A APPENDIX

A.1 BASELINE BAYESIAN FILTER METHODS AND IMPLEMENTATION DETAILS

In this section we define the Kalman filter (KF), Extended Kalman filter (EKF), iterated extended Kalman filter (IEKF), unscented Kalman filter (UKF), particle filter (PF), and variational Kalman filter (VKF) as implemented in the experimental section of this paper. We closely follow the definitions and notation of Särkkä & Svensson (2023) and refer the reader to that text for proofs and a more extensive treatment of these topics.

We use this section to state design choices made for each baseline filter when it comes to the experiments reported in the main section.

A.1.1 KALMAN FILTER

Assume the following state space model:

$$\begin{aligned} \mathbf{x}_t &= \boldsymbol{F}_{t-1}\mathbf{x}_{t-1} + \boldsymbol{q}_{t-1} \\ \mathbf{y}_t &= \boldsymbol{H}_t\mathbf{x}_t + \boldsymbol{r}_t \end{aligned}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state, $\mathbf{y}_t \in \mathbb{R}^m$ is the measurement, $\boldsymbol{F}_{t-1} \in \mathbb{R}^{n\times n}$ is the transition matrix, $\boldsymbol{H}_t \in \mathbb{R}^{m\times n}$ is the measurement model matrix, $\boldsymbol{q}_{t-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_{t-1})$ is the process noise and $\boldsymbol{r}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ is the measurement noise.

The Kalman filter (Kalman, 1960) defines the predictive distribution, filtering distribution, and marginal likelihood at time step t as

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) &= \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_t^-, \boldsymbol{\Sigma}_t^-), \\ p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \\ p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) &= \mathcal{N}(\mathbf{x}_t|\boldsymbol{H}_t\boldsymbol{\mu}_t^-, \boldsymbol{S}_t), \end{aligned}$$

with predict step:

$$\begin{aligned} \boldsymbol{\mu}_t^- &= \boldsymbol{F}_{t-1}\boldsymbol{\mu}_{t-1}, \\ \boldsymbol{\Sigma}_t^- &= \boldsymbol{F}_{t-1}\boldsymbol{\Sigma}_{t-1}\boldsymbol{F}_{t-1}^\top + \boldsymbol{Q}_{t-1}, \end{aligned}$$

and update step:

$$\begin{aligned} \boldsymbol{v}_t &= \mathbf{y}_t - \boldsymbol{H}_t\boldsymbol{\mu}_t^-, \\ \boldsymbol{S}_t, &= \boldsymbol{H}_t\boldsymbol{\Sigma}_t^-\boldsymbol{H}_t^\top + \boldsymbol{R}_t, \\ \boldsymbol{K}_t &= \boldsymbol{\Sigma}_t^-\boldsymbol{H}_t^\top\boldsymbol{S}_t^{-1}, \\ \boldsymbol{\mu}_t &= \boldsymbol{\mu}_t^- + \boldsymbol{K}_t\boldsymbol{v}_t, \\ \boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_t^- - \boldsymbol{K}_t\boldsymbol{S}_t\boldsymbol{K}_t^\top. \end{aligned}$$

Recursion begins from some $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. In the stochastic Lorenz Attractor, the Gaussian random walk (GRW) case is equivalent to a Kalman filter where $\boldsymbol{F}_{t-1} = \boldsymbol{I}_3$ and $\boldsymbol{H}_t = \boldsymbol{I}_3$ at every time step t. We do not see the linear-Gaussian Kalman filter in any other experiments.

A.1.2 EXTENDED KALMAN FILTER

In the extended Kalman filter (EKF), we use first-order Taylor approximations of the nonlinear transition function f and measurement model h where appropriate. We denote the Jacobians of these functions as $\boldsymbol{F}_{\mathbf{x}}(\cdot)$ and $\boldsymbol{H}_{\mathbf{x}}(\cdot)$. The predict step is now

$$\begin{aligned} \boldsymbol{\mu}_t^- &= f(\boldsymbol{\mu}_{t-1}), \\ \boldsymbol{\Sigma}_t^- &= \boldsymbol{F}_{\mathbf{x}}(\boldsymbol{\mu}_{t-1})\boldsymbol{\Sigma}_{t-1}\boldsymbol{F}_{\mathbf{x}}(\boldsymbol{\mu}_{t-1})^\top + \boldsymbol{Q}_{t-1}, \end{aligned}$$

and the update step is

$$\begin{aligned}
\boldsymbol{v}_t &= \mathbf{y}_t - h(\boldsymbol{\mu}_t^-),\\
\boldsymbol{S}_t, &= \boldsymbol{H}_{\mathbf{x}}(\boldsymbol{\mu}_t^-)\boldsymbol{\Sigma}_t^-\boldsymbol{H}_{\mathbf{x}}(\boldsymbol{\mu}_t^-)^\top + \boldsymbol{R}_t,\\
\boldsymbol{K}_t &= \boldsymbol{\Sigma}_t^-\boldsymbol{H}_{\mathbf{x}}(\boldsymbol{\mu}_t^-)^\top\boldsymbol{S}_t^{-1},\\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_t^- + \boldsymbol{K}_t\boldsymbol{v}_t,\\
\boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_t^- - \boldsymbol{K}_t\boldsymbol{S}_t\boldsymbol{K}_t^\top,
\end{aligned}$$

with recursion starting from some $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ as before.

A.1.3 ITERATED EXTENDED KALMAN FILTER

The iterated extended Kalman filter (IEKF) (Gelb, 1974) shares an identical predict step with the EKF. The update step is redefined as the following iterative procedure, starting from $\mathbf{x}_t^{(0)} = \boldsymbol{\mu}_t^-$ from the predict step:

- For $i = 1, 2, \dots K$:

$$\begin{aligned}
\boldsymbol{v}_t^{(i)} &= \mathbf{y}_t - h(\mathbf{x}_t^{(i-1)}) - \boldsymbol{H}_{\mathbf{x}}(\mathbf{x}_t^{(i-1)})(\boldsymbol{\mu}^- - \mathbf{x}_t^{(i-1)}),\\
\boldsymbol{S}_t^{(i)}, &= \boldsymbol{H}_{\mathbf{x}}(\mathbf{x}_t^{(i-1)})\boldsymbol{\Sigma}_t^-\boldsymbol{H}_{\mathbf{x}}(\mathbf{x}_t^{(i-1)})^\top + \boldsymbol{R}_t,\\
\boldsymbol{K}_t^{(i)} &= \boldsymbol{\Sigma}_t^-\boldsymbol{H}_{\mathbf{x}}(\mathbf{x}_t^{(i-1)})^\top\left[\boldsymbol{S}_t^{(i)}\right]^{-1},\\
\mathbf{x}_t^{(i)} &= \boldsymbol{\mu}_t^- + \boldsymbol{K}_t^{(i)}\boldsymbol{v}_t^{(i)}.
\end{aligned}$$

- Set $\boldsymbol{\mu}_t = \mathbf{x}_t^{(K)}$.
- Set $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_t^- - \boldsymbol{K}_t^{(K)}\boldsymbol{S}_t^{(K)}\left[\boldsymbol{K}_t^{(K)}\right]^\top$.

For both the stochastic Lorenz Attractor and the toy nonlinear system, we ran IEKFs for iterations $K \in \{1, 3, 5, 10, 25, 50, 100\}$. We consistently report the IEKF that showed the highest average RMSE over 100 Monte Carlo experiments.

A.1.4 UNSCENTED KALMAN FILTER

The predict step for the Unscented Kalman filter (UKF) is as follows:

- Form $2n + 1$ sigma points where

$$\begin{aligned}
\chi_{t-1}^{(0)} &= \boldsymbol{\mu}_{t-1},\\
\chi_{t-1}^{(i)} &= \boldsymbol{\mu}_{t-1} + \sqrt{n+\lambda}\left[\sqrt{\boldsymbol{\Sigma}_{t-1}}\right]_i,\\
\chi_{t-1}^{(i+n)} &= \boldsymbol{\mu}_{t-1} - \sqrt{n+\lambda}\left[\sqrt{\boldsymbol{\Sigma}_{t-1}}\right]_i,
\end{aligned}$$

  for $i = 1, \dots, n$ where n is the dimension of the state space and λ is a tuneable parameter defined below.
- Apply the dynamics model to the sigma points:

$$\hat{\chi}_t^{(i)} = f(\chi_{t-1}^{(i)})$$

  for $i = 0, \dots, 2n$.
- Compute the predict step mean and covariance:

$$\begin{aligned}
\boldsymbol{\mu}_t^- &= \sum_{i=0}^{2n} w_i^{(m)}\hat{\chi}_t^{(i)},\\
\boldsymbol{\Sigma}_t^- &= \sum_{i=0}^{2n} w_i^{(c)}\left(\hat{\chi}_t^{(i)} - \boldsymbol{\mu}_t^-\right)\left(\hat{\chi}_t^{(i)} - \boldsymbol{\mu}_t^-\right)^\top + \boldsymbol{Q}_{t-1},
\end{aligned}$$

where

$$\begin{aligned}
w_0^{(m)} &= \frac{\lambda}{n+\lambda}, \\
w_0^{(c)} &= \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta), \\
w_i^{(m)} &= \frac{\lambda}{2(n+\lambda)}, \\
w_i^{(c)} &= \frac{\lambda}{2(n+\lambda)},
\end{aligned}$$

where α, β are tuneable parameters.

The update step:

- Form $2n+1$ sigma points as before except replace $\boldsymbol{\mu}_{t-1}$ and $\boldsymbol{\Sigma}_{t-1}$ with $\boldsymbol{\mu}_t^-$ and $\boldsymbol{\Sigma}_t^-$ from the predict step. Denote these sigma points as $\hat{\chi}_t^{-(i)}$ for $i = 0, \ldots, 2n$.
- Apply the measurment model to the sigma points:

  $$\hat{\mathcal{Y}}_t^{(i)} = h(\chi_{t-1}^{(i)}),$$

  for $i = 0, \ldots, 2n$.
- Compute:

  $$\begin{aligned}
  \boldsymbol{m}_t &= \sum_{i=0}^{2n} w_i^{(m)} \hat{\mathcal{Y}}_t^{(i)}, \\
  \boldsymbol{S}_t &= \sum_{i=0}^{2n} w_i^{(c)} \left(\hat{\mathcal{Y}}_t^{(i)} - \boldsymbol{m}_t\right) \left(\hat{\mathcal{Y}}_t^{(i)} - \boldsymbol{m}_t\right)^\top + \boldsymbol{R}_t, \\
  \boldsymbol{C}_t &= \sum_{i=0}^{2n} w_i^{(c)} \left(\hat{\chi}_t^{-(i)} - \boldsymbol{\mu}_t^-\right) \left(\hat{\mathcal{Y}}_t^{(i)} - \boldsymbol{m}_t\right)^\top.
  \end{aligned}$$

- Compute the Kalman gain and perform the update:

  $$\begin{aligned}
  \boldsymbol{K}_t &= \boldsymbol{C}_t \boldsymbol{S}_t^{-1}, \\
  \boldsymbol{\mu}_t &= \boldsymbol{\mu}_t^- + \boldsymbol{K}_t \left(\mathbf{y}_t - \boldsymbol{m}_t\right), \\
  \boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_t^- - \boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^\top.
  \end{aligned}$$

For both the stochastic Lorenz Attractor and the toy nonlinear system, we set $\alpha = 1$, $\beta = 3 - n$, $\lambda = \alpha^2 \cdot (n + \beta) - n$.

A.1.5 PARTICLE FILTER

We ran a Bootstrap filter (BF) with 1000 particles for every experiment reported in the main section. The BF algorithm at every time step is as follows:

- Sample

  $$\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)})$$

  for every ith particle.
- Compute weights

  $$w_t^{(i)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(i)})$$

  for every ith particle and normalize.

- Perform resampling.

We did multinomial resampling at every time step t. For the yearbook dataset, since the transition distribution was unknown, we used the following transition distribution:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}) = \mathcal{N}(p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \sigma^2 \boldsymbol{I}) \tag{17}$$

where $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. Selecting σ^2 requires a grid search, which is described in Appendix A.4.

A.1.6 VARIATIONAL KALMAN FILTER

A variational Kalman filter (VKF) was used in the Yearbook experiments due to the storage complexity of the EKF and UKF. Similar to the Implicit MAP Filter, the VKF we implemented propagates a point-mass forward in time but differs in that it models the prior uncertainty *explicitly* rather than *implicitly*.

Assume that the initial distribution is $p(\mathbf{w}_0) = \mathcal{N}(\mathbf{w}_0|\mathbf{0}, \sigma_0^2\boldsymbol{I})$, the transition distribution is $p(\mathbf{w}_t|\mathbf{w}_{t-1}) = \mathcal{N}(\mathbf{w}_t|\mathbf{w}_{t-1}, \sigma^2\boldsymbol{I})$, and the likelihood is $p(\mathbf{y}_t|\boldsymbol{X}_t) = \prod_{i=1}^{n_t} p_{\mathbf{w}_t}(y_{ti}|\boldsymbol{X}_{ti})$, where $p_{\mathbf{w}}(y|\boldsymbol{X})$ is the output of a neural network with parameters $\mathbf{w}$ that produces a distribution over target y given input $\boldsymbol{X}$. We assume that $p(\mathbf{w}_t|\boldsymbol{X}_{1:t}, \mathbf{y}_{1:t}) \approx \delta(\hat{\mathbf{w}}_t - \mathbf{w}_t)$ where $\delta(\cdot)$ denotes a Dirac-delta function centered at $\mathbf{w}_t$.

The VKF algorithm at every time step is as follows:

- Predict step:

$$\begin{aligned} p(\mathbf{w}_t|\boldsymbol{X}_{1:t-1}, \mathbf{y}_{1:t-1}) &= \int \delta(\hat{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1})\mathcal{N}(\mathbf{w}_t|\mathbf{w}_{t-1}, \sigma^2\boldsymbol{I})\mathrm{d}\mathbf{w}_{t-1} \\ &= \mathcal{N}(\mathbf{w}_t|\hat{\mathbf{w}}_{t-1}, \sigma^2\boldsymbol{I}) \end{aligned}$$

- Update step:

$$p(\mathbf{w}_t|\boldsymbol{X}_{1:t}, \mathbf{y}_{1:t}) \propto \mathcal{N}(\mathbf{w}_t|\hat{\mathbf{w}}_{t-1}, \sigma^2\boldsymbol{I}) \prod_{i=1}^{n_t} p_{\mathbf{w}_t}(y_{ti}|\boldsymbol{X}_{ti})$$

Now, we approximate $p(\mathbf{w}_t|\boldsymbol{X}_{1:t}, \mathbf{y}_{1:t})$ with a point mass $\hat{\mathbf{w}}_t$ such that

$$\begin{aligned} \hat{\mathbf{w}}_t &\triangleq \arg\max_{\mathbf{w}_t} \left[\log \mathcal{N}(\mathbf{w}_t|\hat{\mathbf{w}}_{t-1}, \sigma^2\boldsymbol{I}) + \sum_{i=1}^{n_t} \log p_{\mathbf{w}_t}(y_{ti}|\boldsymbol{X}_{ti}) \right] \\ &= \arg\min_{\mathbf{w}_t} \left[\underbrace{-\frac{1}{n_t} \sum_{i=1}^{n_t} \log p_{\mathbf{w}_t}(y_{ti}|\boldsymbol{X}_{ti})}_{\text{mean binary cross entropy}} + \underbrace{\frac{1}{2n_t\sigma^2} \sum_{d=1}^{D} (w_{td} - \hat{w}_{t-1,d})^2}_{\text{weight decay centered at } \hat{\mathbf{w}}_{t-1}} \right] \end{aligned}$$

In the yearbook experiment, we run 5 configurations with $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. We optimize using the same procedure as the direct fit case, with the exception that the objective has explicit regularization.

A.2 VARIATIONAL INFERENCE INTERPRETATION OF THE UPDATE STEP

In this section, we show how truncated gradient descent on the likelihood (12) can be interpreted as variational inference with the variational distribution $q_t(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t \mid \mathbf{m}_t, \mathbf{M}_t)$ where $\mathbf{M}_t^{-1} \succ \mathbf{0}$ is fixed. The derivation in this section is based on Section 4.1 of Khan & Rue (2023), but adapts it to the setting of the update step of a Bayesian filter and further applies the equivalence due to Santos (1996) in order to deduce the implied filtering covariance $\boldsymbol{\Sigma}_t^-$. The variational lower bound on the log marginal likelihood for time step t is:

$$\begin{aligned} \log p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) &\geq \mathbb{E}_{q_t}\left[\log \frac{p(\mathbf{x}_t, \mathbf{y}_t \mid \mathbf{y}_{1:t-1})}{q_t(\mathbf{x}_t)}\right] && (18) \\ &= \mathbb{E}_{q_t}[\log p(\mathbf{y}_t \mid \mathbf{x}_t) + \log p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})] + \mathcal{H}(q_t) && (19) \end{aligned}$$

Then the optimal variational distribution is:

$$q_t^*(\mathbf{x}_t) = \arg\min_{q_t} \mathbb{E}_{q_t}[\bar{\ell}_t(\mathbf{x}_t)] - \mathcal{H}(q_t), \text{ where} \tag{20}$$

$$\bar{\ell}_t(\mathbf{x}_t) \triangleq -\log p(\mathbf{y}_t \mid \mathbf{x}_t) - \log p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) \tag{21}$$

$$= -\log \mathcal{N}(\mathbf{y}_t \mid \mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t) - \log \mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t^-, \boldsymbol{\Sigma}_t^-) \tag{22}$$

We take the approach of Khan & Rue (2023) where the variational optimization is performed using natural gradient descent. In particular, assume the variational distribution to be of the form

$$q_{\boldsymbol{\lambda}_t}(\mathbf{x}_t) = h(\mathbf{x}_t)\exp[\langle \boldsymbol{\lambda}_t, \mathbf{T}(\mathbf{x}_t)\rangle - A(\boldsymbol{\lambda}_t)] \tag{23}$$

Natural gradient descent on the variational objective can then be written as:

$$\boldsymbol{\lambda}_t^{(k+1)} \leftarrow (1-\rho_t^{(k)})\boldsymbol{\lambda}_t^{(k)} - \rho_t^{(k)}\nabla_{\boldsymbol{\mu}}\mathbb{E}_{q_t^{(k)}}[\bar{\ell}_t(\mathbf{x}_t) + \log h(\mathbf{x}_t)] \tag{24}$$

With the choice $q_t^{(k)}(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t \mid \mathbf{m}_t^{(k)}, \mathbf{M}_t)$, we have $\boldsymbol{\lambda}_t^{(k)} = \mathbf{M}_t^{-1}\mathbf{m}_t^{(k)}$, $\boldsymbol{\mu}_t^{(k)} = \mathbf{m}_t^{(k)}$ and $2\log h(\mathbf{x}_t) = P\log|2\pi\mathbf{M}_t^{-1}| - \mathbf{x}_t^\top\mathbf{M}_t^{-1}\mathbf{x}_t$. After substituting, we find that the updates can be written as follows:

$$\mathbf{x}_t^{(k+1)} \leftarrow \mathbf{x}_t^{(k)} - \rho_t^{(k)}\mathbf{S}_t^{-1}\nabla_{\mathbf{x}_t}\bar{\ell}_t(\mathbf{x}_t)|_{\mathbf{x}_t=\mathbf{m}_t^{(k)}}. \tag{25}$$

The gradient $\nabla_{\mathbf{x}_t}\bar{\ell}(\mathbf{x}_t)$ takes the following form:

$$\nabla_{\mathbf{x}_t}\bar{\ell}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t}\left[\frac{1}{2}\|\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t\|^2_{\mathbf{R}_t} + \frac{1}{2}\|\mathbf{x}_t - \boldsymbol{\mu}_t^-\|^2_{\boldsymbol{\Sigma}_t^-}\right] \tag{26}$$

$$= -\mathbf{H}_t^\top\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t) + (\boldsymbol{\Sigma}_t^-)^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t^-) \tag{27}$$

Now if we optimize this to convergence, we would recover the setting $\mathbf{m}_t^*$ such that $\mathcal{N}(\mathbf{x}_t \mid \mathbf{m}_t^*, \mathbf{M}_t) \approx p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$. This requires knowledge of both $\boldsymbol{\mu}_t^-$ and $\boldsymbol{\Sigma}_t^-$.

On the other hand, consider truncated gradient descent on the negative log-likelihood $\ell_t(\mathbf{x}_t)$:

$$\mathbf{x}_t^{(k+1)} \leftarrow \mathbf{x}_t^{(k)} - \rho_t^{(k)}\mathbf{M}_t\nabla_{\mathbf{x}_t}\ell_t(\mathbf{x}_t)|_{\mathbf{x}_t=\mathbf{m}_t^{(k)}}, \text{ where} \tag{28}$$

$$\ell(\mathbf{x}_t) \triangleq -\log\mathcal{N}(\mathbf{y}_t \mid \mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t) \tag{29}$$

$$\nabla_{\mathbf{x}_t}\ell(\mathbf{x}_t) = \nabla_{\mathbf{x}_t}\left[\frac{1}{2}\|\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t\|^2_{\mathbf{R}_t}\right] \tag{30}$$

$$= -\mathbf{H}_t^\top\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t) \Rightarrow \tag{31}$$

$$\mathbf{x}_t^{(k+1)} \leftarrow \mathbf{x}_t^{(k)} + \rho_t^{(k)}\mathbf{M}_t\mathbf{H}_t^\top\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t) \tag{32}$$

Assume that K such steps of gradient descent are taken where $\mathbf{x}_t^{(0)} = \boldsymbol{\mu}_t^-$ and $\rho_t^{(k)} = \rho_t$ for $k = 0, 1, \ldots, K-1$. Then by Santos (1996),

$$\mathbf{x}_t^{(K)} = \arg\min_{\mathbf{x}_t}\left[\frac{1}{2}\|\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t\|^2_{\mathbf{R}_t} + \frac{1}{2}\|\mathbf{x}_t - \boldsymbol{\mu}_t^-\|^2_{\boldsymbol{\Sigma}_t^-}\right], \tag{33}$$

where $\boldsymbol{\Sigma}_t^-$ is defined implicitly by the choice of ρ_t, $\mathbf{M}_t$, and K, in the sense we make explicit here. Let $\mathbf{C}_t$ simultaneously diagonalize $(1/\rho_t)\mathbf{M}_t^{-1}$ and $\mathbf{H}_t^\top\mathbf{R}_t^{-1}\mathbf{H}_t$ (this is possible since both are symmetric and $(1/\rho_t)\mathbf{M}_t^{-1}$ is positive definite as assumed above):

$$\mathbf{C}_t^\top(1/\rho_t)\mathbf{M}_t^{-1}\mathbf{C} = \mathbf{I} \tag{34}$$

$$\mathbf{C}_t^\top\mathbf{H}_t^\top\mathbf{R}_t^{-1}\mathbf{H}_t\mathbf{C}_t = \boldsymbol{\Lambda}_t \tag{35}$$

Then the equivalent $\boldsymbol{\Sigma}_t^-$ is defined to be:

$$\boldsymbol{\Sigma}_t^- = \mathbf{C}_t\text{diag}(\sigma_i)\mathbf{C}, \text{ where} \tag{36}$$

$$\sigma_i = (1/\lambda_i)[(1-\lambda_i)^{-k} - 1] \text{ if } \lambda_i \neq 0 \text{ and } 1 \text{ otherwise.} \tag{37}$$

A.3 JUSTIFICATION SETTING MEASUREMENT NOISE TO IDENTITY

For simplicity, consider the arbitrary specification of $\boldsymbol{R}_t = \boldsymbol{I}$. In an optimization scheme over a Gaussian likelihood, this is preferable because it reduces the objective function to the mean squared error loss, which is simple to implement and fast to compute. Such an arbitrary choice implies the *compensated* predict step covariance in the Kalman filter $\boldsymbol{\Sigma}_t^- = \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{R}_t^*\right)^{-1} \left(\boldsymbol{H}_t^\top\right)^+$, where $\boldsymbol{\Sigma}_t^*$ is the true predict step covariance and $\boldsymbol{R}_t^*$ is the true measurement noise at timestep t. The Moore–Penrose inverse of matrix $\boldsymbol{A}$ is denoted $\boldsymbol{A}^+$.

Proof: Let $\boldsymbol{\Sigma}_t^*$ be the true predict step covariance and $\boldsymbol{R}_t^*$ be the true measurement noise at timestep t. Suppose we wish to identify the quantity of $\boldsymbol{X}$ that makes the following expression true:

$$\boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} = \boldsymbol{X} \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \boldsymbol{I}_M\right)^{-1}.$$

This is exactly our Kalman gain expression where the L.H.S represents the optimal estimate and the R.H.S shows an arbitrary matrix $\boldsymbol{X}$ given $\boldsymbol{R}_t = \boldsymbol{I}_M$.

$$\begin{aligned}
\boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} &= \boldsymbol{X} \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \boldsymbol{I}_M\right)^{-1} \\
\boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} \boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} &= \boldsymbol{X} \boldsymbol{H}_t^\top \\
\left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} \boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \left(\boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top + \boldsymbol{R}_t^*\right)^{-1} &= \left(\boldsymbol{H}_t^\top\right)^+ \left(\boldsymbol{\Sigma}_t^*\right)^{-1} \boldsymbol{X} \boldsymbol{H}_t^\top \\
\boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \boldsymbol{I}_M = \boldsymbol{H}_t \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{H}_t^\top\right)^+ \left(\boldsymbol{\Sigma}_t^*\right)^{-1} \boldsymbol{X} \boldsymbol{H}_t^\top &+ \boldsymbol{R}_t^* \left(\boldsymbol{H}_t^\top\right)^+ \left(\boldsymbol{\Sigma}_t^*\right)^{-1} \boldsymbol{X} \boldsymbol{H}_t^\top \\
\boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top + \boldsymbol{I}_M = \boldsymbol{H}_t \boldsymbol{X} \boldsymbol{H}_t^\top &+ \boldsymbol{R}_t^* \left(\boldsymbol{H}_t^\top\right)^+ \left(\boldsymbol{\Sigma}_t^*\right)^{-1} \boldsymbol{X} \boldsymbol{H}_t^\top \\
\boldsymbol{X} = \boldsymbol{\Sigma}_t^* \boldsymbol{H}_t^\top \left(\boldsymbol{R}_t^*\right)^{-1} \left(\boldsymbol{H}_t^\top\right)^+ &= \boldsymbol{\Sigma}_t^-
\end{aligned}$$

□

Such a specification considerably simplifies computation in an *implicit* scheme. By setting $\boldsymbol{R}_t = \boldsymbol{I}$, we do not need to store $\boldsymbol{R}_t$ or perform any computations with it. We just need to pick an optimizer that correctly specifies this *compensated* predict step covariance. The argument extends to the nonlinear case with only minor adjustments.

A.4 GRID SEARCH DETAILS

The framework proposed in this paper attempts to define Bayesian filtering equations *implicitly* via specifying an appropriate optimizer. This requires a small validation set and a hyperparameter search, which we will now describe in the context of our experiments. We also describe the grid search performed for the extended Kalman filter (EKF) and unscented Kalman filter (UKF) in Section 5.2.

A.4.1 TOY NONLINEAR SYSTEM & STOCHASTIC LORENZ ATTRACTOR

In both the toy nonlinear system and the stochastic Lorenz attactor system, we perform a grid search for Adam (Kingma & Ba, 2015), RMSprop (Tieleman & Hinton, 2012), Adagrad (Duchi et al., 2011), Adadelta (Zeiler, 2012), and gradient descent. For 5 separate Monte Carlo (MC) experiments, we test every combination of step size $K \in \{1, 3, 5, 10, 25, 50, 100\}$, learning rate $\eta \in \{1.0, 0.5, 0.1, 0.05, 0.01\}$, and decay terms $\gamma, \beta_1, \beta_2 \in \{0.1, 0.5, 0.9\}$ where applicable. Adam's decay terms are β_1 and β_2 whereas γ only applies to RMSprop. For Adam, we always set $\beta_2 = \beta_1$ to simplify the search. In total, this corresponds to 7 test configurations for Adadelta, 35 test configurations each for gradient descent and Adagrad, and 105 test configurations each for RMSprop and Adam.

For only the stochastic Lorenz Attractor, we also perform a grid search for the process noise covariance $\boldsymbol{Q}_{t-1}$. The process noise covariance $\boldsymbol{Q}_{t-1}$ is determined by a scalar α that defines the spectral density of a three-dimensional Wiener process. We test four systems in this paper where $\alpha = 1, 5, 10, 20$. To optimize $\boldsymbol{Q}_{t-1}$, which is determined by α, we perform a grid search over 500 evenly spaced values of $\alpha \in [0.5, 250]$ for both the EKF and UKF in all four experiments. Comparatively, we test roughly 14 times the number of EKF configurations than the Implicit MAP Filter with gradient descent and roughly 2 times more configurations than all Implicit MAP Filters combined.

A.4.2 YEARBOOK

For this experiment, we reduce the grid search to only five configurations of Adam where all hyperparameters are set to standard settings ($\eta = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$) and the number of steps K is selected from the set $\{1, 10, 25, 50, 100\}$. As described in Appendix A.5.3, we divide up the 80 filtering years of the yearbook dataset into 40 tuning years and 40 testing years. During the tuning years, we have a small held-out validation set of 16 examples for each year that we use to calculate the classification accuracy for the parameters found by each optimizer. In the main paper, we report the optimizer that performs the best on this validation set.

We similarly perform a grid search for the variational Kalman filter (VKF) and particle filter (PF) for the σ^2 term in the transition distribution. Since the transition distribution is unknown, a grid search is required. We report the results for the VKF and PF that perform the best on the same validation set, with $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$.

A.5 EXTENDED EXPERIMENTAL RESULTS AND DETAILS

We report additional results, tables, and figures from all three experiments in this section. For the yearbook dataset, we give a complete description of the experiment performed.

A.5.1 TOY NONLINEAR SYSTEM

In the main paper, we report results for 100 Monte Carlo (MC) experiments from the *best* optimizer hyperparameters found using a grid search over 5 separate MC experiments. To give the reader some intuition about good hyperparameters in this system, we show the top 20 performing optimizers from the grid search in Table 4 ($\boldsymbol{Q}_{t-1} = 3, \boldsymbol{R}_t = 2$). 287 configurations are tested in total across Adam (Kingma & Ba, 2015), RMSprop (Tieleman & Hinton, 2012), Adadelta (Zeiler, 2012), Adagrad (Duchi et al., 2011), and gradient descent. Of those 287 configurations, the top 20 performing configurations are exclusively between Adam and RMSprop.

Table 4: Top 20 optimizers found in grid search for the toy nonlinear system ($\boldsymbol{Q}_{t-1} = 3, \boldsymbol{R}_t = 2$). The RMSprop and Adam configurations reported in the main paper are shown in **bold**. K denotes the number of steps used at every time step. η is the learning rate. γ, β_1, β_2 are the decay terms specific to each optimizer.

Method	RMSE
RMSprop ($K = 5, \eta = 1.0, \gamma = 0.5$)	6.391 ± 0.230
RMSprop ($K = 50, \eta = 0.1, \gamma = 0.9$)	6.380 ± 0.230
RMSprop ($K = 10, \eta = 0.5, \gamma = 0.1$)	6.293 ± 0.222
RMSprop ($K = 10, \eta = 0.5, \gamma = 0.5$)	6.248 ± 0.240
Adam ($K = 10, \eta = 0.5, \beta_1, \beta_2 = 0.9$)	6.205 ± 0.241
RMSprop ($K = 100, \eta = 0.05, \gamma = 0.9$)	6.201 ± 0.236
Adam ($K = 5, \eta = 1.0, \beta_1, \beta_2 = 0.9$)	6.112 ± 0.239
Adam ($K = 10, \eta = 0.5, \beta_1, \beta_2 = 0.1$)	6.059 ± 0.221
Adam ($K = 25, \eta = 0.1, \beta_1, \beta_2 = 0.9$)	6.027 ± 0.238
RMSprop ($K = 50, \eta = 0.1, \gamma = 0.5$)	6.018 ± 0.223
RMSprop ($K = 100, \eta = 0.05, \gamma = 0.5$)	6.010 ± 0.229
RMSprop ($\boldsymbol{K = 50, \eta = 0.1, \gamma = 0.1}$)	$\boldsymbol{6.000 \pm 0.227}$
RMSprop ($K = 100, \eta = 0.05, \gamma = 0.1$)	5.987 ± 0.225
Adam ($K = 5, \eta = 1.0, \beta_1, \beta_2 = 0.5$)	5.973 ± 0.218
Adam ($K = 50, \eta = 0.05, \beta_1, \beta_2 = 0.9$)	5.963 ± 0.224
Adam ($K = 10, \eta = 0.5, \beta_1, \beta_2 = 0.5$)	5.953 ± 0.219
Adam ($\boldsymbol{K = 50, \eta = 0.1, \beta_1, \beta_2 = 0.1}$)	$\boldsymbol{5.842 \pm 0.231}$
Adam ($K = 100, \eta = 0.05, \beta_1, \beta_2 = 0.1$)	5.830 ± 0.232
Adam ($K = 50, \eta = 0.1, \beta_1, \beta_2 = 0.5$)	5.794 ± 0.216
Adam ($K = 100, \eta = 0.05, \beta_1, \beta_2 = 0.5$)	5.780 ± 0.220

From Table 4, it is clear that there is a diverse set of optimizer hyperparameters that can produce good results. The goal of hyperparameter selection in this context is to strike a balance between overfitting and underfitting the likelihood such that it reflects a similar balance between the prior

Table 5: RMSEs on the toy nonlinear system ($\boldsymbol{R}_t = 1$). Results show the average RMSE over 100 MC simulations with 95% confidence intervals.

Method	**RMSE ($\boldsymbol{Q}_{t-1} = 1$)**	**RMSE ($\boldsymbol{Q}_{t-1} = 3$)**	**RMSE ($\boldsymbol{Q}_{t-1} = 5$)**
EKF	29.188 ± 5.335	38.075 ± 5.564	45.102 ± 3.876
IEKF ($K = 5$)	11.252 ± 0.643	17.071 ± 0.453	19.203 ± 0.494
IMAP (Adadelta)*	30.984 ± 5.688	33.600 ± 8.007	21.210 ± 3.865
IMAP (Gradient Descent)*	5.564 ± 0.224	7.931 ± 0.159	10.142 ± 0.172
IMAP (Adagrad)*	5.181 ± 0.225	6.549 ± 0.223	9.446 ± 0.230
IMAP (RMSprop)*	5.138 ± 0.219	5.966 ± 0.224	8.829 ± 0.264
IMAP (Adam)*	5.109 ± 0.201	5.708 ± 0.239	8.341 ± 0.315
UKF	4.178 ± 0.274	6.572 ± 0.392	12.735 ± 0.616
PF ($n = 1000$)	1.263 ± 0.043	2.406 ± 0.128	4.264 ± 0.161

*Methods where the reported hyperparameters were found via grid search (see Appendix A.4).

Table 6: RMSEs on the toy nonlinear system ($\boldsymbol{R}_t = 3$). Results show the average RMSE over 100 MC simulations with 95% confidence intervals.

Method	**RMSE ($\boldsymbol{Q}_{t-1} = 1$)**	**RMSE ($\boldsymbol{Q}_{t-1} = 3$)**	**RMSE ($\boldsymbol{Q}_{t-1} = 5$)**
EKF	24.019 ± 3.322	34.099 ± 3.706	39.908 ± 4.277
IEKF ($K = 5$)	8.756 ± 0.606	13.860 ± 0.494	17.252 ± 0.510
IMAP (Adadelta)*	37.356 ± 12.733	26.783 ± 6.720	28.075 ± 11.094
IMAP (Gradient Descent)*	5.714 ± 0.251	8.099 ± 0.163	10.082 ± 0.184
IMAP (Adagrad)*	5.781 ± 0.224	6.630 ± 0.213	9.349 ± 0.238
IMAP (RMSprop)*	5.444 ± 0.254	6.120 ± 0.224	8.300 ± 0.408
IMAP (Adam)*	6.045 ± 0.176	5.978 ± 0.209	7.865 ± 0.307
UKF	4.720 ± 0.250	5.665 ± 0.230	8.490 ± 0.334
PF ($n = 1000$)	1.815 ± 0.055	3.153 ± 0.104	4.757 ± 0.152

*Methods where the reported hyperparameters were found via grid search (see Appendix A.4).

and measurement noise in the explicit filtering sense. It is clear that momentum plays a beneficial role here, since gradient descent and the EKF do not perform as well as optimizers with momentum.

In the main paper, we reported results where the true process noise $\boldsymbol{Q}_{t-1} = 1, 3, 5$ and the true measurement noise $\boldsymbol{R}_t = 2$. In Table 5, we show results for $\boldsymbol{Q}_{t-1} = 1, 3, 5$ and $\boldsymbol{R}_t = 1$. In Table 6, we show results for $\boldsymbol{Q}_{t-1} = 1, 3, 5$ and $\boldsymbol{R}_t = 3$. This demonstrates 6 additional configurations of the toy nonlinear system, each showing a similar result. In low to medium process noise settings, the Implicit MAP Filter is comparable to the Unscented Kalman Filter (UKF). In high process noise settings, the UKF is outperformed by the Implicit MAP Filter. The particle filter (PF) performs the best universally. The extended Kalman filter (EKF) and iterated extended Kalman filter (IEKF) diverged under all settings.

In summary, Table 1, Table 5, and Table 6 show that the PF outperforms our Implicit MAP Filter in 9 out of 9 experiments, the UKF outperforms the Implicit MAP Filter in 5 out 9 experiments (with performance being relatively comparable across all 9 experiments), and the EKF/IEKF outperforms the Implicit MAP Filter in 0 out of 9 experiments.

In Figure 3, we take a random experimental seed (for $\boldsymbol{Q}_{t-1} = 3, \boldsymbol{R}_t = 2$) and show the filtering distributions from every other time step produced by a particle filter with $100,000$ particles. On top of the filtering distributions, we show the maximum a posteriori (MAP) estimates produced by Adam with gradient steps $n = 50$, learning rate $\eta = 0.1$, and momentum terms $\beta_1, \beta_2 = 0.1$ (red). On most time steps, our Implicit MAP Filter exactly maximizes the filtering distribution maintained by the particle filter, despite all time steps being non-convex optimization problems. The trials where the Implicit MAP Filter estimates do not correspond to the true MAP most often come on transition trials, where the state is making an aggressive crossing over the barrier in the double-well. These are the trials where the EKF tends to diverge and the UKF equivalently struggles to produce accurate estimates.

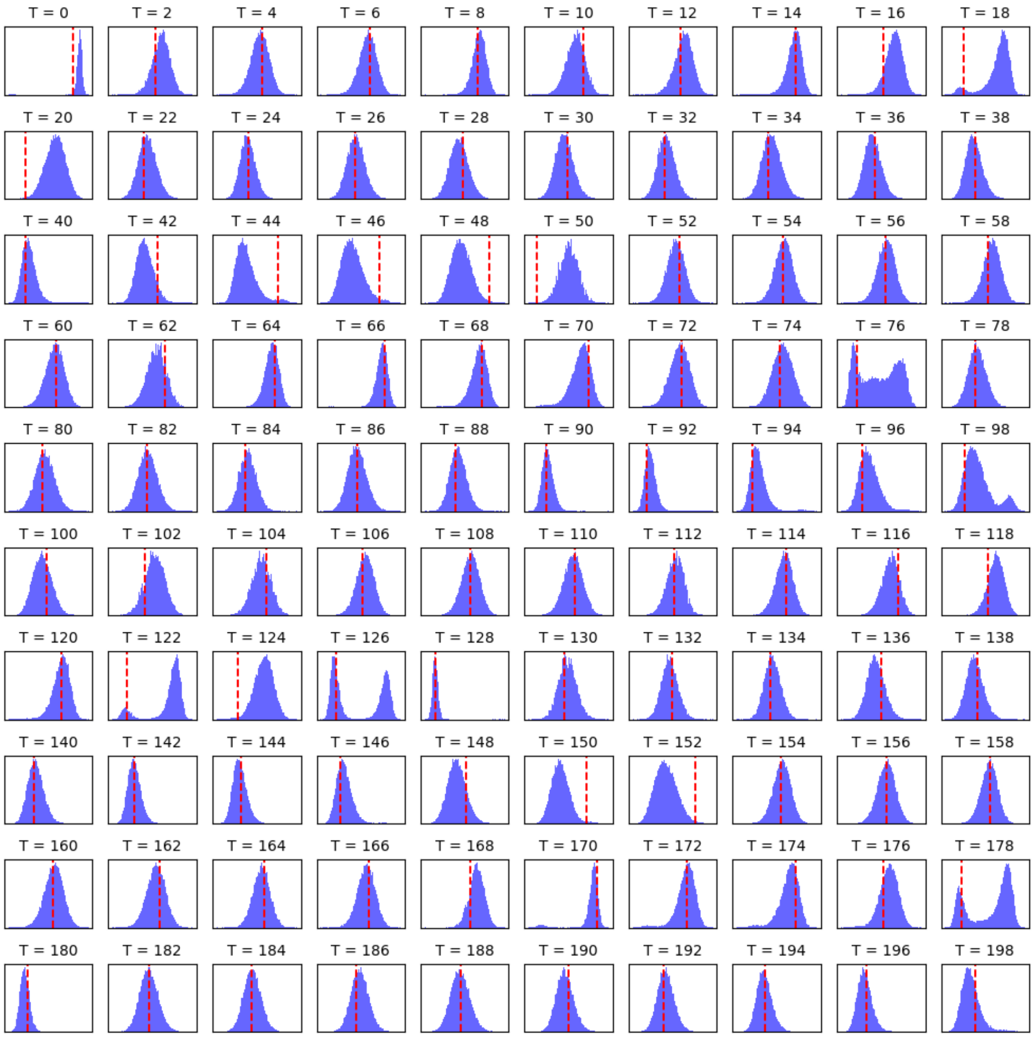
T = 0 T = 2 T = 4 T = 6 T = 8 T = 10 T = 12 T = 14 T = 16 T = 18
T = 20 T = 22 T = 24 T = 26 T = 28 T = 30 T = 32 T = 34 T = 36 T = 38
T = 40 T = 42 T = 44 T = 46 T = 48 T = 50 T = 52 T = 54 T = 56 T = 58
T = 60 T = 62 T = 64 T = 66 T = 68 T = 70 T = 72 T = 74 T = 76 T = 78
T = 80 T = 82 T = 84 T = 86 T = 88 T = 90 T = 92 T = 94 T = 96 T = 98
T = 100 T = 102 T = 104 T = 106 T = 108 T = 110 T = 112 T = 114 T = 116 T = 118
T = 120 T = 122 T = 124 T = 126 T = 128 T = 130 T = 132 T = 134 T = 136 T = 138
T = 140 T = 142 T = 144 T = 146 T = 148 T = 150 T = 152 T = 154 T = 156 T = 158
T = 160 T = 162 T = 164 T = 166 T = 168 T = 170 T = 172 T = 174 T = 176 T = 178
T = 180 T = 182 T = 184 T = 186 T = 188 T = 190 T = 192 T = 194 T = 196 T = 198

Figure 3: Filtering distribution from the particle filter (blue) for 100000 particles and the MAP estimates from the Implicit MAP Filter (red) using a single random experiment ($\boldsymbol{Q}_{t-1} = 3, \boldsymbol{R}_t = 2$).

In Figure 4, we visualize the update steps of both an EKF and an Implicit MAP Filter with an Adam optimizer ($K = 25, \eta = 0.1, \beta_1 = 0.9, \beta_2 = 0.9$). Both of these methods rely on similar gradient information so it is important to understand why the EKF fails catastrophically and diverges but adaptive optimizers, such as Adam, do not suffer the same effect in this system. This figure sheds some light on the situation, showing that the toy nonlinear system can be broken into three different modes: pre-transition, transition, and post-transition. At time step $t = 5$, we are in the pre-transition stage where the objective is approximately convex within the local optimization region. Both Adam and the EKF produce similar estimates in this pre-transition stage. At time steps $t = 12$ and $t = 15$, we are entering the transition phase where the double well shape of the likelihood begins to flatten out. Time step $t = 16$ is the most important step. Here, the update step must cross the barrier at 0 in order to avoid divergence. Adam crosses the barrier at 0 because its incorporation of momentum, which continues to move the estimates despite there being a lack of gradient. The EKF update step does not have momentum. At time step $t = 17$, post-transition, the EKF finds itself on the wrong side of the double well. Since measurements occur only on one side of the double well, as shown at time step $T = 27$, the EKF cannot recover, hence the divergence.

Finally, in Table 7, we show the effect of changing the number of gradient steps for the Adam optimizer we report in the main section, holding all other hyperparameters fixed. The number of gradient steps, K, is the most natural hyperparameter to adjust, and Table 7 demonstrates that the

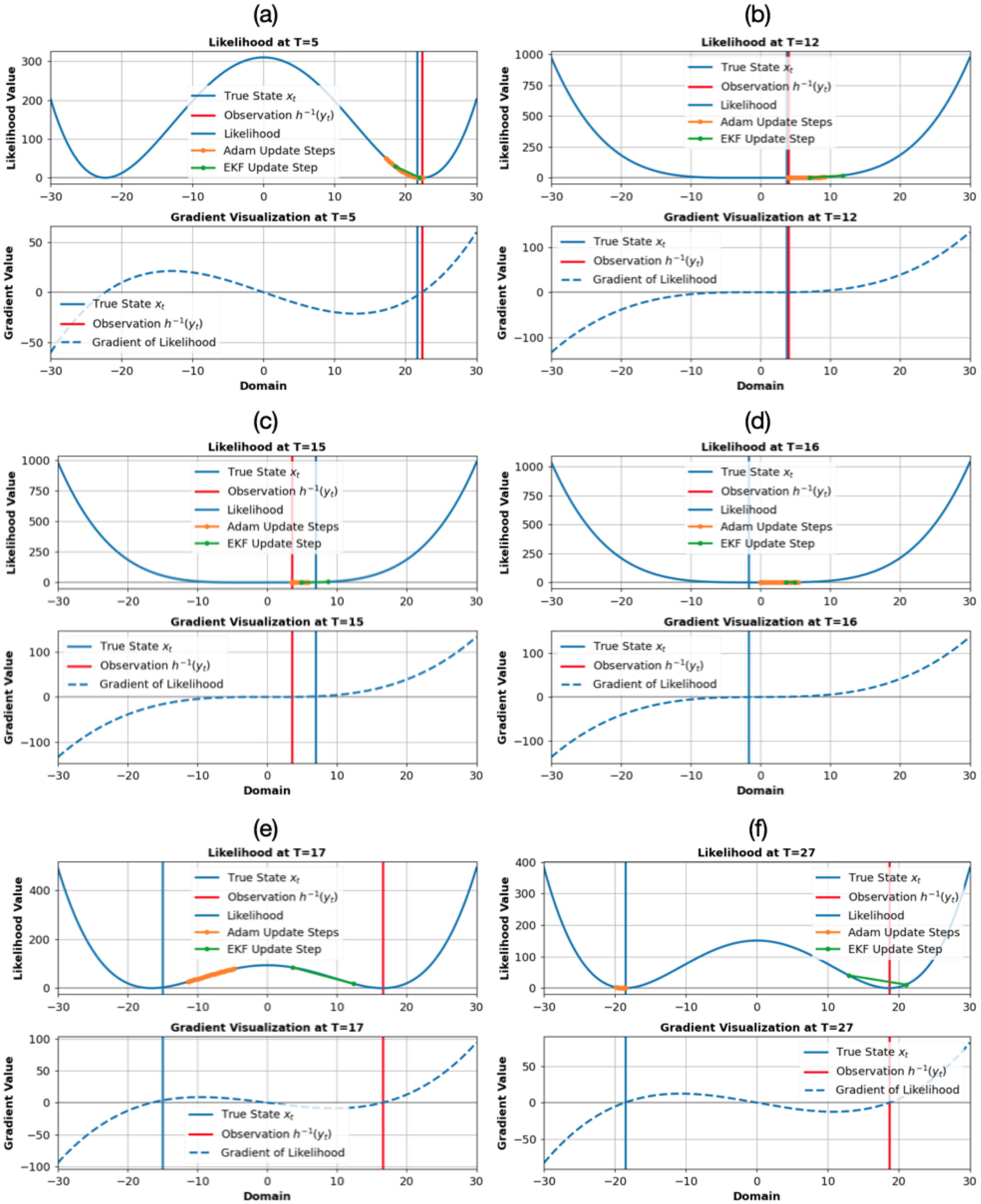

Figure 4: Likelihood and gradient visualization for the toy nonlinear system. (a) pre-transition time steps, (b,c,d) transition time steps. (e, f) post-transition time steps. In (a), the objective function is locally convex and gradient information can be used effectively. The likelihood takes the shape of a double-well. In (b, c, d), gradients approach zero as the double-well flattens out. In (e,f), the double-well picks up again, requiring estimates to be on the correct side of the double-well in order to perform close to optimal. Experiment was run with a random seed and $\boldsymbol{Q}_{t-1} = 3, \boldsymbol{R}_t = 2$. Other random seeds produce nearly identical observations.

specification of K plays a strong role in performance when attempting to implicitly filter in a time-varying setting.

Table 7: The effect of modifying the number of gradient steps using the optimizer configuration reported in the main section for Adam on the toy nonlinear system. Ordered by RMSE.

Method	**RMSE**
Adam ($K = 1, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	10.510 ± 0.263
Adam ($K = 25, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	10.478 ± 0.409
Adam ($K = 3, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	9.749 ± 0.288
Adam ($K = 5, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	9.244 ± 0.266
Adam ($K = 10, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	9.218 ± 0.291
Adam ($K = 100, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	6.575 ± 0.347
Adam ($K = 50, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	5.842 ± 0.231

A.5.2 STOCHASTIC LORENZ ATTRACTOR

Similar to the previous section, we report the top 20 optimizers found from the grid search in Table 8 using the ideal condition (RK4) ($\alpha = 10$). Exactly as before, we tested 287 configurations in total across Adam, RMSprop, Adadelta, Adagrad, and gradient descent. Of those 287 configurations, we achieve a more diverse set of optimizers that performed well compared to the previous system.

Table 8: Top 20 optimizers found in grid search for the stochastic Lorenz attractor ($\alpha = 10$) with respect to the RK4 case. We also show performance of the same optimizer configurations for Euler and GRW cases. RMSEs that outperform the EKF (original $\boldsymbol{Q}_{t-1}$) are shown in **bold**. The same configuration is robust from RK4 → Euler. GRW requires different hyperparameters for the optimizer since the top RK4 optimizers do not correspond to top GRW optimizers.

Method	**RMSE (RK4)**	**RMSE (Euler)**	**RMSE (GRW)**
Adam ($K = 10, \eta = 0.05, \beta_1, \beta_2 = 0.5$)	0.903 ± 0.014	$\mathbf{1.189 \pm 0.025}$	6.728 ± 0.101
RMSprop ($K = 10, \eta = 0.05, \gamma = 0.5$)	0.903 ± 0.112	$\mathbf{1.525 \pm 0.009}$	7.526 ± 0.105
RMSprop ($K = 50, \eta = 0.01, \gamma = 0.1$)	0.900 ± 0.079	$\mathbf{1.470 \pm 0.111}$	7.516 ± 0.105
RMSprop ($K = 50, \eta = 0.01, \gamma = 0.5$)	0.896 ± 0.077	$\mathbf{1.473 \pm 0.111}$	7.518 ± 0.105
Adam ($K = 25, \eta = 0.01, \beta_1, \beta_2 = 0.9$)	0.895 ± 0.015	$\mathbf{1.221 \pm 0.026}$	8.335 ± 0.108
Adam ($K = 5, \eta = 0.1, \beta_1, \beta_2 = 0.5$)	0.893 ± 0.014	$\mathbf{1.219 \pm 0.026}$	6.896 ± 0.102
RMSprop ($K = 5, \eta = 0.1, \gamma = 0.5$)	0.893 ± 0.102	$\mathbf{1.118 \pm 0.015}$	7.531 ± 0.105
RMSprop ($K = 10, \eta = 0.05, \gamma = 0.1$)	0.890 ± 0.064	$\mathbf{1.472 \pm 0.112}$	7.517 ± 0.105
Gradient Descent ($K = 10, \eta = 0.01$)	$\mathbf{0.888 \pm 0.090}$	$\mathbf{2.286 \pm 0.148}$	5.962 ± 0.088
Adam ($K = 5, \eta = 0.1, \beta_1, \beta_2 = 0.1$)	$\mathbf{0.882 \pm 0.049}$	$\mathbf{1.416 \pm 0.105}$	7.411 ± 0.104
RMSprop ($K = 10, \eta = 0.05, \gamma = 0.9$)	$\mathbf{0.881 \pm 0.114}$	$\mathbf{1.540 \pm 0.123}$	7.554 ± 0.104
RMSprop ($K = 5, \eta = 0.1, \gamma = 0.9$)	$\mathbf{0.881 \pm 0.113}$	$\mathbf{1.545 \pm 0.124}$	7.555 ± 0.104
Adam ($K = 10, \eta = 0.05, \beta_1, \beta_2 = 0.1$)	$\mathbf{0.877 \pm 0.039}$	$\mathbf{1.417 \pm 0.122}$	7.395 ± 0.104
Adam ($K = 50, \eta = 0.01, \beta_1, \beta_2 = 0.1$)	$\mathbf{0.876 \pm 0.037}$	$\mathbf{1.391 \pm 0.083}$	7.384 ± 0.104
Gradient Descent ($K = 1, \eta = 0.1$)	$\mathbf{0.849 \pm 0.077}$	$\mathbf{2.114 \pm 0.125}$	5.852 ± 0.086
Adagrad ($K = 50, \eta = 0.05$)	$\mathbf{0.846 \pm 0.014}$	$\mathbf{1.213 \pm 0.027}$	6.905 ± 0.102
Gradient Descent ($K = 3, \eta = 0.1$)	$\mathbf{0.799 \pm 0.009}$	$\mathbf{0.960 \pm 0.012}$	$\mathbf{3.061 \pm 0.038}$
Gradient Descent ($K = 5, \eta = 0.05$)	$\mathbf{0.743 \pm 0.010}$	$\mathbf{0.996 \pm 0.014}$	3.575 ± 0.046
Gradient Descent ($K = 25, \eta = 0.01$)	$\mathbf{0.738 \pm 0.010}$	$\mathbf{1.002 \pm 0.015}$	3.627 ± 0.047
Gradient Descent ($K = 3, \eta = 0.05$)	$\mathbf{0.701 \pm 0.018}$	$\mathbf{1.316 \pm 0.027}$	4.911 ± 0.068

It is clear that the same optimizers perform well when the accuracy of the numerical integration degrades. The number of optimizers that outperform the EKF (original $\boldsymbol{Q}_{t-1}$) increases as we move from 4th order Runge-Kutta (RK4) to Euler's Method. This reflects a robustness property that is desirable in this implicit formulation. When numerical integration is completely removed, as it is for the Gaussian random walk, most of the optimizers that work well with RK4 do not work well here, demonstrating that the Implicit MAP Filter is not perfectly robust to dynamics misspecification. This result is sensible, as the transition distribution itself changed, not just the numerical approximation of it.

In Figure 5, we show the fitted trajectories for 8 different stochastic Lorenz attractor experiments using gradient descent with 3 steps and learning rate $\eta = 0.05$. Two things are important to note: (i) the stochastic Lorenz attractor has extreme variation in the trajectories it produces and (ii) gradient

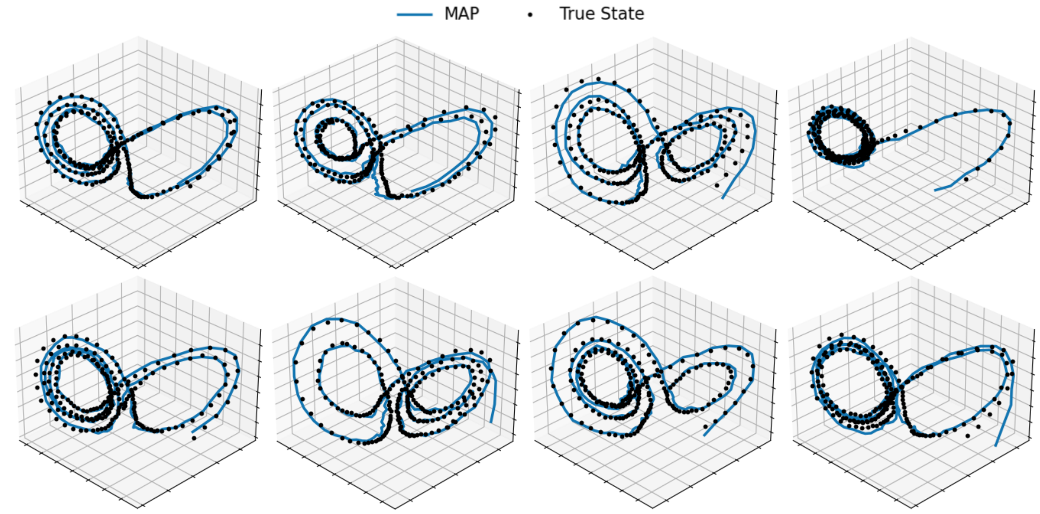

Figure 5: Eight random trajectories for the stochastic Lorenz attractor with 4th order Runge-Kutta and the filter estimates for IMAP with gradient descent. 3 steps with learning rate $\eta = 0.05$.

descent does a remarkably good job of fitting this system despite the stochastic differential equation being chaotic. The extreme variation in the trajectories makes every random seed a different filtering problem.

Table 9: The effect of modifying the number of gradient steps using the optimizer configuration reported in the main section for Gradient Descent on the stochastic Lorenz attractor (RK4). Ordered by RMSE.

Method	**RMSE**
Gradient Descent ($K = 100, \eta = 0.05$)	1.985 ± 0.011
Gradient Descent ($K = 1, \eta = 0.05$)	1.937 ± 0.282
Gradient Descent ($K = 50, \eta = 0.05$)	1.847 ± 0.010
Gradient Descent ($K = 25, \eta = 0.05$)	1.493 ± 0.009
Gradient Descent ($K = 10, \eta = 0.05$)	0.987 ± 0.008
Gradient Descent ($K = 5, \eta = 0.05$)	0.743 ± 0.010
Gradient Descent ($K = 3, \eta = 0.05$)	0.701 ± 0.018

Table 10: RMSEs on the stochastic Lorenz attractor ($\alpha = 1$). Results show the average RMSE over 100 MC simulations with 95% confidence intervals. RK4 indicates 4th order Runge-Kutta method. Euler indicates Euler's method. GRW indicates a Gaussian random walk.

Method	**RMSE (RK4)**	**RMSE (Euler)**	**RMSE (GRW)**
EKF (original $\boldsymbol{Q}_{t-1}$)	4.948 ± 0.600	8.940 ± 0.310	5.953 ± 0.041
EKF (optimized $\boldsymbol{Q}_{t-1}$)*	$\mathbf{0.621 \pm 0.028}$	$\mathbf{0.939 \pm 0.011}$	$\mathbf{1.568 \pm 0.011}$
IEKF ($K = 5$)	4.948 ± 0.600	8.940 ± 0.310	5.953 ± 0.041
IMAP (Adadelta)*	6.480 ± 0.450	7.713 ± 0.441	10.774 ± 0.216
IMAP (Adam)*	0.845 ± 0.035	1.168 ± 0.014	1.897 ± 0.011
IMAP (Adagrad)*	0.824 ± 0.067	1.098 ± 0.017	1.797 ± 0.010
IMAP (RMSprop)*	0.823 ± 0.082	1.076 ± 0.014	1.752 ± 0.015
IMAP (Gradient Descent)*	$\mathbf{0.626 \pm 0.031}$	$\mathbf{0.947 \pm 0.011}$	$\mathbf{1.568 \pm 0.011}$
UKF (original $\boldsymbol{Q}_{t-1}$)	5.834 ± 0.251	6.232 ± 0.255	9.398 ± 0.183
UKF (optimized $\boldsymbol{Q}_{t-1}$)*	1.396 ± 0.011	1.410 ± 0.011	1.738 ± 0.012
PF ($n = 1000$)	1.461 ± 0.021	1.725 ± 0.031	15.280 ± 0.232

*Methods where the reported hyperparameters were found via grid search (see Appendix A.4).

Table 11: RMSEs on the stochastic Lorenz attractor ($\alpha = 5$). Results show the average RMSE over 100 MC simulations with 95% confidence intervals. RK4 indicates 4th order Runge-Kutta method. Euler indicates Euler's method. GRW indicates a Gaussian random walk.

Method	**RMSE (RK4)**	**RMSE (Euler)**	**RMSE (GRW)**
EKF (original $\boldsymbol{Q}_{t-1}$)	1.510 ± 0.305	5.430 ± 0.302	3.954 ± 0.032
EKF (optimized $\boldsymbol{Q}_{t-1}$)*	$\mathbf{0.651 \pm 0.025}$	$\mathbf{0.938 \pm 0.012}$	$\mathbf{1.564 \pm 0.010}$
IEKF ($K = 5$)	1.510 ± 0.305	5.430 ± 0.302	3.954 ± 0.032
IMAP (Adadelta)*	6.212 ± 0.424	7.560 ± 0.477	10.736 ± 0.232
IMAP (Adam)*	0.870 ± 0.015	1.163 ± 0.021	1.899 ± 0.011
IMAP (Adagrad)*	0.843 ± 0.114	1.126 ± 0.011	1.797 ± 0.010
IMAP (RMSprop)*	0.835 ± 0.119	1.076 ± 0.016	1.755 ± 0.0165
IMAP (Gradient Descent)*	$\mathbf{0.665 \pm 0.045}$	$\mathbf{0.947 \pm 0.012}$	$\mathbf{1.566 \pm 0.010}$
UKF (original $\boldsymbol{Q}_{t-1}$)	4.066 ± 0.124	4.208 ± 0.127	7.106 ± 0.083
UKF (optimized $\boldsymbol{Q}_{t-1}$)*	1.396 ± 0.010	1.410 ± 0.010	1.736 ± 0.012
PF ($n = 1000$)	1.531 ± 0.022	1.877 ± 0.058	14.690 ± 0.312

*Methods where the reported hyperparameters were found via grid search (see Appendix A.4).

Table 12: RMSEs on the stochastic Lorenz attractor ($\alpha = 20$). Results show the average RMSE over 100 MC simulations with 95% confidence intervals. RK4 indicates 4th order Runge-Kutta method. Euler indicates Euler's method. GRW indicates a Gaussian random walk.

Method	**RMSE (RK4)**	**RMSE (Euler)**	**RMSE (GRW)**
EKF (original $\boldsymbol{Q}_{t-1}$)	0.852 ± 0.016	1.193 ± 0.027	2.294 ± 0.045
EKF (optimized $\boldsymbol{Q}_{t-1}$)*	$\mathbf{0.838 \pm 0.012}$	$\mathbf{1.000 \pm 0.013}$	$\mathbf{1.558 \pm 0.012}$
IEKF ($K = 5$)	0.852 ± 0.016	1.193 ± 0.027	2.294 ± 0.045
IMAP (Adadelta)*	5.795 ± 0.421	7.244 ± 0.468	10.490 ± 0.288
IMAP (Adam)*	0.987 ± 0.028	1.183 ± 0.015	1.889 ± 0.010
IMAP (Adagrad)*	0.981 ± 0.011	1.148 ± 0.014	1.813 ± 0.019
IMAP (RMSprop)*	0.991 ± 0.011	1.127 ± 0.017	1.781 ± 0.016
IMAP (Gradient Descent)*	$\mathbf{0.841 \pm 0.011}$	$\mathbf{1.013 \pm 0.015}$	$\mathbf{1.587 \pm 0.010}$
UKF (original $\boldsymbol{Q}_{t-1}$)	2.000 ± 0.034	2.049 ± 0.038	4.293 ± 0.092
UKF (optimized $\boldsymbol{Q}_{t-1}$)*	1.428 ± 0.011	1.445 ± 0.012	1.734 ± 0.014
PF ($n = 1000$)	1.612 ± 0.033	1.698 ± 0.034	13.535 ± 0.474

*Methods where the reported hyperparameters were found via grid search (see Appendix A.4).

In Table 9, we show the affect of modifying the number of gradient steps. In Table 10, Table 11, and Table 12 we show additional results for the stochastic Lorenz attractor for $\alpha = 1, 5, 20$, respectively. These additional results show both the original EKFs and UKFs and the EKFs and UKFs with $\boldsymbol{Q}_{t-1}$ optimized by grid search. For the optimized EKFs and UKFs, we test 500 $\boldsymbol{Q}_{t-1}$ matrices and report the best one. For the Implicit MAP Filters, we test 287 in total (35 Implicit MAP Filters are gradient descent). Again, we use 5 separate Monte Carlo simulations to select the Implicit MAP Filters to report.

The results are consistent with Table 2 in Section 5.2. In all cases the EKF with optimized $\boldsymbol{Q}_{t-1}$ is statistically equivalent in performance to the Implicit MAP Filter with gradient descent. Notably, the EKF has access to a larger search space. Further, the stochastic Lorenz attractor is relatively convex, which suits the EKF, but the EKF diverges in highly nonlinear systems. The Implicit MAP Filter does not have this issue, working well in both Section 5.1 and Section 5.2.

A.5.3 YEARBOOK

The yearbook dataset consists of 37,921 frontal-facing American high school yearbook photos from 1905 - 2013 from 128 high schools in 27 states. Each image is resized to a $32 \times 32 \times 1$ grayscale image and paired with a binary label y, representing the student's gender. The years 1905 - 1930 only have 20 years with images available and years with as little as one photo available, making them not ideal for training and testing. Thus, we use these 20 years as a single pre-training dataset of 886 images.

For all five methods we test, we use a 4-layer convolutional neural network (CNN). Each convolutional layer has a kernel size of 3×3, stride of 1×1, same padding, 32 output channels, ReLU activation, and a 2D max pool layer with kernel size 2×2. We use stochastic gradient descent (SGD) with a fixed learning rate of 10^{-3}, the pre-training dataset of 886 images, and a batch size of 64 to train the initial network for 200 steps. Since the results we report are the average over 10 random seeds, we pre-train the initial network 10 different times in exactly this fashion. This is supposed to resemble samples from some approximate initial starting distribution over neural network weights.

From 1931 - 2010, we take 32 randomly chosen images as a training set and 100 randomly chosen images as a test set at every time step. From 1931 - 1970, we take an additional 16 images as a validation set. In the main paper, we report test accuracy of the configurations with the best average validation performance from 1931 - 1970. The years 1941 and 2006 only had 93 and 70 images available, respectively. Thus, 1941 has a test set size of 45 and 2006 has a test set size of 38. All other training sets, validation sets, and test sets are exactly as described.

For the static weights approach, we do not do any additional training after the 200 steps of SGD over the pre-training dataset. We report the accuracy of those weights from 1931 - 2010 without any adaptation.

The direct fit approach uses 1000 full batch steps of the Adam optimizer at a fixed learning rate of 10^{-3} at every time step. This resembles what a practitioners might do to fit the training data. Instead of re-training from scratch, they would likely use the weight initialization from the previous time step. This inherently resembles the Implicit MAP filtering approach we propose here, but we are misspecifying the number of gradient steps by using 1000. 1000 gradient steps, or optimization to near convergence, assumes that the measurement noise is close to $\mathbf{0}$ at every time step. By using this overfitting Implicit MAP Filter, which we call direct fit, we are trying to show why it is important to appropriately define the *implicit* filtering equations. It is naive to not consider the number of gradient steps as an important hyperparameter when working with time-varying objectives.

For the particle filter (PF) and variational Kalman filter (VKF), we explicitly specify a transition distribution $p(\mathbf{w}_t|\mathbf{w}_{t-1}) = \mathcal{N}(\mathbf{w}_t|\mathbf{w}_{t-1}, \sigma^2 \boldsymbol{I})$ where $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. We run both of these methods exactly as described in Appendix A.1.

Finally, we test five configurations of Adam for the Implicit MAP Filter where all hyperparameters were set to standard settings ($\eta = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$) and the number of steps K was selected from the set $\{1, 10, 25, 50, 100\}$. We use the validation set from the first 40 years to select the number of steps to report in the main paper.

Figures 6, 7, and 8 show the classification accuracy of every IMAP filter, PF, and VKF, respectively. For Adam, performance was maximized by $K = 50$ and $K = 100$. The static weights case can be seen as an Implicit MAP Filter with $\mathbf{0}$ process noise over a transition function that is the identity function. This approach is a clear misspecification.

The particle filters were generally only marginally better than the static weights case. This is due to a poorly defined proposal distribution, since it is hard to imagine that we could find the true transition distribution via grid search.

The VKF saw better performance than the PF for nearly all configurations tested. This is due to the fact that we are explicitly optimizing the network weights. However, by explicitly regularizing the network instead of implicitly regularizing via early stopping, we do not match the performance of the Implicit MAP Filter or even the direct fit approach. This is likely due to the fact that the transition distribution is far from correctly specified. Again, we cannot imagine that we could find the correct transition distribution via grid search.

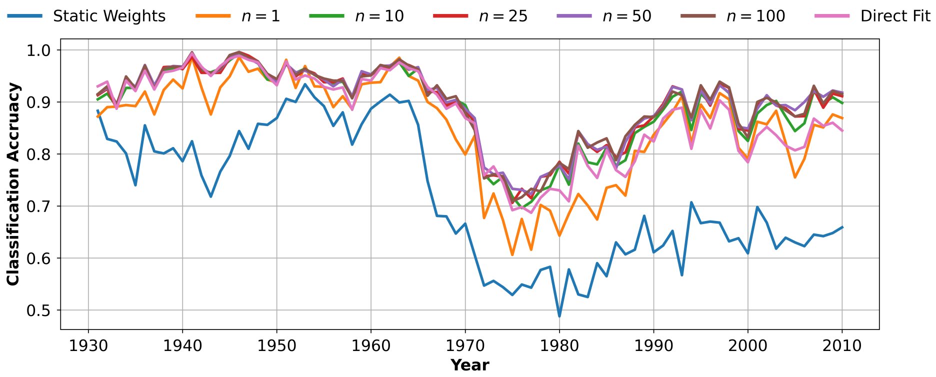

Figure 6: Yearbook dataset filtering results using standard Adam hyperparameters over number of steps $K \in \{1, 10, 25, 50, 100\}$. We also plot the static weights case and direct fit case as comparative baselines.

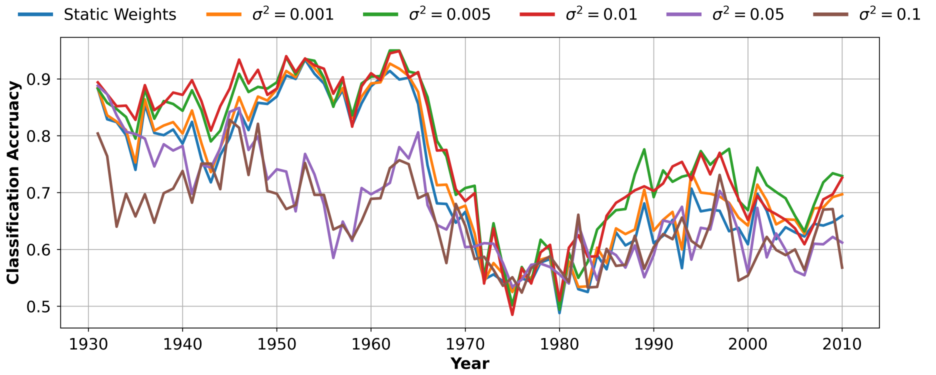

Figure 7: Yearbook dataset filtering results using a particle filter with transition $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. We plot the static weights case as a comparative baselines.

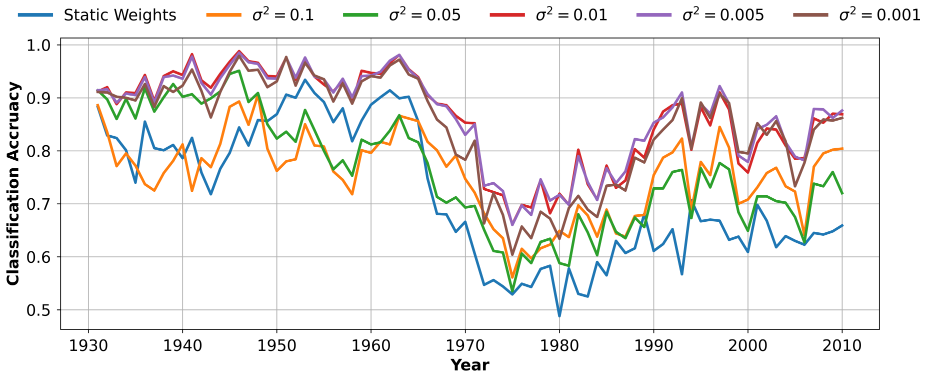

Figure 8: Yearbook dataset filtering results using a variational Kalman filter with transition $\sigma^2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. We plot the static weights case as a comparative baselines.