# Differentiable and Stable Long-Range Tracking of Multiple Posterior Modes


**Ali Younis** ayounis@uci.edu

**Erik Sudderth** sudderth@uci.edu

UCI University of California, Irvine


## 1. Introduction

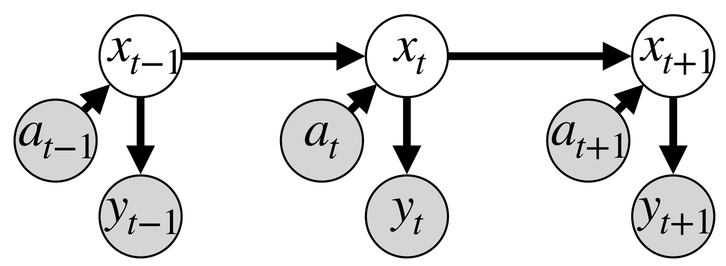
***New discriminative learnable particle filter method: Mixture Density Particle Filter***

$x_t$ = state $a_t$ = action $y_t$ = observation

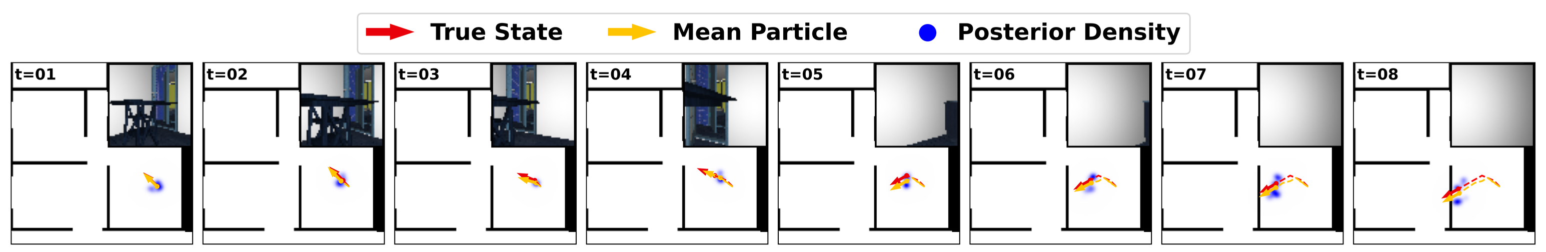
**Given**

**Actions** *(optional)*: E.g. Noisy Odometry

**Observations:** E.g. Images



**Estimate Posterior Density 3D state:** $x_t$ = (position, orientation)




## 2. Particle Filtering

**Step 0: Initialize Particle Set**

$\{x_1^{(1)}, \dots, x_1^{(N)}\}$ $\{w_1^{(1)}, \dots, w_1^{(N)}\}$

*(particles)* *(weights)*

**Step 1: Particle Proposal (Noisy Dynamics):**

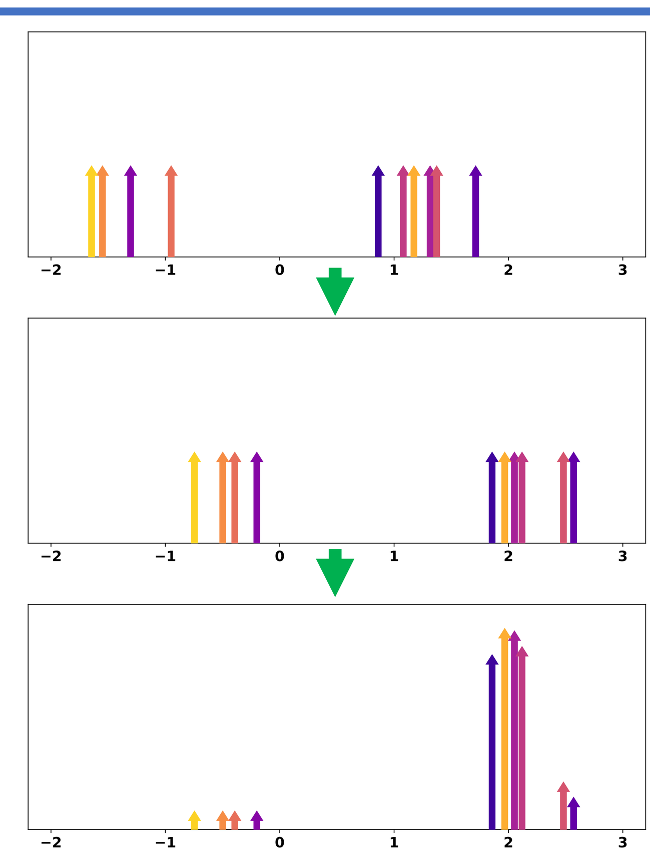
$x_t^{(i)} \sim f(x_t | x_{t-1} = x_{t-1}^{(i)}, a_t)$

**Step 2: Measurement Update:**

$w_t^{(i)} = l(y_t | x_t^{(i)}) \cdot w_{t-1}^{(i)}$ *(Generative)*

$w_t^{(i)} = l(x_t^{(i)} | y_t) \cdot w_{t-1}^{(i)}$ *(Discriminative)*

Normalize: $\sum_{i=1}^{N} w_t^{(i)} = 1$

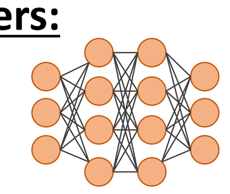


**Classical Particle Filters:**

$f(x_t | x_{t-1} = x_{t-1}^{(i)}, a_t)$, $l(x_t^{(i)} | y_t)$ := Known, Human Engineered

**Learnable Particle Filters:**

$f(x_t | x_{t-1} = x_{t-1}^{(i)}, a_t)$, $l(x_t^{(i)} | y_t)$ := Learnable Networks



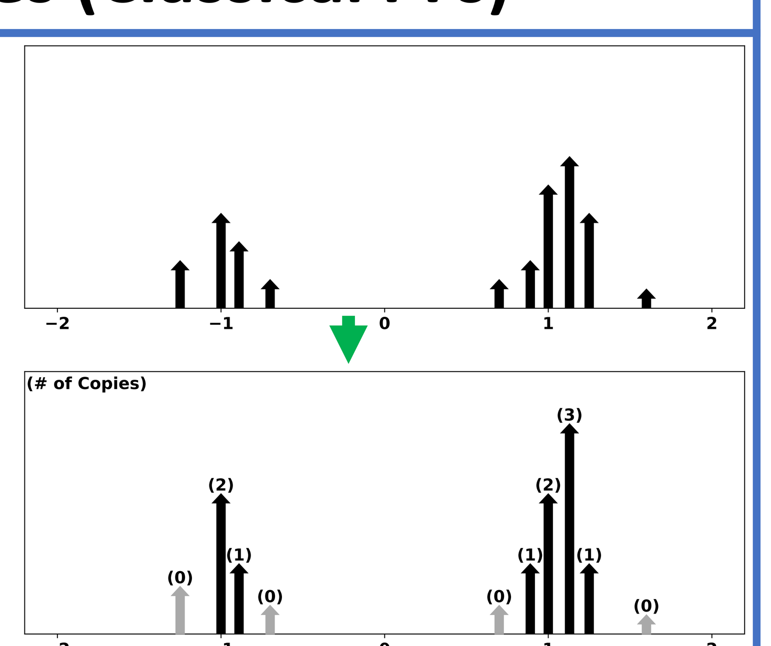
*Trained end-to-end*

## 3. Resampling Particles (Classical PFs)

**Discrete Resampling with Replacement:**

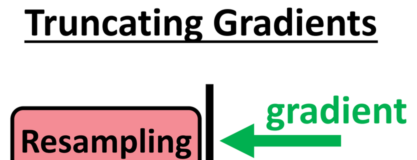
$\tilde{x}_t^{(i)} = x_t^{(j)} \quad j \sim \text{Cat}(w_t^{(1)}, \dots, w_t^{(N)})$

*Discrete sampling is not differentiable*



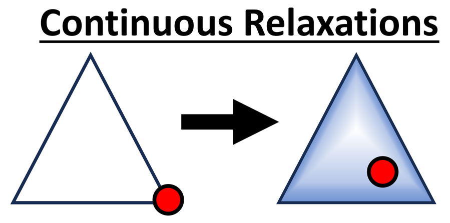

**Some Previously Proposed Methods:**

**Truncating Gradients**




Gradient truncated at each particle resampling

**Continuous Relaxations**



Concrete Distribution (Gumbel-Softmax)
Entropy-Regularized Optimal Transport

**Soft-Resampling**

$\widehat{w}_t^{(i)} = (1-\lambda) w_t^{(i)} + \lambda \frac{1}{N}$

$\tilde{x}_t^{(i)} = x_t^{(j)} \quad j \sim \text{Cat}(\widehat{w}_t^{(1)}, \dots, \widehat{w}_t^{(N)})$

$\widetilde{w}_t^{(i)} = w_t^{(i)} / \widehat{w}_t^{(i)}$

Gradient estimates biased

## 4. Mixture Density Particle Filter (MDPF)

**MDPF Resampling:**

1. Define mixture using *Kernel Density Estimation:*

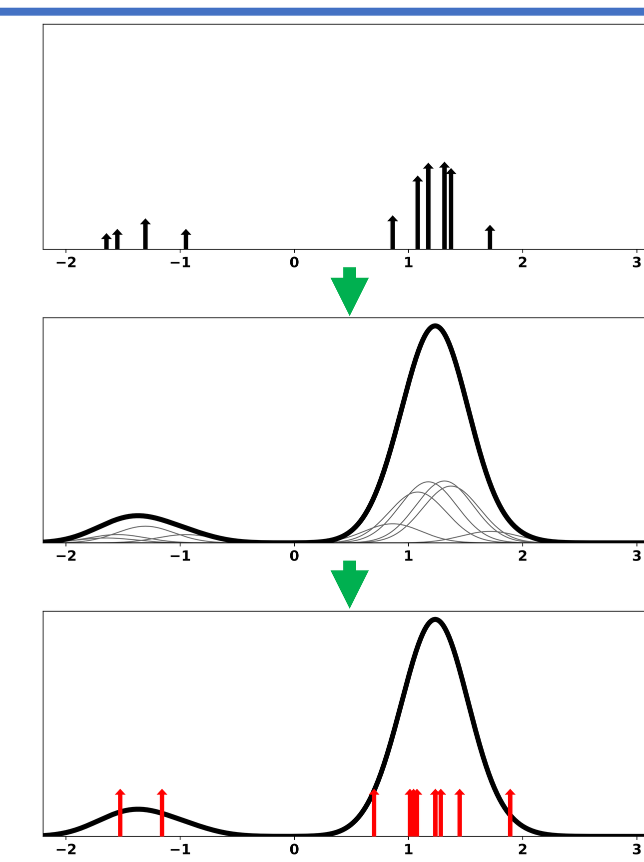
$$m_\phi(x) = \sum_{i=1}^{N} w_t^{(i)} \cdot K(x; x_t^{(i)}, \beta)$$

$\beta$:= Learned standard deviation (Bandwidth) (dimension specific)

$K(\cdot)$ := Kernel function (Any valid PDF)

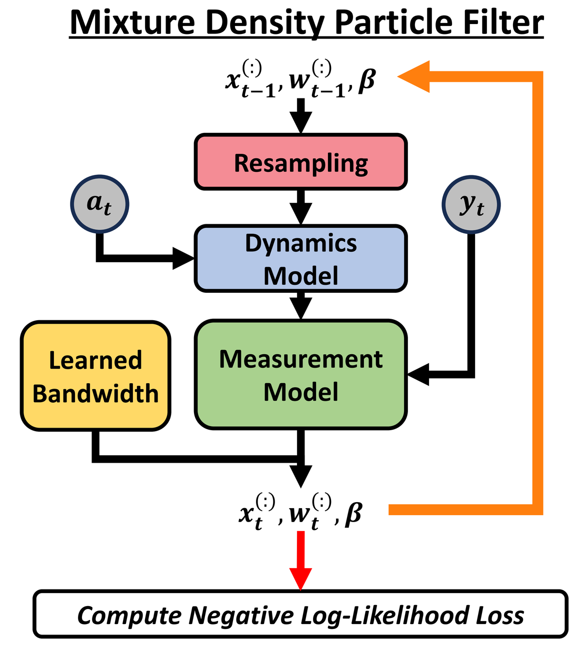
2. Resample From mixture distribution:

$$\tilde{x}_t^{(i)} \sim \sum_{j=1}^{N} w_t^{(j)} \cdot K(x; x_t^{(j)}, \beta)$$

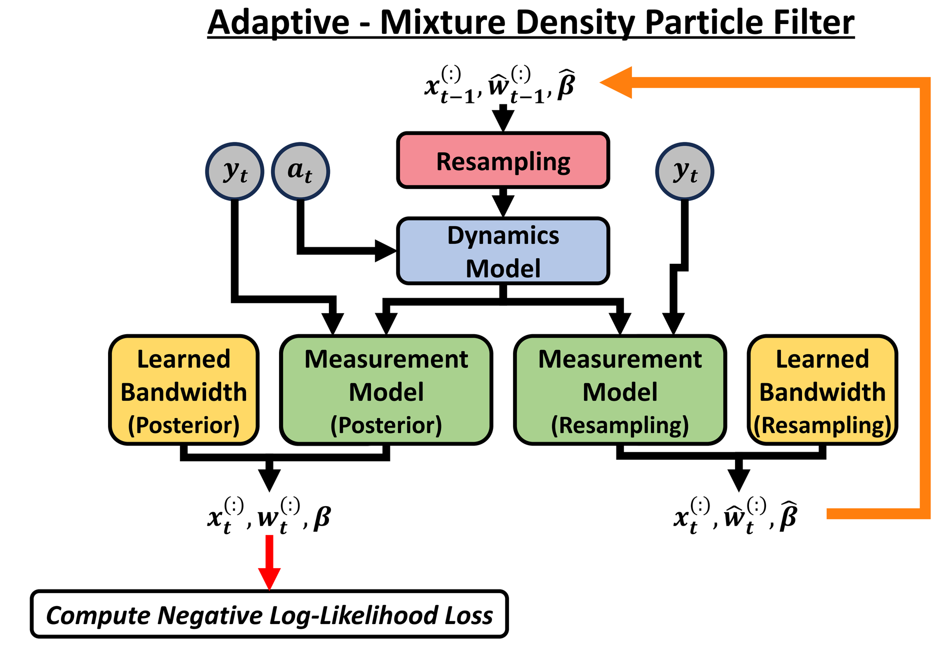


**Mixture Density Particle Filters:**

**Mixture Density Particle Filter**




**Adaptive - Mixture Density Particle Filter**




## 5. Existing Reparameterization Method

**Implicit Reparameterization Gradients**

Standardization Function (CDF) $\left(S_\phi\left(\tilde{x}_t^{(i)}\right) = \epsilon\right)$ + Implicit differentiation → $\nabla_\phi \tilde{x}_t^{(i)} = -\left(\nabla_{\tilde{x}_t^{(i)}} S_\phi\left(\tilde{x}_t^{(i)}\right)\right)^{-1} \nabla_\phi S_\phi\left(\tilde{x}_t^{(i)}\right)$

*Captures changes in the distribution by non-smoothly moving samples (particles can jump between modes)*

## 6. Importance Weighted Sample Gradients (IWSG)

**Desire samples from:** $\tilde{x}_t^{(i)} \sim m_\phi\left(\tilde{x}_t^{(i)}\right)$

**Define a proposal distribution:** $\tilde{x}_t^{(i)} \sim q\left(\tilde{x}_t^{(i)}\right)$

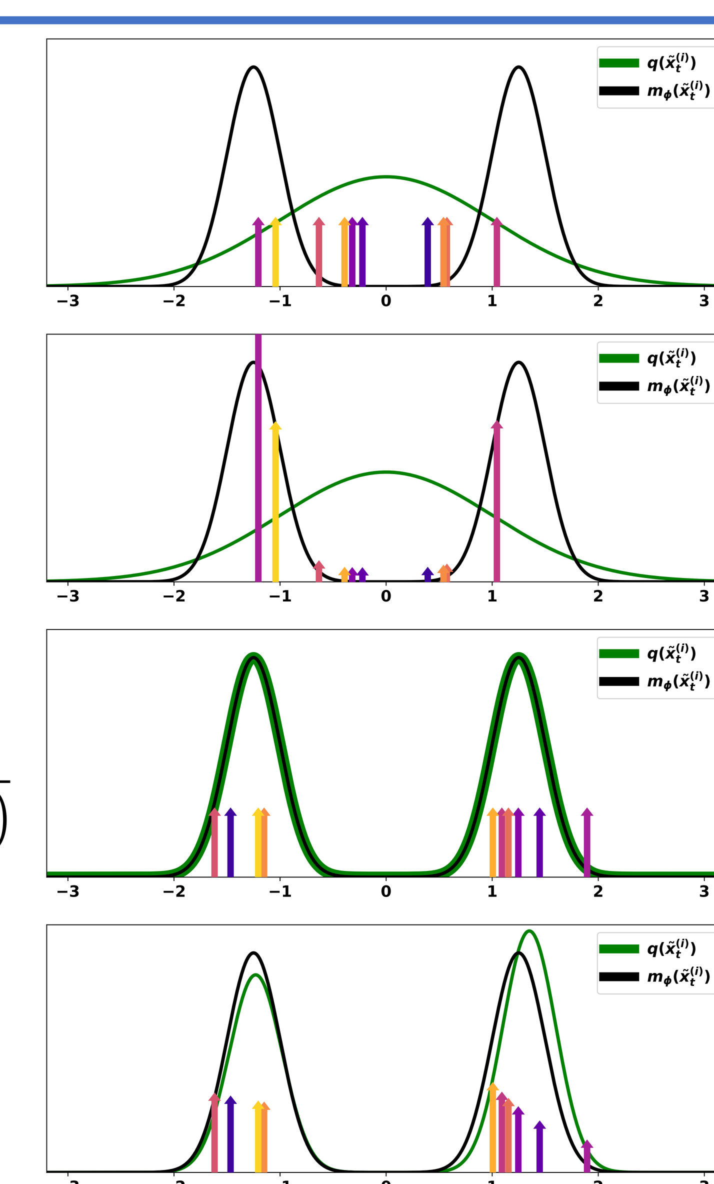
**Importance weight the samples:**

$$\widetilde{w}_t^{(i)} = \frac{m_\phi\left(\tilde{x}_t^{(i)}\right)}{q\left(\tilde{x}_t^{(i)}\right)} \qquad \nabla_\phi \widetilde{w}_t^{(i)} = \frac{\nabla_\phi m_\phi\left(\tilde{x}_t^{(i)}\right)}{q\left(\tilde{x}_t^{(i)}\right)}$$

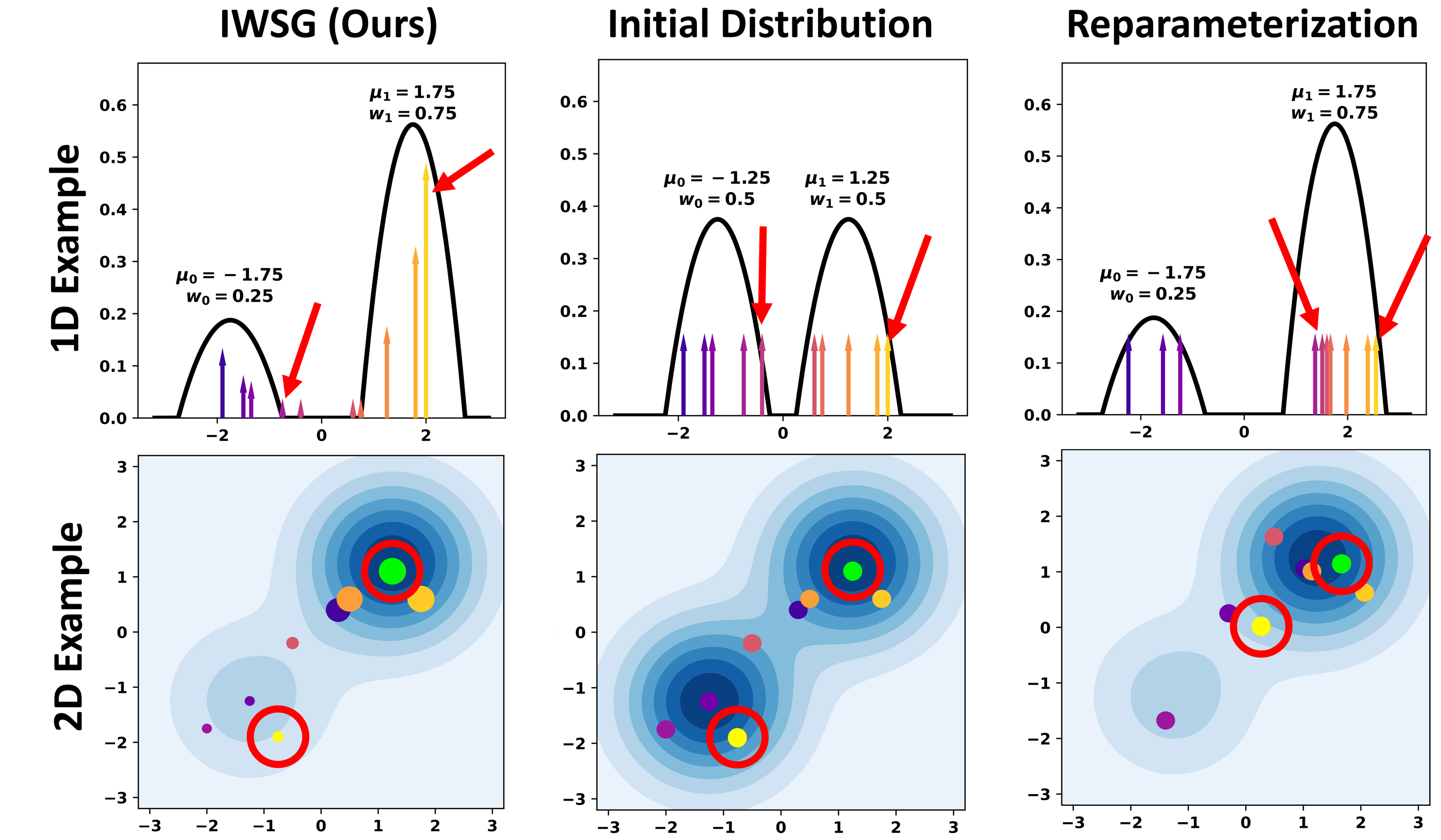
**Set** $q\left(\tilde{x}_t^{(i)}\right) = m_{\phi_0}\left(\tilde{x}_t^{(i)}\right)$ : *(with $\phi_0 = \phi$)*

$$\widetilde{w}_t^{(i)} = \frac{m_\phi\left(\tilde{x}_t^{(i)}\right)}{m_{\phi_0|\phi_0=\phi}\left(\tilde{x}_t^{(i)}\right)} = 1 \qquad \nabla_\phi \widetilde{w}_t^{(i)} = \frac{\nabla_\phi m_\phi\left(\tilde{x}_t^{(i)}\right)}{m_{\phi_0|\phi_0=\phi}\left(\tilde{x}_t^{(i)}\right)}$$

*Captures changes in the distribution by smoothly re-weighting samples*



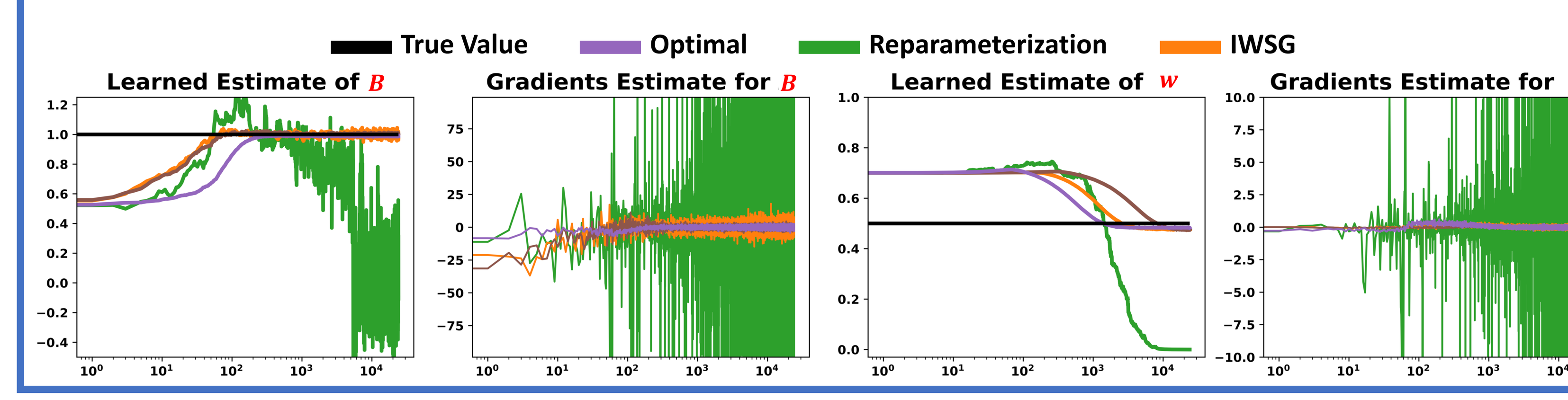
## 7. Reparameterization vs IWSG




**Linear Dynamical System Example:**

**True Dynamics:** $p(x_t | x_{t-1}, a_t) = \mathcal{N}(x_t | Ax_{t-1} + Ba_t, \sigma^2)$

**True Measurement model:** $p(y_t | x_t) = (1-w)\,\mathcal{N}(y_t | C_1 x_t + c_1, \gamma^2) + w\,\mathcal{N}(y_t | C_2 x_t + c_2, \gamma^2)$

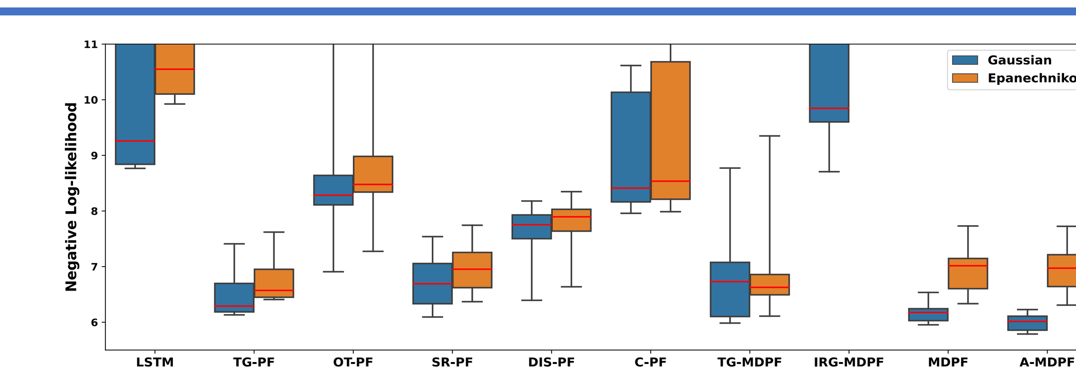


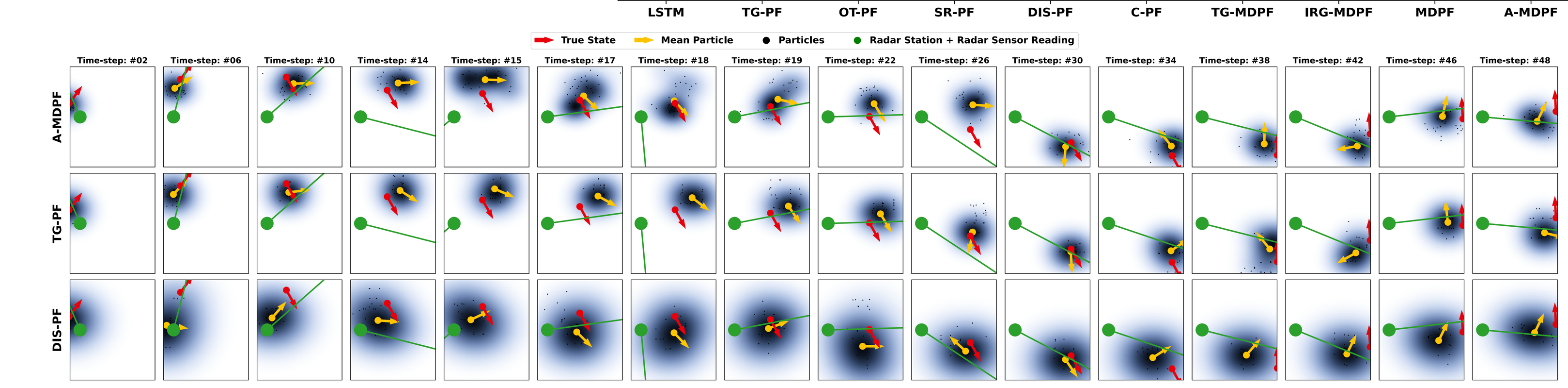

## 8. Results

**Bearings Only Task:**

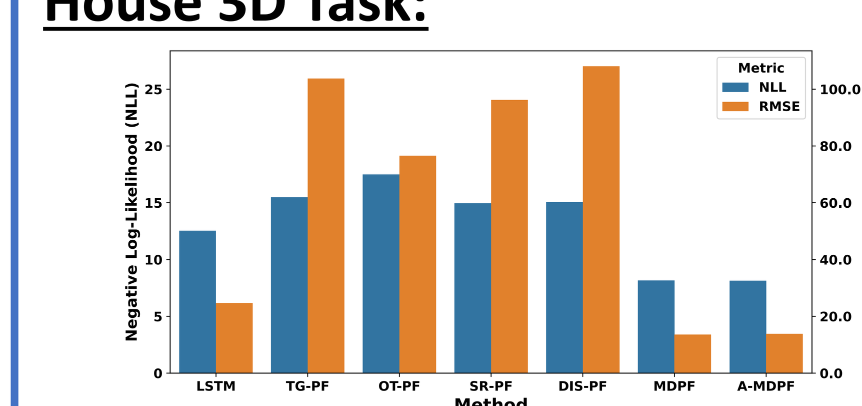
**Observation likelihood**

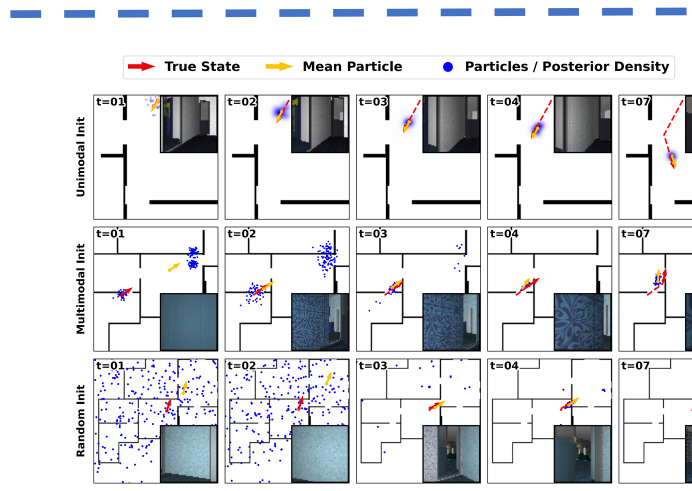
$p(y_t) = \alpha \text{VonMises}(y_t; \mu, \beta) + (1-\alpha)\text{Uniform}(0, 2\pi)$

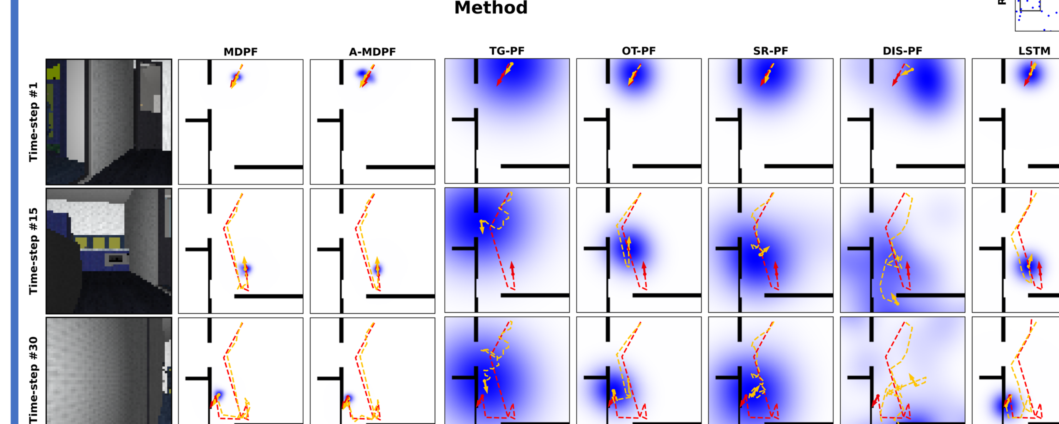


**House 3D Task:**



**Baselines:**

**TG-MDPF:** Truncated Gradients MDPF
**IRG-MDPF:** MDPF w/ Implicit Reparam. (Figurnov Neurips 2018)
**LSTM:** LSTM Baseline (Hochreiter Neural Computation 1997)
**TG-PF:** Truncated Gradient PF (Jonschkowski RSS 2018)
**OT-PF:** Optimal Transport PF (Corenflos ICML 2021)
**DIS-PF:** Discrete Importance Sampling PF (Ścibior *PROBPROG* 2021)
**SR-PF:** Soft Resampling PF (Karkus CORL 2018)
**C-PF:** Concrete distribution (Maddison ICLR 2017) PF


This research supported in part by:
- NSF Robust Intelligence Award No. IIS-1816365
- ONR Award No. N00014-23-1-2712