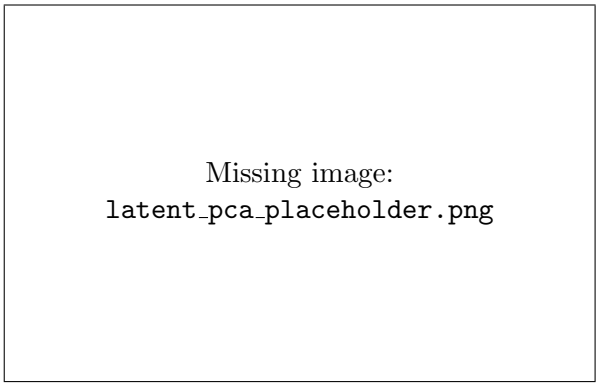


Figure 1: Example PCA visualization of hidden states colored by acceleration $a$.

## 9 Conclusion

This minimal Transformer-based experiment demonstrates how a GPT-style model embeds physical dynamics inside its fixed-dimensional latent space. By visualizing these hidden representations under different physical parameters, we gain intuition for how large language models internally encode structured, dynamic information.

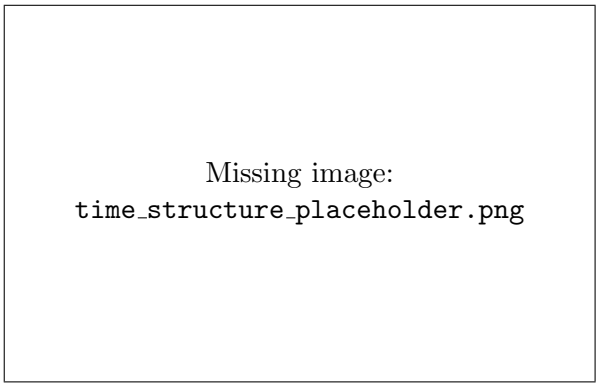


Figure 2: Temporal progression of latent states for a fixed $a$.