Fast Inference and Update of Probabilistic Density Estimation on Trajectory Prediction

Takahiro Maeda, Norimichi Ukita
Toyota Technological Institute, Japan
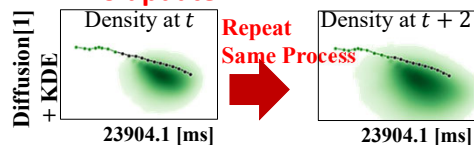
ICCV23 PARIS

Overview
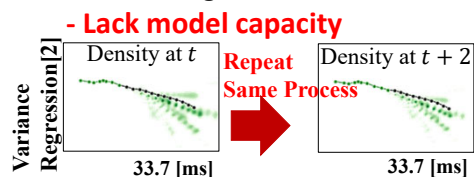
Trajectory Prediction needs
- Probability Density Estimation
- Fast Inference

Previous Methods
- Generative models (*e.g.* diffusions)
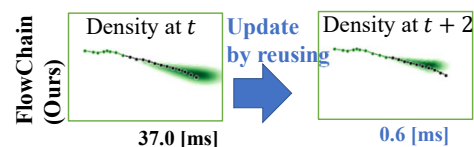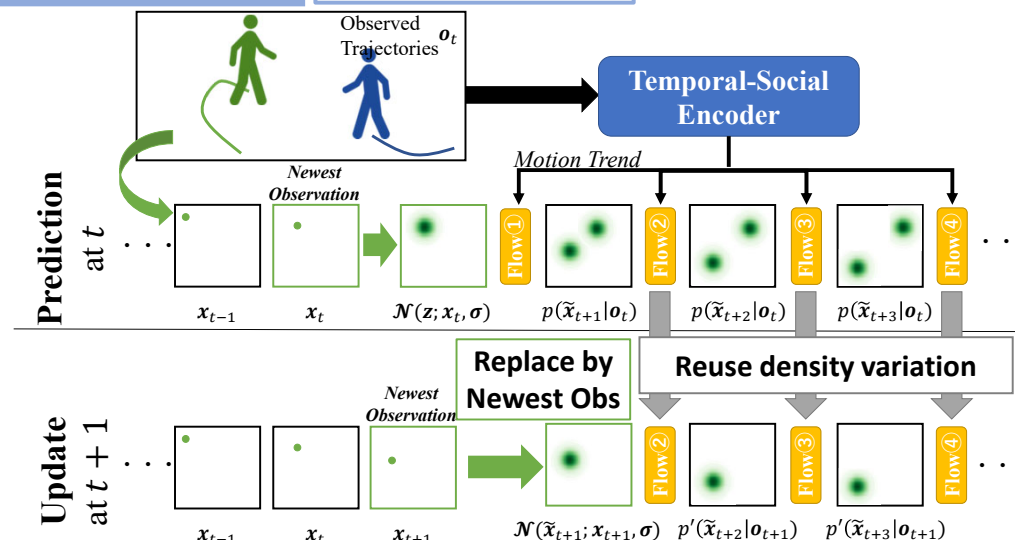 - **Slow** due to KDE and sampling
  - **No update**

Diffusion[1] + KDE

Density at t — 23904.1 [ms]
Repeat Same Process
Density at $t+2$ — 23904.1 [ms]

- Variance Regression
  - **Lack model capacity**

Variance Regression[2]

Density at t — 33.7 [ms]
Repeat Same Process
Density at $t+2$ — 33.7 [ms]

Our FlowChain
- **Fast** thanks to analytical comp.
- **Reusing and Ultra-fast update**

FlowChain (Ours)

Density at t — 37.0 [ms]
Update by reusing
Density at $t+2$ — 0.6 [ms]

Methodology

Model Overview

Observed Trajectories o_t → Temporal-Social Encoder

Motion Trend

Prediction at t

x_{t-1} x_t $\mathcal{N}(z;x_t,\sigma)$ | Flow1 $p(\tilde{x}_{t+1}|o_t)$ | Flow2 $p(\tilde{x}_{t+2}|o_t)$ | Flow3 $p(\tilde{x}_{t+3}|o_t)$ | Flow4 ...

Replace by Newest Obs | **Reuse density variation**

Update at $t+1$

x_{t-1} x_t x_{t+1} *Newest Observation* $\mathcal{N}(\tilde{x}_{t+1};x_{t+1},\sigma)$ | Flow2 $p'(\tilde{x}_{t+2}|o_{t+1})$ | Flow3 $p'(\tilde{x}_{t+3}|o_{t+1})$ | Flow4 ...

Why reusing is possible?

Observations at t — $t-1$, t
Observations at $t+1$ — $t-1$, t, $t+1$

Reused variation based on *Motion Trend* e.g. Velocity, Direction

Small difference

Why reusing matters?

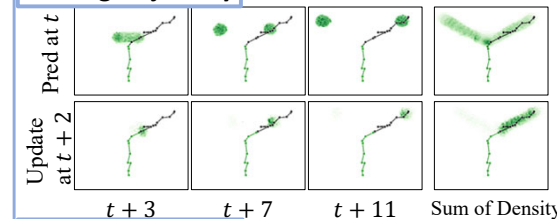**99% of comp. cost is due to density variation**

- **Trajectory Encoder**
- **Log-det-jacobian of flow**

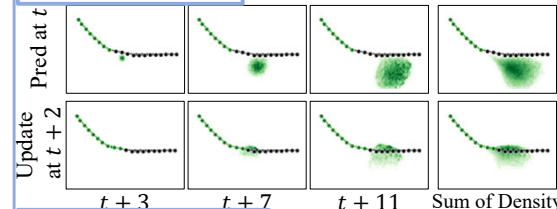**By reusing density variation, we achieve Ultra-fast update**

Results

Observation: | Estimated dist.: | GT future:

Forking trajectory

Pred at t / Update at $t+2$ — $t+3$, $t+7$, $t+11$, Sum of Density

Turning trajectory

Pred at t / Update at $t+2$ — $t+3$, $t+7$, $t+11$, Sum of Density

Quantitative Results

Method	Accumulated Comp Time[s]			Log-probability↑		
	Initial	Update@1	Update@5	Initial	Update@1	Update@5
MID[1]	23.9	47.8	119.5	-0.90	-0.90	-0.90
Trajectron[2]	0.47	0.94	2.35	-281.86	-281.86	-281.86
Ours	0.63	0.63	0.63	**-0.26**	**-0.45**	**-0.53**

Limitation
- Performance degrades via our ultra-fast update.
- FlowChain only accepts traj. with fixed interval

Reference & Link
[1] Motion Indeterminacy diffusion. CVPR2022
[2] Trajectron++. ECCV2020

Code is available on Github!!!