SBIN 1.0 - Physics-Informed Learning Pipeline

1. Model Overview

The SBIN 1.0 (Self-Learned Brachistochrone Informed Network) is a physics-informed deep learning framework implemented in PyTorch. It is designed to learn time-optimal descent trajectories (brachistochrones) under gravitational or inertial dynamics.

The model integrates classical mechanics constraints into a deep neural network by embedding the following losses:

* Euler-Lagrange Loss (L_EL): Enforces the Euler-Lagrange dynamics.
* Hamiltonian Loss (L_HH): Ensures energy conservation through dH/dt ~ 0.
* Brachistochrone Loss (L_BC): Penalizes non-optimal paths using classical variational calculus.
* Relation Loss (L_rel): Encourages consistency within learned trajectories via unsupervised KMeans clustering.

The core of the architecture is a TemporalPhysicsNet, a 1D CNN followed by two lightweight heads for estimating diagonal mass matrices and potential energies from learned coordinates.

2. Model Architecture

* Backbone: 1D CNN that processes [features + time] to extract hidden dynamics.
* Mass Estimation: A 1x1 convolution followed by a Softplus activation ensures positive diagonal mass matrices.
* Potential Estimation: A separate 1x1 convolutional head outputs scalar potential energy.
* Coordinates (theta): Generalized coordinates inferred through multi-layer 1D convolution.
* Loss Weighting: Uses adaptive log-variance parameters (log sigma^2) that dynamically balance different physics constraints during training.

3. Dataset & Features

* Input Data: Time-series features like roll, pitch, yaw, ax, ay, az.
* Format: CSV file with each row representing a snapshot in time.
* Preprocessing:
  - Missing values are imputed with the column mean.

- Optional: Gaussian noise added to input features to improve robustness.
- KMeans clustering used to construct a relation tensor across points in the dataset.

* Time is provided explicitly for causal learning of the trajectory.

4. Output & Evaluation

* Loss Convergence: The total training loss is plotted over epochs to assess convergence.
* Trajectory Plots: 2D and 3D plots of theta(t), including velocity components (vn, ve) and positions.
* Accuracy: Model success is evaluated through physical consistency (e.g., minimal dH/dt) rather than prediction error alone.
* Visual Outputs: Results are saved as PNG and assembled into final reports.

5. Applications & Flexibility

* Works with any continuous motion dataset involving spatial coordinates, forces, or accelerations.
* Can replace feature heads to adapt to new sensors or coordinate systems.
* Useful in aerospace, robotics, biomechanics, and any domain with motion governed by physics laws.