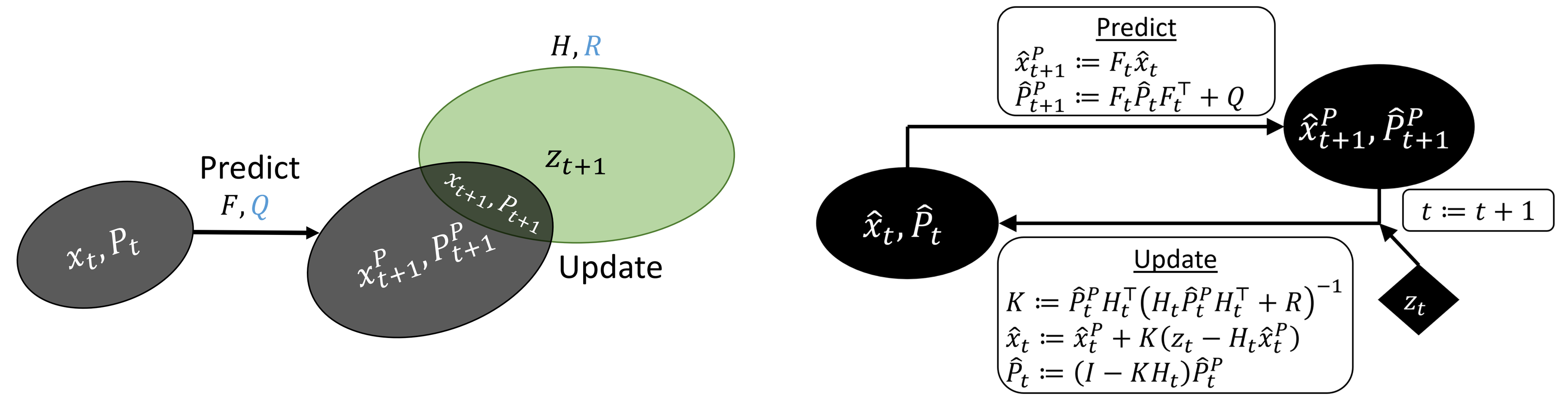
Optimization or Architecture: How to Hack Kalman Filtering

TECHNION[1] Israel Institute of Technology
IAI ELTA[2]
NVIDIA[3]
Ido Greenberg[1], Netanel Yannay[2] and Shie Mannor[1,3], *NeurIPS 2023*

Background: Kalman Filter (KF)

- Sequential prediction from noisy observations

```
For step t:
  Predict: progress time (dynamics F, noise Q)
  Filter: process observation (transformation H, noise R)
```



How to know the noise Q, R?

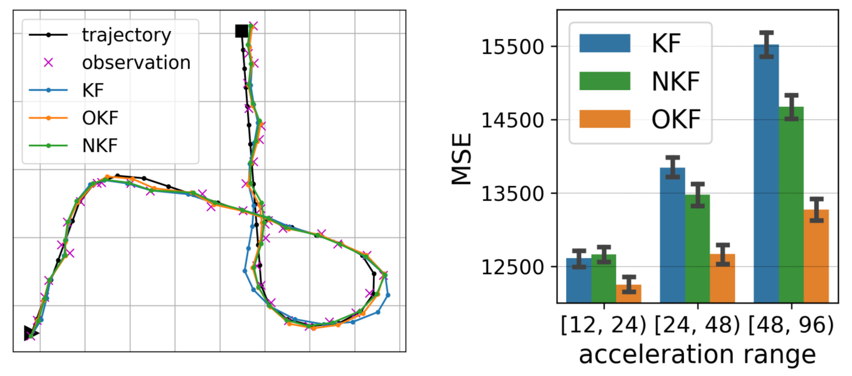
Noise estimation ("Algorithm 1")

$$Q := Cov(\{x_{t+1} - Fx_t\}_t) \qquad R := Cov(\{z_t - Hx_t\}_t)$$

- Need supervised data: states + observations $\{(x,z)\}$
- Optimal MSE under certain assumptions

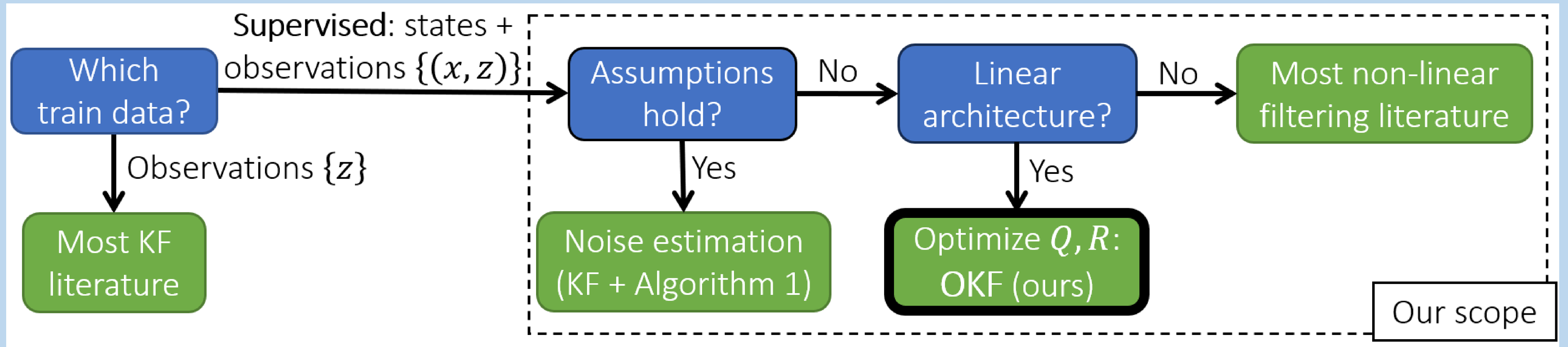
How it started?

- Our original draft:
  - Neural network beats the KF
  - ⇒ Need non-linear *architecture*?
- But then:
  - Optimized KF beats the neural network
  - ⇒ The network was better only because it was *optimized* more than the KF!



Neural network (NKF) beats the KF, but loses to an Optimized KF (OKF)

When in doubt: Always optimize your Kalman Filter



Why?

Otherwise:

- Linear filtering:
  - Sub-optimal predictions
  - Anomalies (e.g., more data ⇒ worse MSE)
- Non-linear filtering:
  - When beating the KF baseline, can't tell if the network is better or just more optimized

How?

- Like RNN
  - predict sequence
  - compute loss (e.g., MSE)
  - backpropagate grads
  - optimize
  - repeat
- SPD ⇒ Cholesky
- PyPI

`Optimized-Kalman-Filter`

Isn't optimization straight-forward?

- Mostly
  - Make sure to optimize your actual loss – less important how
- SPD challenge
  - Q, R must be Symmetric & Positive-Definite
  - Cholesky parameterization to rescue: $Q = LL^\top$
    - Optimize the entries of (lower-triangular) L

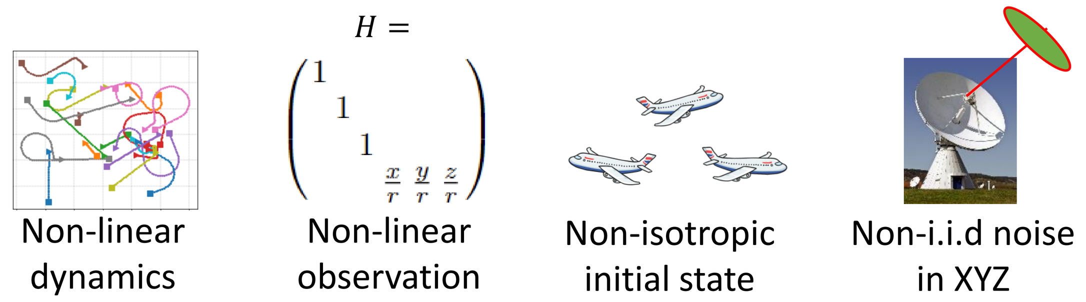
Why optimize?

Aren't Q, R already optimal?

- Only under strict assumptions
  - Linear models; i.i.d noise; known initial distribution
- Otherwise: Noise estimation is *not* a proxy to MSE optimization

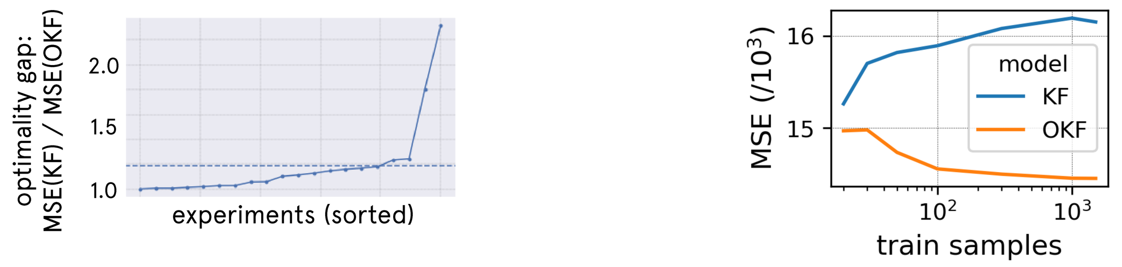
Don't the assumptions hold?

- Example – multiple violations in a simple Doppler radar:



Does it *really* make a difference?

- Proposition: Unbounded sub-optimality in toy Doppler radar problem
- Experiments:



OKF beats KF consistently

Noise estimation does not optimize MSE ⇒ deteriorates with more data!

Is KF *really* used w/o optimization?

- In linear filtering: "the systematic and preferable approach to determine the filter gain is to estimate the covariances from data" [Odelson, 2006]
- In non-linear filtering:
  - We cite 10 works that compare an optimized neural network to a non-optimized KF
    - In all these works – did the network actually help?