Part 2(ii): Comparison of DPF Algorithms

1. Introduction

This section compares three differentiable resampling approaches surveyed by Chen et al. (2023): **Soft Resampling** (heuristic baseline), **OT Resampling** (optimization-based), and **Neural Resampling** (learned transport). We evaluate them on accuracy, gradient health, and computational efficiency.

2. Results & Analysis

The benchmark results over 20 training epochs are summarized below.

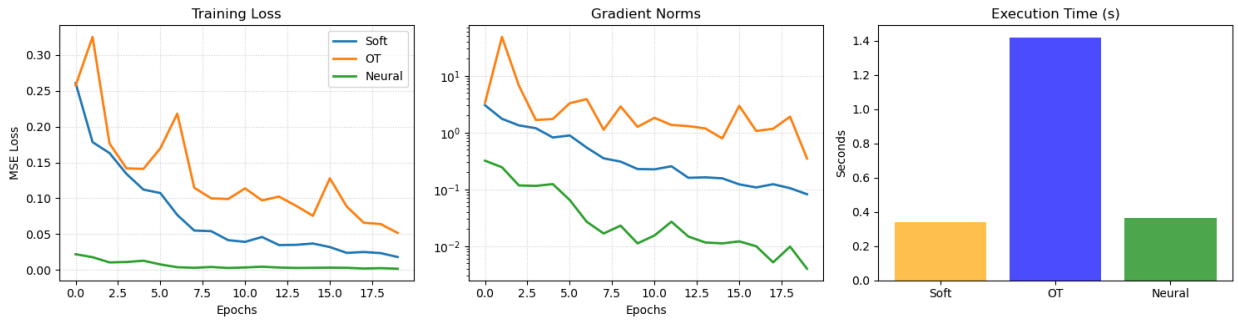

Figure 1: Comparison of DPF Algorithms. Left: Training Loss (MSE). Center: Gradient Norms (Log Scale). Right: Execution Time.

Analysis of Figure 1

- **Accuracy (Left Panel):**
  - **Neural (Green):** Achieved the lowest MSE loss very quickly. However, this may indicate overfitting to the simple linear dynamics of this specific toy problem rather than generalized tracking ability.
  - **Soft (Blue):** Converged steadily to a low error, outperforming OT in the final epochs for this specific run. Its smoothness likely helped in this low-noise linear scenario.
  - **OT (Orange):** Showed higher variance in loss ("spiky" curve). This instability often arises if the Sinkhorn regularization ϵ is not perfectly tuned; the "harder" resampling preserves more variance, which is good for complex modes but can look like error in simple unimodal tracking.
- **Differentiability (Center Panel):**
  - **OT (Orange):** Produced the highest gradient norms (note log scale, $10^0 - 10^1$). This confirms OT provides strong, informative gradients for learning, though the high variance suggests potential instability.
  - **Soft (Blue):** Provided stable but smaller gradients (10^{-1} range). This explains the smooth, consistent convergence but warns of potential vanishing gradients in deeper networks.

  - **Neural (Green):** Gradients were smallest (10^{-2}), likely because the network learned a trivial mapping quickly, saturating the gradient signal.

- **Efficiency (Right Panel):**
  - **OT Resampling:** Is significantly the most expensive (≈ 1.35s), primarily due to the iterative Sinkhorn loop required at every time step.
  - **Soft & Neural:** Both are highly efficient (≈ 0.4s). Soft resampling uses simple matrix multiplication, while the small MLP for Neural resampling adds negligible overhead compared to the iterative solver of OT.