PhD Application Challenge

Instructions (Read Carefully)

Thank you for your interest in joining the research group. To ensure a good technical fit for our work in Scientific Machine Learning, all applicants must complete this short coding task.

1. **Deliverables:** You must produce two files:
   - A `.ipynb` (Jupyter Notebook) containing your code and plots.
   - A `.pdf` export of that notebook (cleanly formatted).
2. **Language:** Python with **PyTorch**.
3. **Submission:** Attach both files to your application email using the subject line format specified in the job post.

The Challenge: Reinforced Smoothing

Background

In scientific machine learning, experimental data is often noisy. A standard neural network tends to "overfit" this noise. Your task is to implement a **custom loss function** that enforces smoothness.

Task Description

1. Data Generation

- Generate a noisy dataset from the ground truth function $y = \sin(x)$ on the domain $x \in [0, 2\pi]$.
- Generate $N = 40$ training points uniformly distributed in $[0, 2\pi]$.
- Add Gaussian noise: $y_{noisy} = \sin(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.2)$.

2. The "Smoothness" Loss Function (Core Task)

Standard Mean Squared Error (MSE) loss trains neural networks to fit data points accurately, but it doesn't encourage the learned function to be smooth or physically plausible.
Your Challenge: Design a custom loss function that balances two objectives:

- Fitting the noisy training data.
- Encouraging the neural network's predictions to be "smooth".

3. Training and Results

- Implement and train your model(s) using your custom loss function.
- Demonstrate the effectiveness of your approach by visualizing the training process and results.

Evaluation Criteria

We are looking for:

- **Code Structure:** Well-organized, modular, and maintainable code.
- **Process Clarity:** Clear demonstration of your approach, methodology, and reasoning.
- **Visualization:** Clear labels, legends, and professional formatting.
- **Technical Proficiency:** Proper use of Python and PyTorch capabilities.