Predicting chaotic systems with sparse data

lessons from numerical weather prediction

David Kelly, Courant Institute, NYU

A toy atmospheric model

We have a three dimensional variable (x, y, z) evolving in time, governed by a differential equation called the Lorenz model

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$

The model is chaotic: an uncertain estimate of the current state becomes more uncertain over time

A single trajectory

An ensemble of trajectories

Small errors in the initial state will get larger over time

Suppose that we make a partial observation of the 'true' trajectory every τ seconds

$$x(n\tau) + \xi_n \qquad n = 1, 2, \ldots$$

where ξ_n is a source of observational noise.

Can we combine the model and the data to improve the estimate of the current state?

Assimilating the x data

The blue curve is the true trajectory

The dots are an ensemble of state estimates

The state estimates synchronize with the true state

Not just for toy models …

This precise idea is used in numerical weather prediction (any many many more applications)

The NWP model is a family of layered, discretized partial differential equations (PDEs).

The model can be extremely high dimensional (~ 10^9 variables).

Not just for toy models …

Enormous amounts of observational data, that are accurate (small noise) but very sparse (we do not observe the whole ocean, the small scales)

Observational data is not simply ‘observing the x variable’, but complicated functions of many variables (eg. how does this GPS enabled buoy move through the ocean?)

The next 25 slides

1. The math behind data assimilation
2. The linear model scenario
3. The nonlinear model scenario

The math of data assimilation

The model

We are tracking a d-dimensional vector u_n whose motion is governed by the discrete time random dynamical system

$$u_{n+1} = \psi(u_n) + \eta_n$$

where $\eta_n \sim N(0, \Sigma)$ is iid Gaussian noise (zero in the Lorenz example) and ψ is a deterministic map.

The initial state u_0 is unknown exactly, but known to be distributed like a Gaussian $u_0 \sim N(m_0, C_0)$

The data

We make a partial, noisy observation at every time step

$$z_{n+1} = h(u_{n+1}) + \xi_{n+1}$$

where $\xi_{n+1} \sim N(0, \Gamma)$ is idd Gaussian noise and h is the observation function

Bayes formula

The state u_n is a random variable. We would like to know the conditional distribution of u_n given all the data up to time n $Z_n = \{z_1, z_2, \ldots, z_n\}$

By Bayes' formula we have

$$\begin{aligned}
P(u_{n+1}|Z_{n+1}) &= P(u_{n+1}|Z_n, z_{n+1}) \\
&= \frac{P(z_{n+1}|u_{n+1}, Z_n)P(u_{n+1}|Z_n)}{P(z_{n+1}|Z_n)} \\
&= \frac{P(z_{n+1}|u_{n+1})P(u_{n+1}|Z_n)}{P(z_{n+1}|Z_n)}
\end{aligned}$$

The filtering cycle

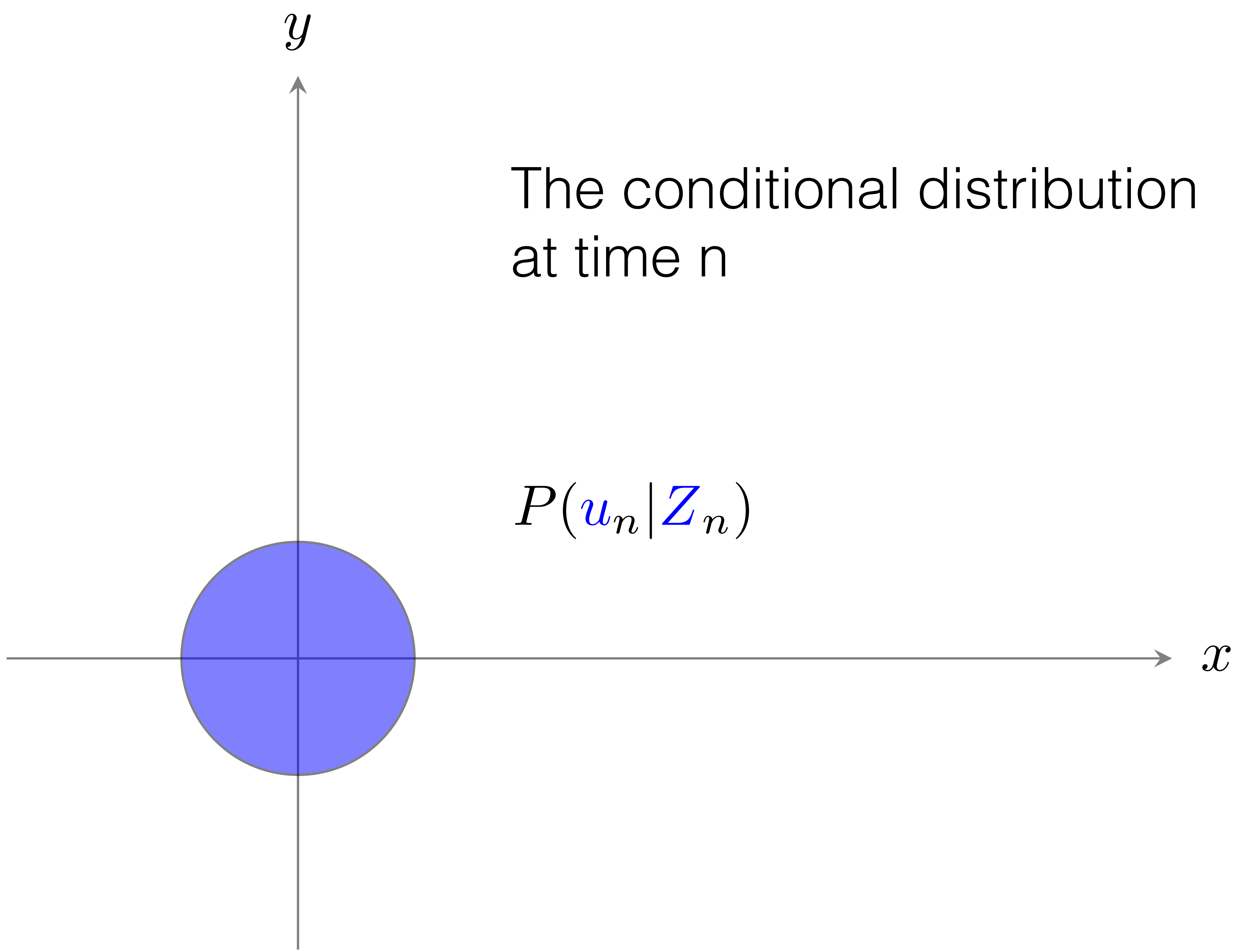
Ignoring the normalization constant …

$$P(u_{n+1}|Z_{n+1}) \propto P(z_{n+1}|u_{n+1})P(u_{n+1}|Z_n)$$

We can use this formula to perform an update procedure:

$$P(u_n|Z_n) \mapsto P(u_{n+1}|Z_{n+1})$$

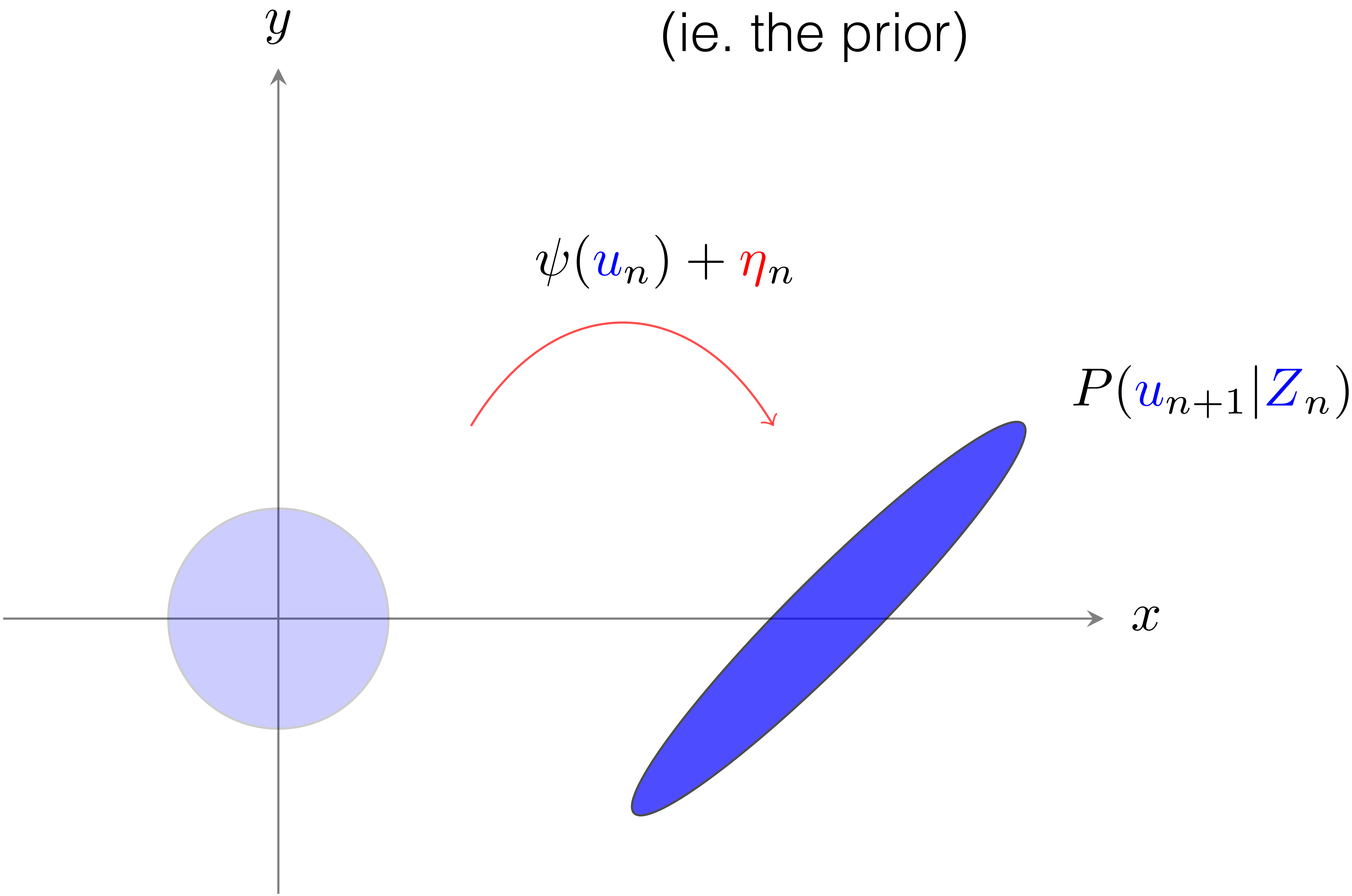
which is called the filtering cycle.



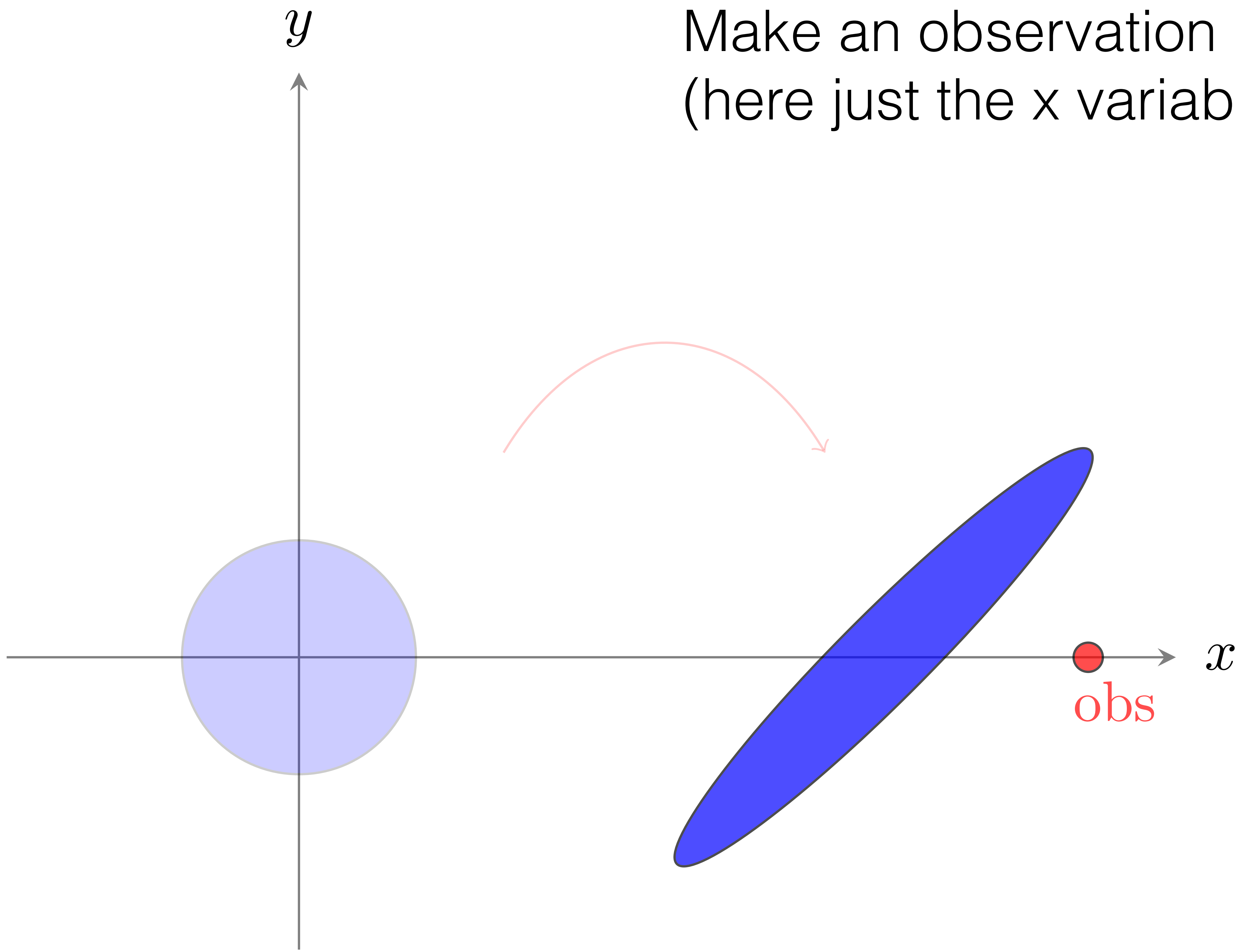
The conditional distribution at time n

$$P(u_n|Z_n)$$

The 'forecast' distribution
(ie. the prior)

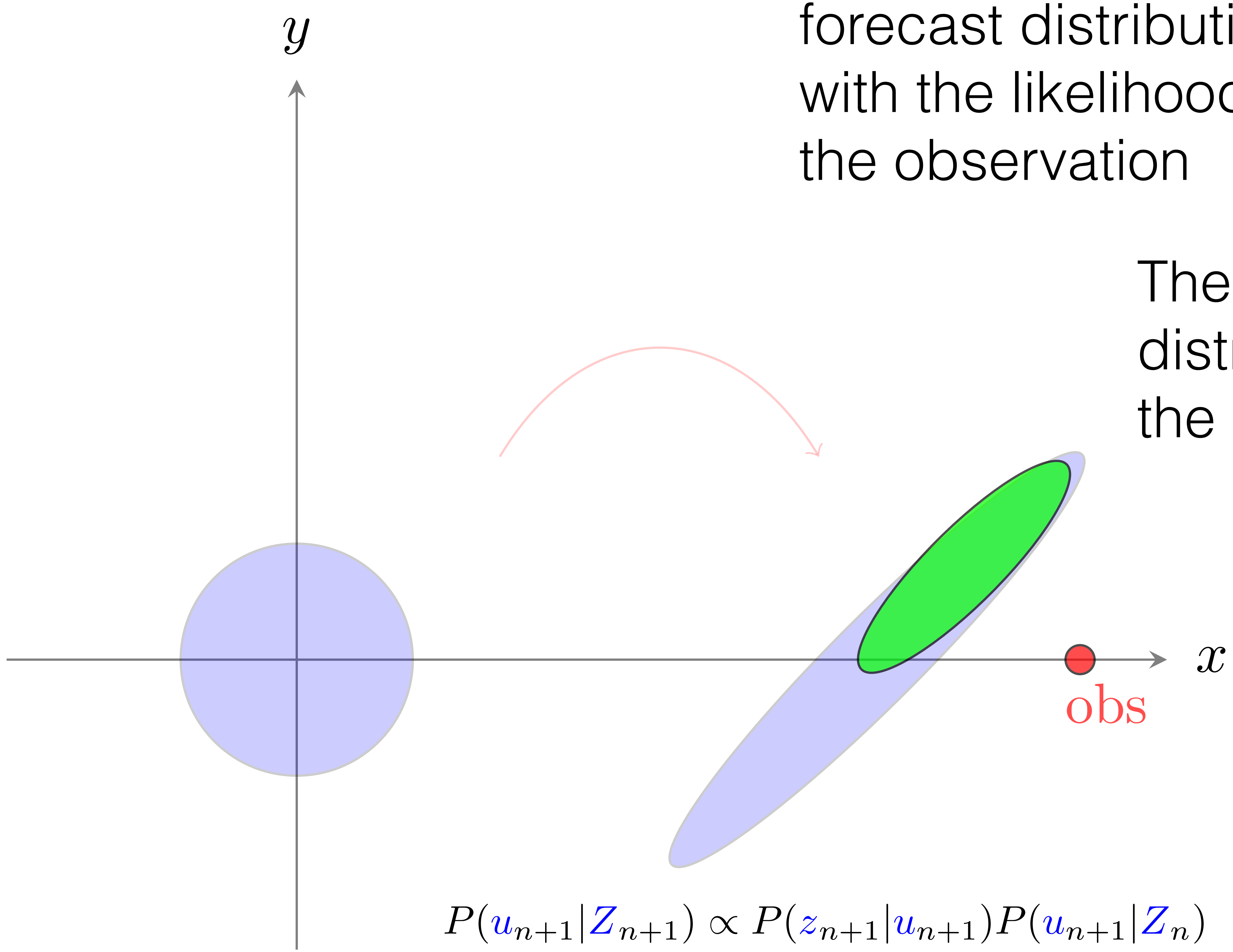


Make an observation
(here just the x variable)



Re-weight the forecast distribution with the likelihood of the observation

The green distribution is the posterior



$$P(u_{n+1}|Z_{n+1}) \propto P(z_{n+1}|u_{n+1})P(u_{n+1}|Z_n)$$

In the linear model scenario, everything has a formula!

The Kalman filter

When the dynamics ψ and the observation function h are both linear, the conditional random variable $u_n | Z_n$ is a Gaussian and is characterized completely by its mean and covariance (m_n, C_n)

The mean and covariance satisfy a recursion formula

$$m_{n+1} = (I - K_{n+1}H)\psi(m_n) + K_{n+1}z_{n+1}$$

$$C_{n+1} = (I - K_{n+1}H)(\psi C_n \psi^T + \Sigma)$$

The Kalman gain K_{n+1} is the correct convex combination of model and data, it is determined by the forecast and data covariances.

What can we do for nonlinear models?

3DVAR algorithm

Obtain a state estimate m_n using the Kalman update formula

$$m_{n+1} = (I - KH)\psi(m_n) + Kz_{n+1}$$

Since the model is nonlinear, distributions are no longer Gaussian and there is no 'correct' choice for the Kalman gain

$$H = (1, 0, 0), K = (1, 1, 1)^T$$

$$m_{n+1} = \left[\begin{array}{c} x((n+1)\tau) + \xi_{n+1} \\ \psi_y(m_n) + (x((n+1)\tau) + \xi_{n+1} - \psi_x(m_n)) \\ \psi_z(m_n) + (x((n+1)\tau) + \xi_{n+1} - \psi_x(m_n) \end{array} \right]$$

The 3DVAR algorithm is accurate (if properly tuned): in that the estimates get closer to the true state when observational noise is small and when enough variables are observed.

When the observations are sparse, we cannot expect accuracy. Instead we would like a set of samples from the posterior.

$$P(u_n|Z_n)$$

The Ensemble Kalman filter (EnKF) uses an ensemble of ‘particles’ to find a state estimate and quantify the uncertainty of the estimate.

The EnKF maintains an ensemble $\{u_n^{(1)}, u_n^{(2)}, \ldots, u_n^{(K)}\}$ to represent the whole posterior and not just the mean

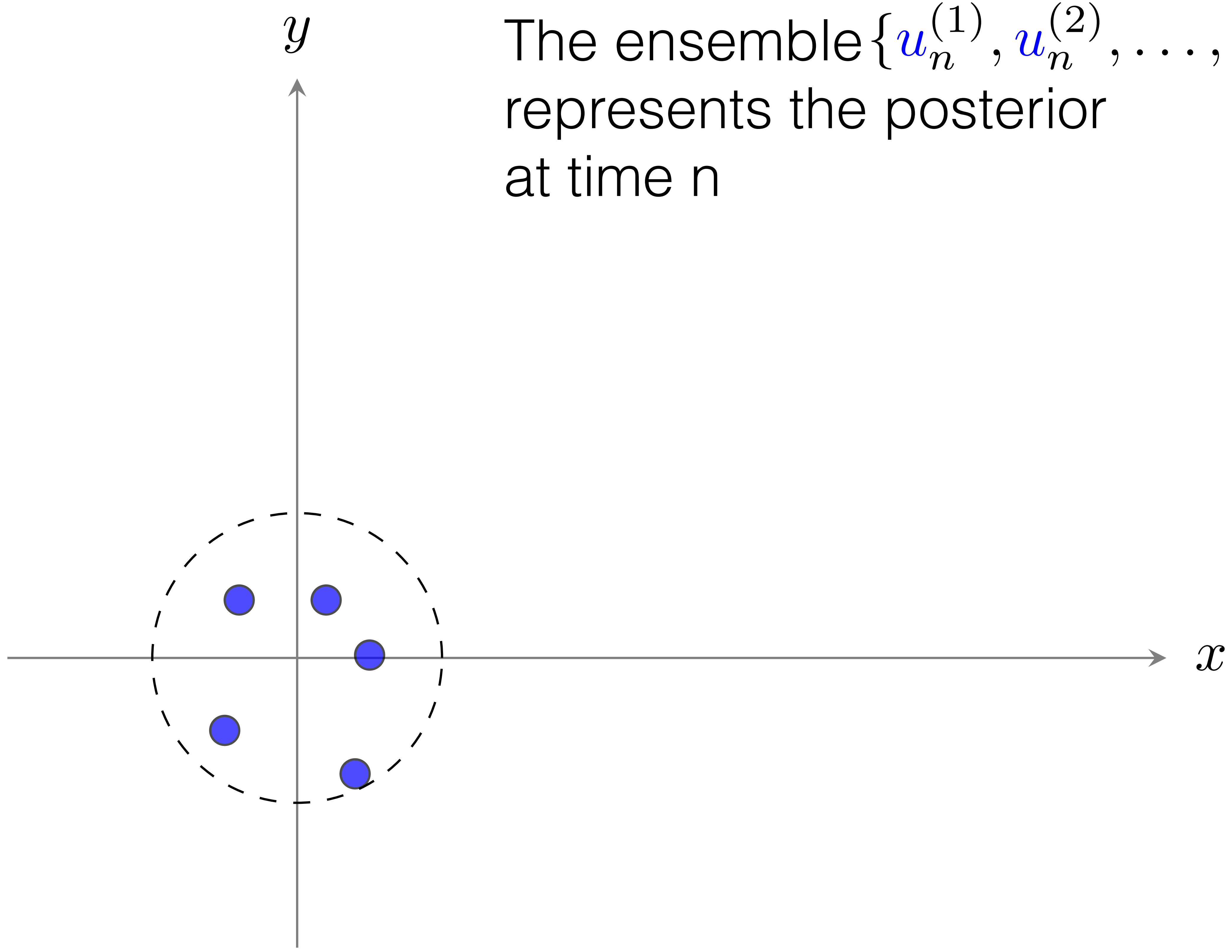
Each particle is updated much like the 3DVAR

$$u_{n+1}^{(k)} = (I - K_{n+1}H)\psi(u_n^{(k)}) + K_{n+1}z_{n+1}$$

But the Kalman gain is chosen using the empirical covariance of the ‘forecast ensemble’

$$\{\psi(u_n^{(1)}), , \ldots, \psi(u_n^{(K)})\}$$

The ensemble $\{u_n^{(1)}, u_n^{(2)}, \ldots, u_n^{(K)}\}$ represents the posterior at time n

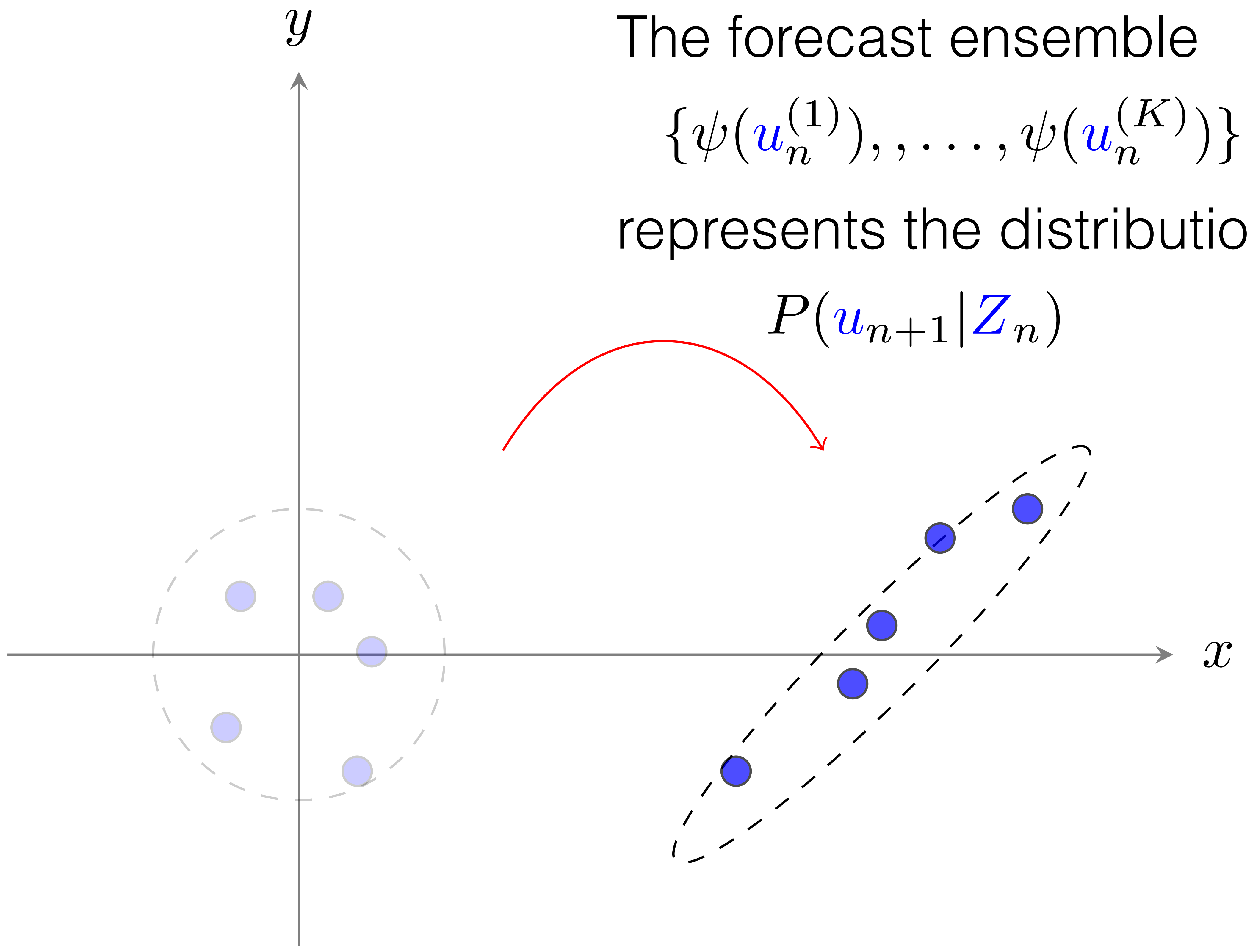


The forecast ensemble

$$\{\psi(u_n^{(1)}), , \ldots, \psi(u_n^{(K)})\}$$

represents the distribution

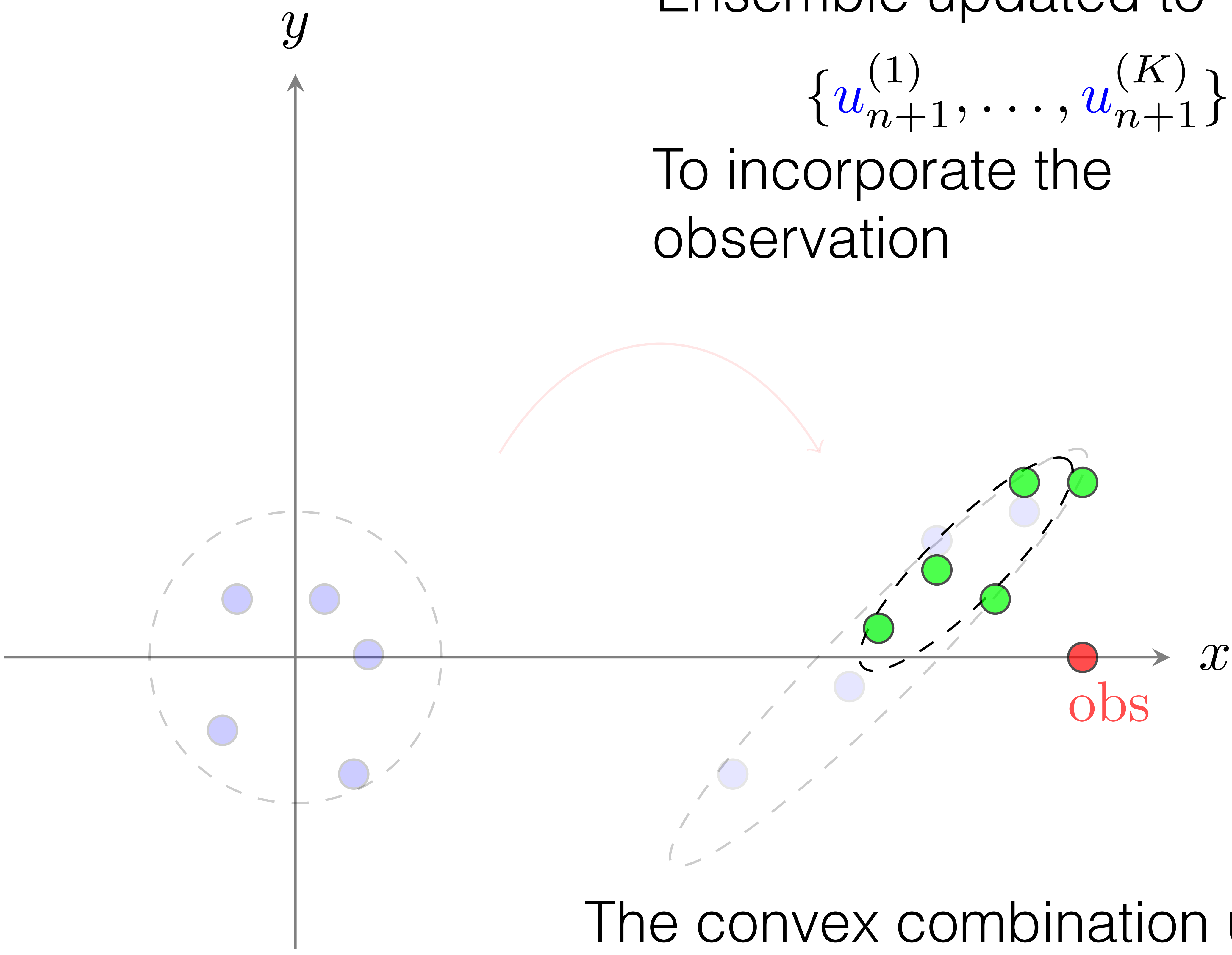
$$P(u_{n+1}|Z_n)$$



Ensemble updated to

$$\{u_{n+1}^{(1)}, \ldots, u_{n+1}^{(K)}\}$$

To incorporate the observation



The convex combination uses the forecast covariance

The Canadian Weather Bureau uses EnKF for operational NWP, with around 100 particles for a ~ 10^9 dimensional state variable (!!!)

EnKF works surprisingly well with high dimensional models that are effectively lower dimensional.

The covariance of the forecast ensemble

$$\{\psi(u_n^{(1)}), , \dots, \psi(u_n^{(K)})\}$$

only represents the directions of highest model variation. Often much smaller dimension than state.

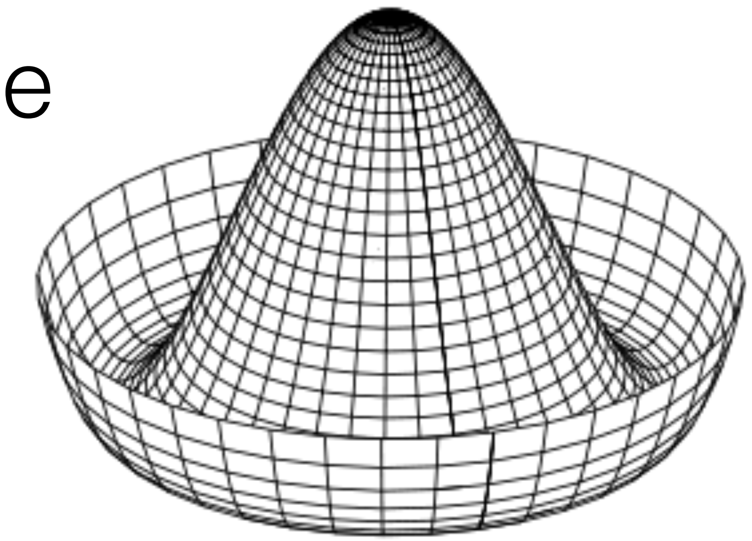
EnKF and the Sombrero SDE

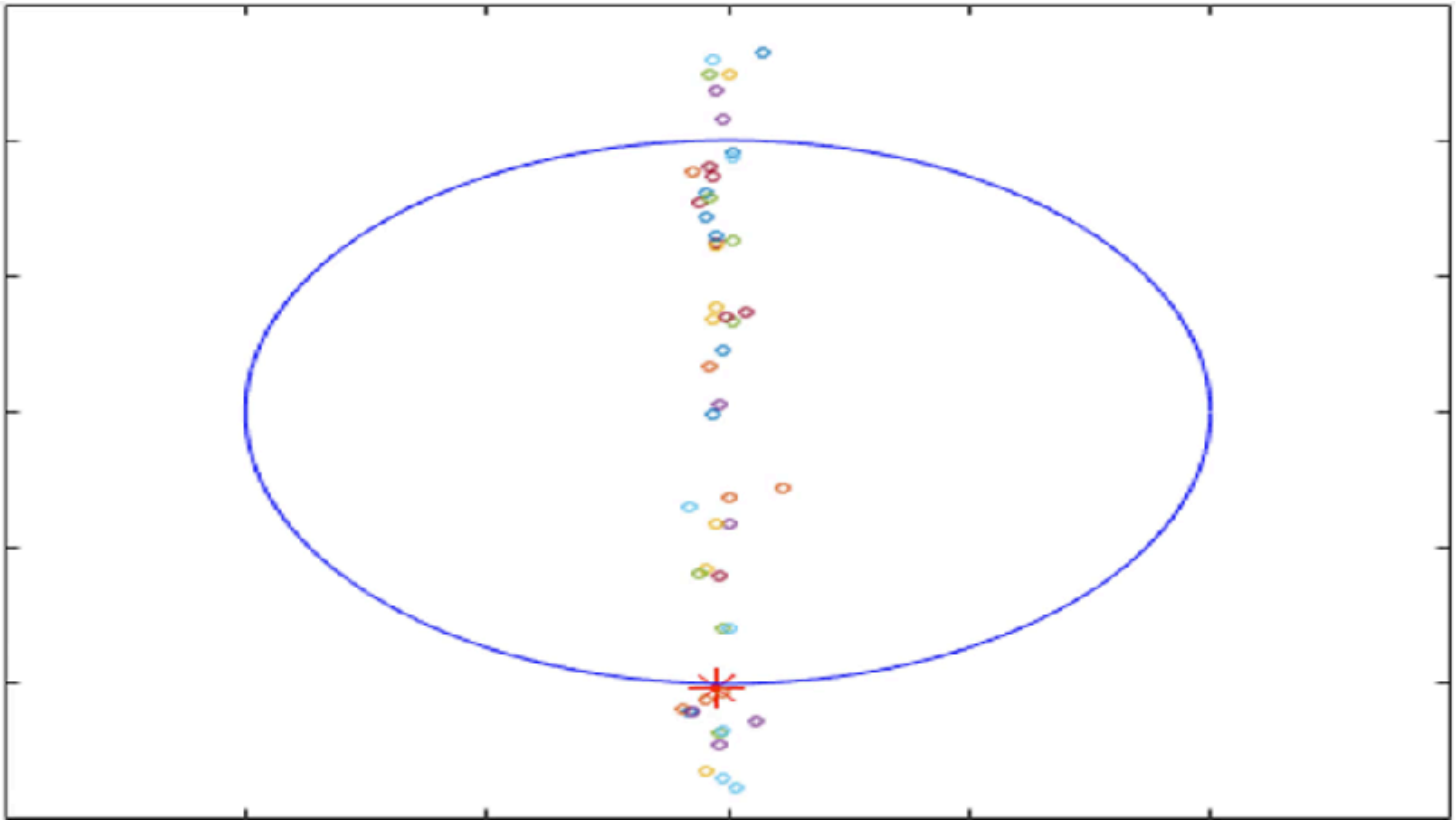
Consider the two dimensional stochastic differential equation u = *(x,y)*

$$du = -\nabla V(u)dt + \sigma dW$$

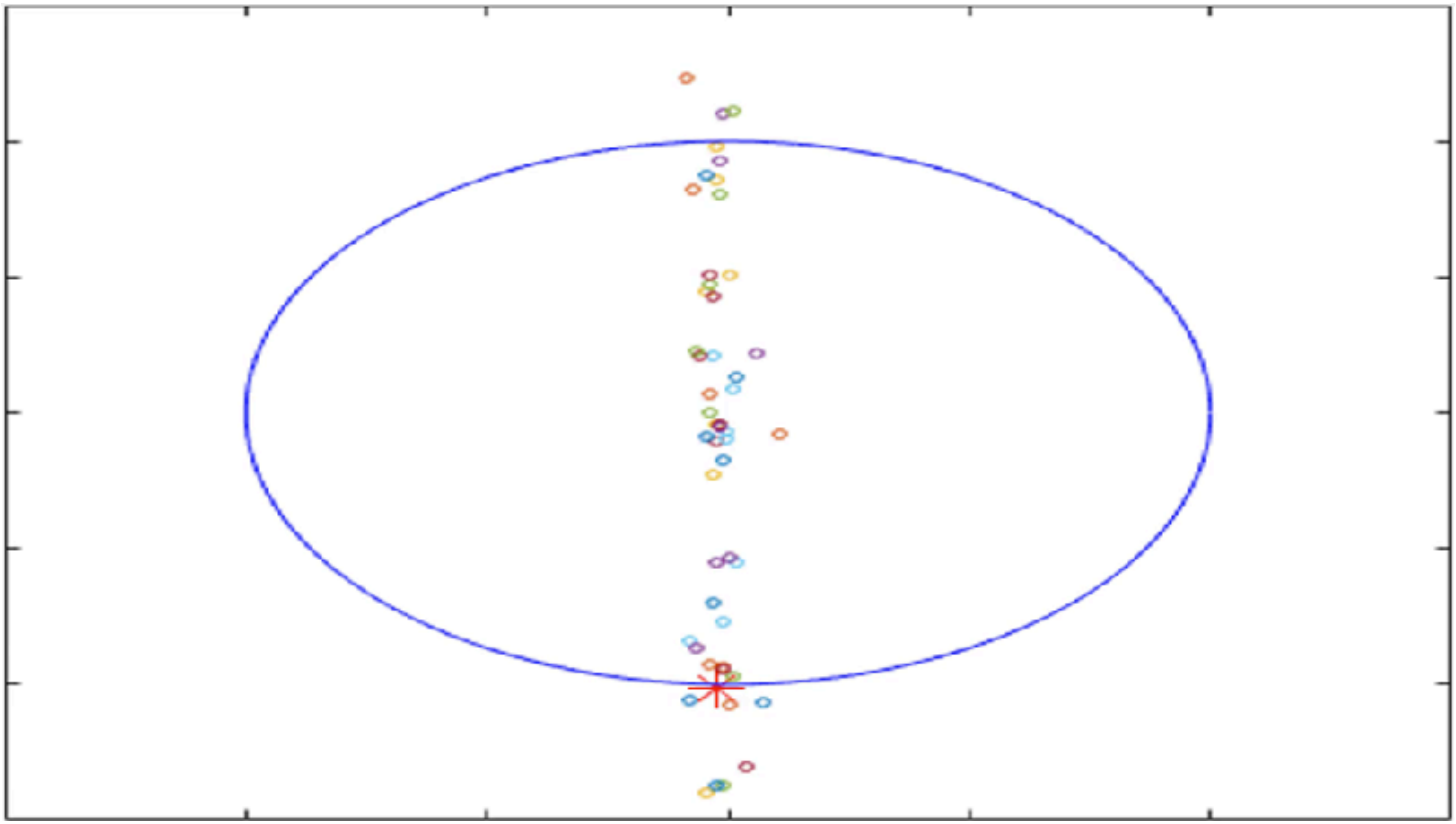
with the 'Sombrero potential' $V(x,y) = (1 - x^2 - y^2)^2$

And we only observe the x variable

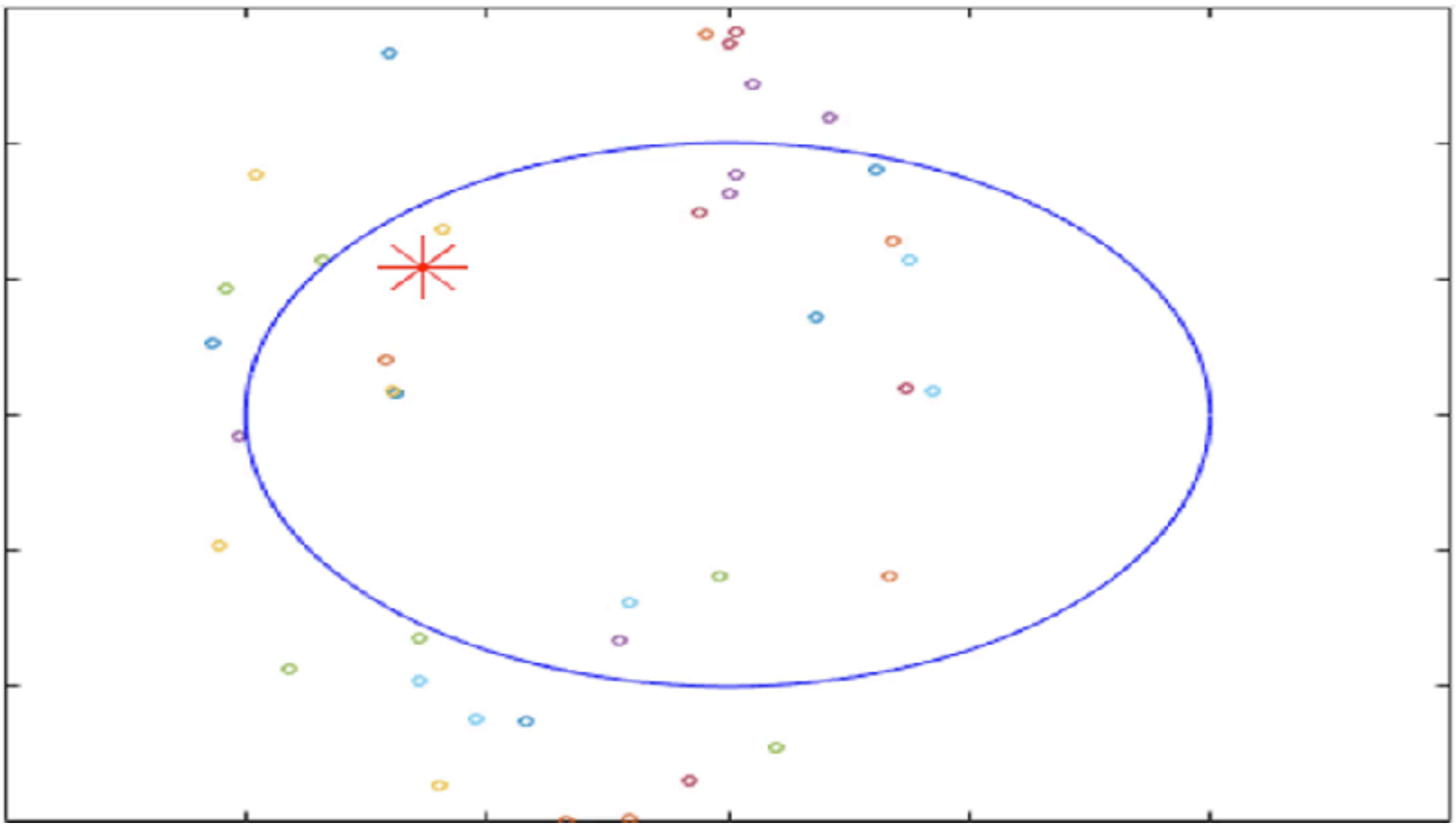




EnKF - only observing x, big red star is true state



EnKF - exact same truth, different model noise



Particle filter - sampling from the posterior, not just tracking the truth

Pros and cons

Kalman filter - very efficient, but requires linearity. Can be expensive in high dimensions.

3DVAR - very efficient, requires tuning and has no uncertainty quantification

EnKF - very efficient, works in high dimensions, provides UQ but not for non-Gaussians

Particle filter - samples from the posterior, but very inefficient in high dimensions.

Thank you for listening!

Slides are on my website dtbkelly.com