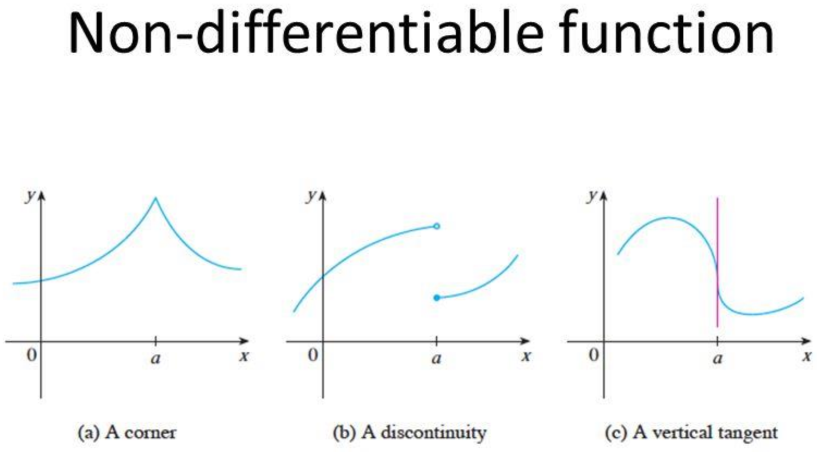
# CSCE-642 Reinforcement Learning
# Derivative Free Methods

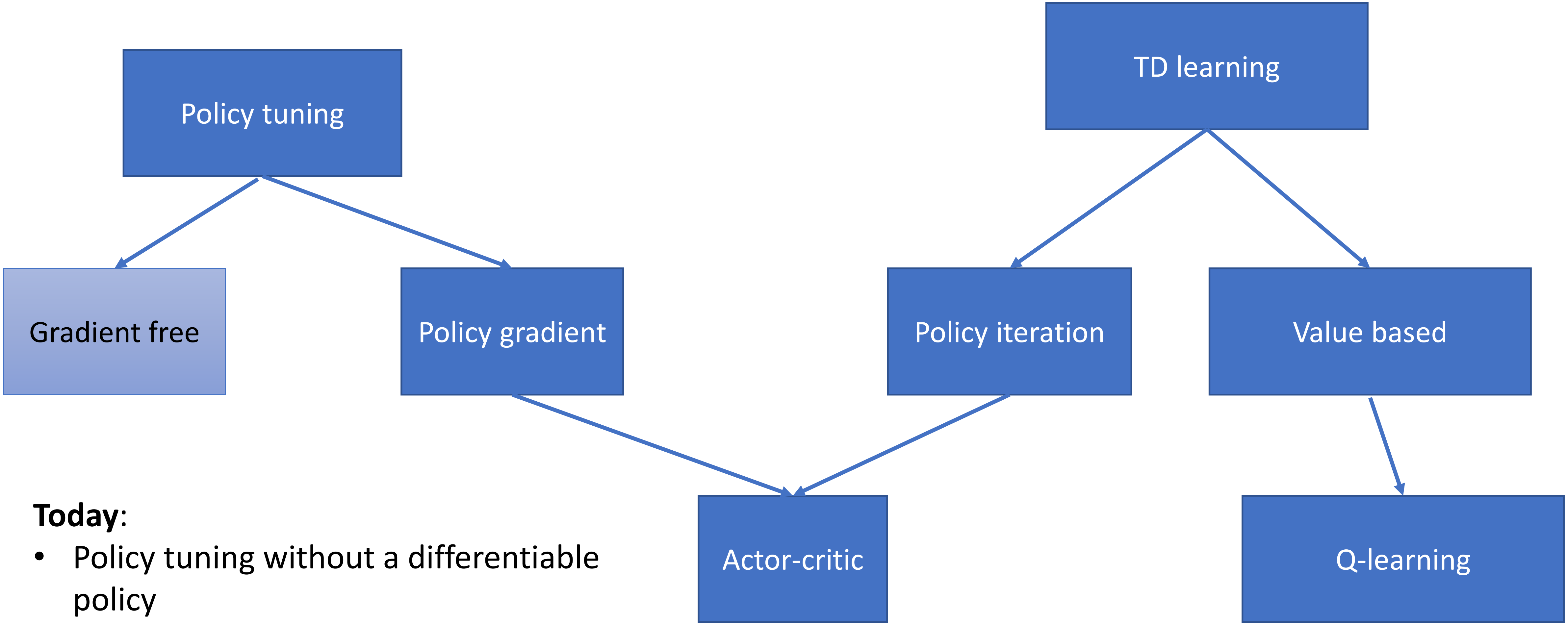



Instructor: Guni Sharon

# Final report

- Please limit to 6 pages
- Be concise and on-point
- Follow a clear narrative

# Solver classes

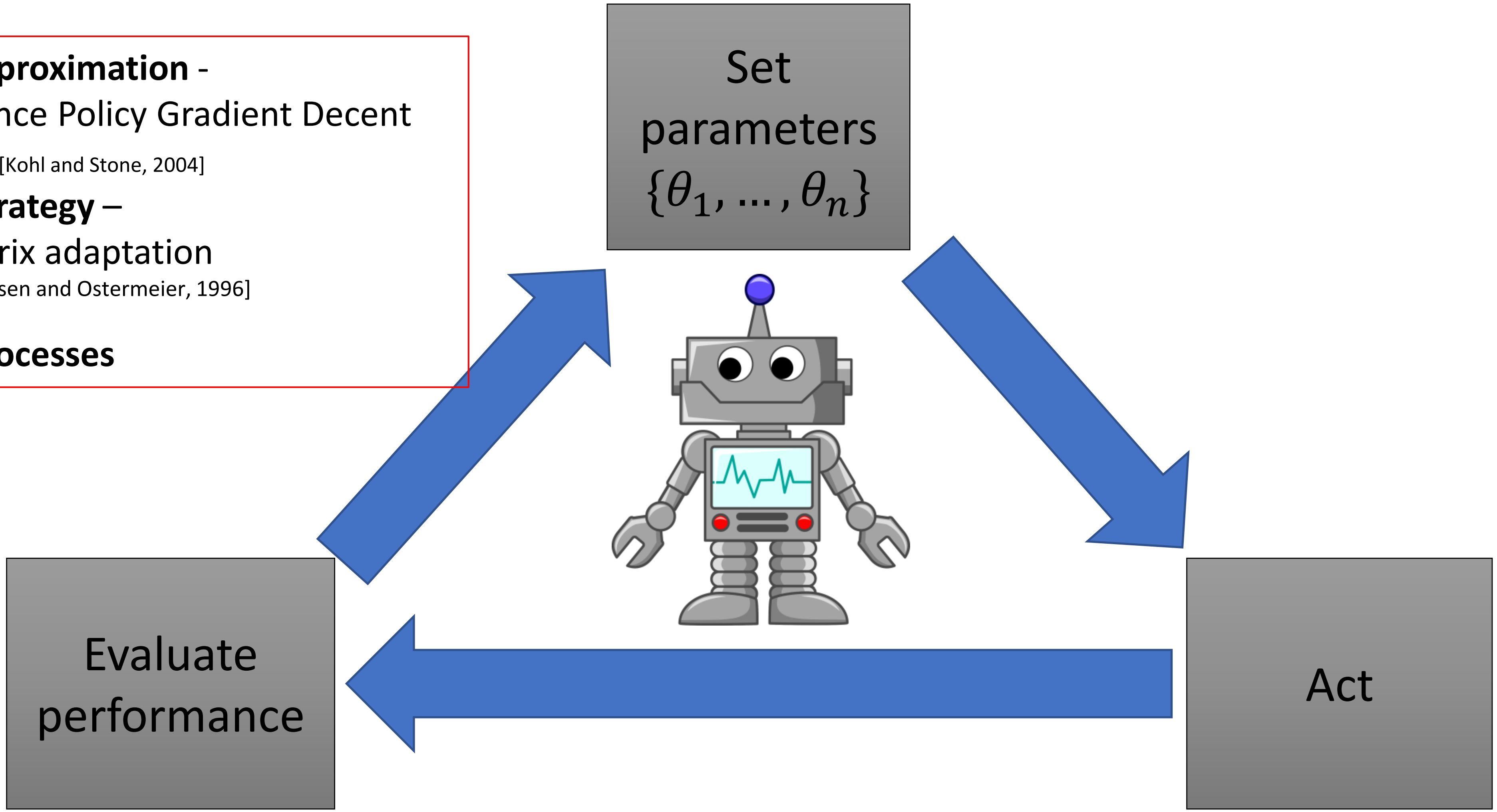



**Today**:

- Policy tuning without a differentiable policy
- Also known as gradient free methods

# Online parameter tuning

1. **Gradient approximation** -
   Finite Difference Policy Gradient Decent
   [Kohl and Stone, 2004]
2. **Evolution strategy** –
   covariance matrix adaptation
   [Hansen and Ostermeier, 1996]
3. **Gaussian processes**



# Online parameter tuning

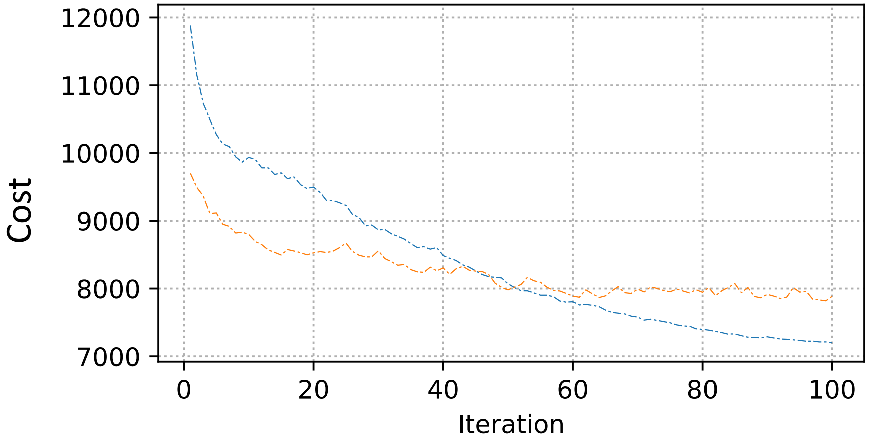
- Minimize $f: \mathbb{R}^n \rightarrow \ \mathbb{R}$
  - Where is the optimum of $f(X)$ ?
  - If convex then at $\nabla_X f \ = 0$ (Lagrange, KKT), else, gradient decent (ADAM, Newton)
- What if $f$ is unknown and we can only sample $f(X)$ for a given $X$ ?
  - Approximate $f$ with supervised learning
- Observations are not provided -> must learn from interactions
  - Reinforcement learning!
- $f$ is unknown/not differentiable
  - Blackbox (derivative free) optimization!

# Examples

- **Comparison between popular genetic algorithm (GA)-based tool and covariance matrix adaptation – evolutionary strategy (CMA-ES) for optimizing indoor daylight**
  Manal Anis, Sumedh Pendurkar, Yun Kyu Yi, and Guni Sharon
  In *Proceedings of Building Simulation 2023: 18th Conference of IBPSA*, 2023
- **Bilevel Entropy based Mechanism Design for Balancing Meta in Video Games**.
  Sumedh Pendurkar, Chris Chow, Luo Jie, and Guni Sharon
  In *Proceedings of the 22th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2023

# Blackbox optimization

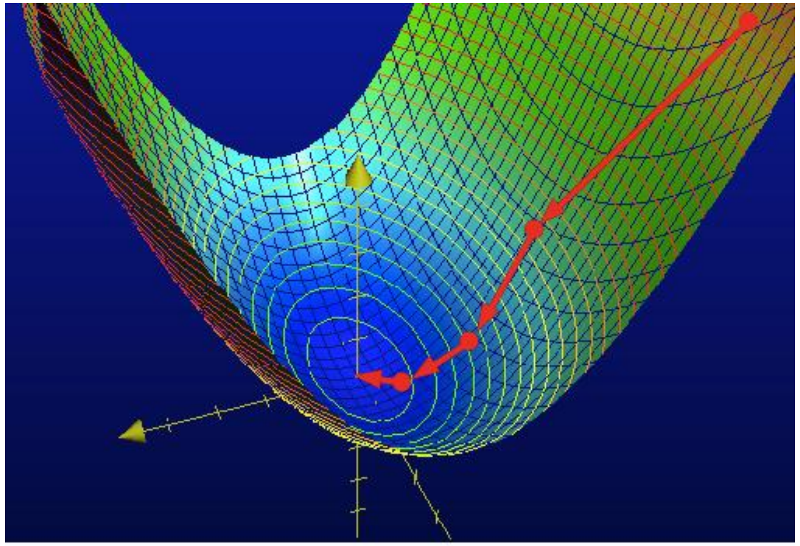
- Challenges:
  - Compute the optimum of an unknown function via samples
  - **Sample efficiency**: minimize the number of samples along the training process
  - **minimize** cumulative regret (for online optimization)



# 1. Gradient Descent

- Minimize $f(x_1, x_2, \ldots, x_n)$
  1. Set initial parameter vector $X^0 = [x_1^0, \ldots, x_n^0]^\top$
  2. While improving
     - $X^{k+1} = X^k - \alpha \nabla f^{\mathrm{k}}$ ?

How can we compute the gradient if $f$ is unknown?



# Empirical gradient

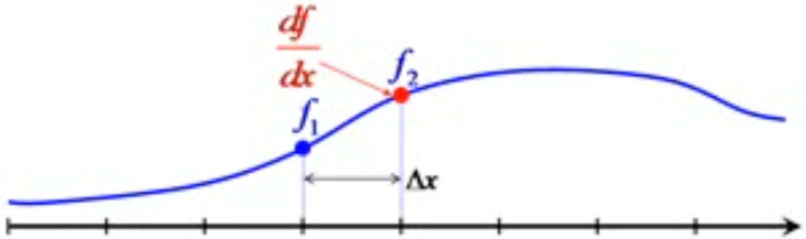
- approximate $\nabla f(x_1, x_2, \ldots, x_n)$ at $X$
  - Through local evaluation
- Finite-difference methods (FDM)
  - By definition: $\frac{\partial f}{\partial x_i} = \lim_{x_i - x_i' \to 0} \frac{f(X) - f(X')}{x_i - x_i'}$
  - Not well-defined for non-differentiable $f$
  - However, we can avoid infinities by setting $x_i - x_i' > 0$, $\frac{\partial f}{\partial x_i} \cong \frac{f(X) - f(X')}{x_i - x_i'}$

# Finite difference approach

Biased towards the rear

Backward difference

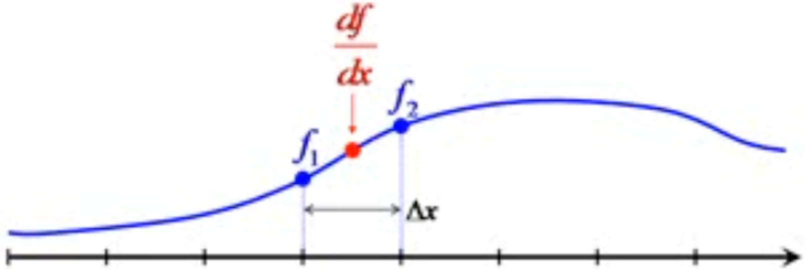
$$\frac{df_2}{dx} \approx \frac{f_2 - f_1}{\Delta x}$$




Unbiased
X2 samples

Central difference

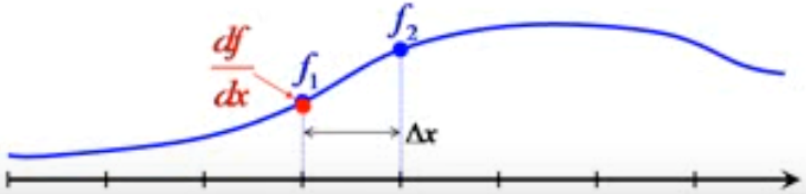
$$\frac{df_{1.5}}{dx} \approx \frac{f_2 - f_1}{\Delta x}$$




Biased towards the front

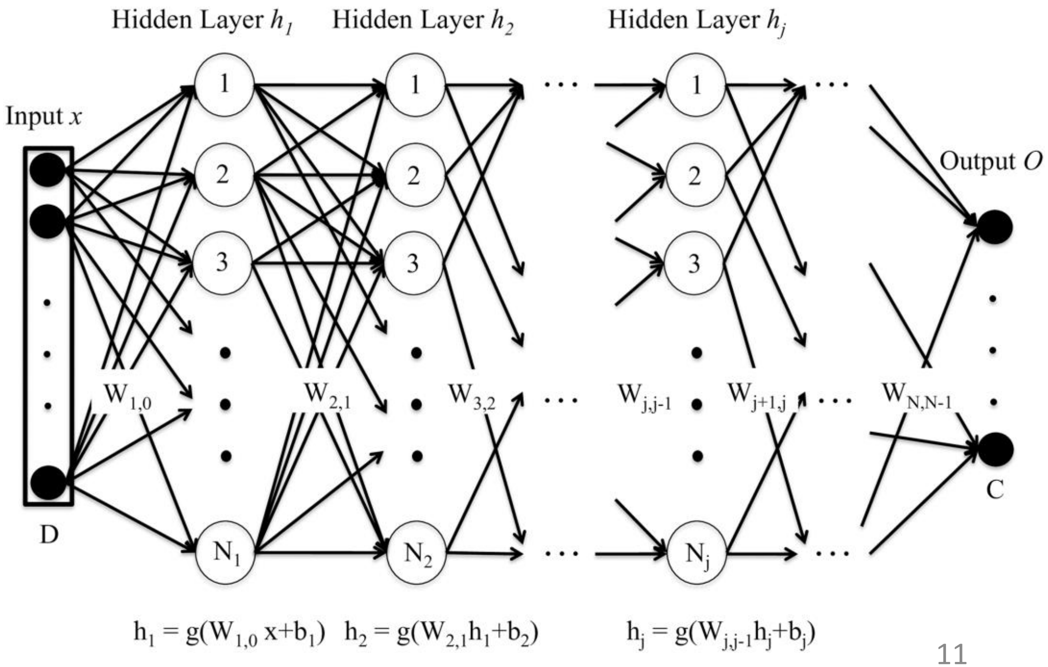
Forward difference

$$\frac{df_1}{dx} \approx \frac{f_2 - f_1}{\Delta x}$$



# High dimensionality

- Approximate the gradient of a DNN with 1M parameters
- Requires 2M observations!
  - Not practical
- A stochastic approach is required



# Finite Difference Policy Gradient

(Kohl and Stone, 2004)

- Evaluate $\nabla f^{\mathrm{k}}$ through $M$ observations
  - For $m = [1, \dots, M]$
    - $\tilde{X}^{k,m} =< x_1^k + \delta_1^m, \dots, x_n^k + \delta_n^m >^T$
  - $c_{+\varepsilon n}^k = \mathrm{Mean}_{x \in \tilde{X}_{+\varepsilon n}^k}\left(f(x)\right)$

  Similar definition for $c_{-\varepsilon n}^k$ and $c_{0n}^k$

$\delta_n^m = Rand(-\varepsilon, 0, \varepsilon)$

$\tilde{X}_{+\varepsilon n}^k$ = all parameter vectors where $x_n^k$ was increased by $\varepsilon$

- $\frac{\partial f^{\mathrm{k}}}{\partial x_n^k} = \begin{cases} 0 & , c_{+\varepsilon n}^k < c_{0n}^k \text{ \& } c_{-\varepsilon n}^k < c_{0n}^k \\ c_{+\varepsilon n}^k - c_{-\varepsilon n}^k , & else \end{cases}$

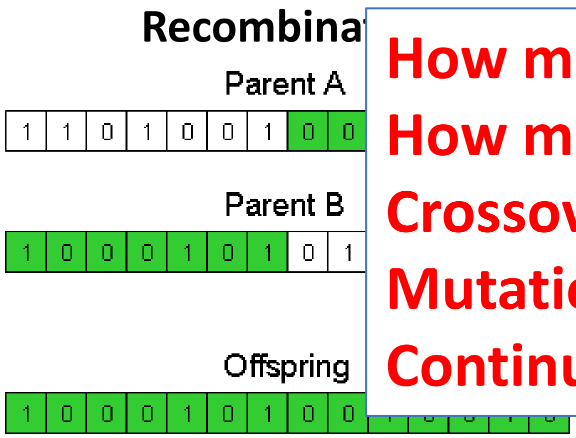
# Example

- Minimize $f(x_1, x_2)$, $M = 4$, $\varepsilon = 1$, $\alpha = 1$
  - Initial parameter vector $X^0 = [0,0]^\top$
    - $\tilde{X}^{0,0} = [1,1]^\top$, $f(\tilde{X}^{0,0}) = 5$
    - $\tilde{X}^{0,1} = [-1,0]^\top$, $f(\tilde{X}^{0,1}) = 3$
    - $\tilde{X}^{0,2} = [0,0]^\top$, $f(\tilde{X}^{0,2}) = 6$
    - $\tilde{X}^{0,3} = [1,-1]^\top$, $f(\tilde{X}^{0,3}) = 4$
- $c^0_{+\varepsilon 1},\ c^0_{-\varepsilon 1}, c^0_{01}, c^0_{+\varepsilon 2},\ c^0_{-\varepsilon 2}, c^0_{02} =$
- $\nabla f^0 =$
- $X^1 =$

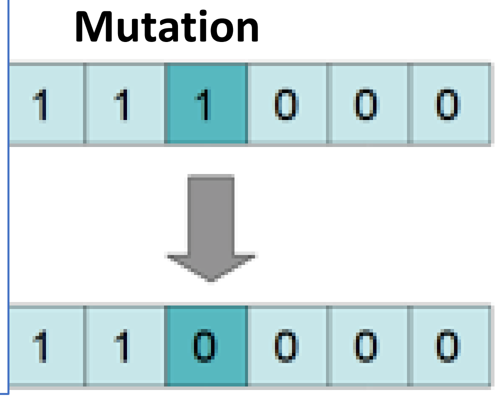
# Genetic Algorithm

1. Generate $M$ parameter vectors $\{\tilde{X}^{0,0}, \tilde{X}^{0,1}, \dots, \tilde{X}^{0,2}\}$
2. **Selection**: Survival of the fittest
   - $G \leftarrow$ set of parameter vectors with best fitness: $f\left(\tilde{X}^{k,m}\right)$
3. Generate next generation through **recombination** and **mutation** on $G$
4. Goto 2. until termination criterion fulfilled

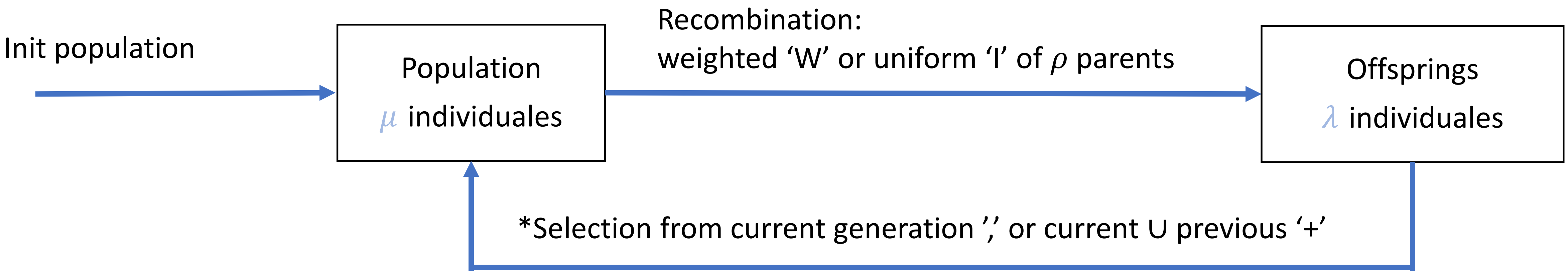
Might apply to the unification of current and former generations




**How many parents?**
**How many offsprings?**
**Crossover technique?**
**Mutation technique?**
**Continuous space?**



# $\left(\mu/\rho_{\{I,W\}} \overset{+}{,} \lambda\right)$-Evolutional Strategy



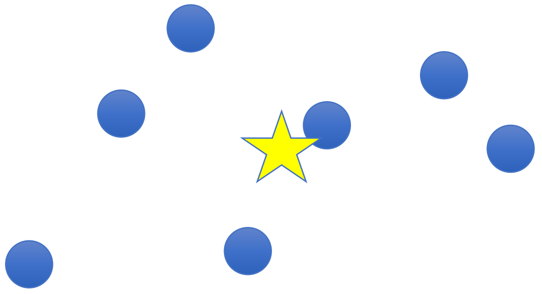

If comma selection, then $\lambda > \mu$

*While the ‘,’ selection is recommended for unbounded search spaces (Schwefel, 1987), the ‘+’ selection should be used in discrete finite size search spaces, e.g., in combinatorial optimization problems (Herdy, 1990; Beyer, 1992).

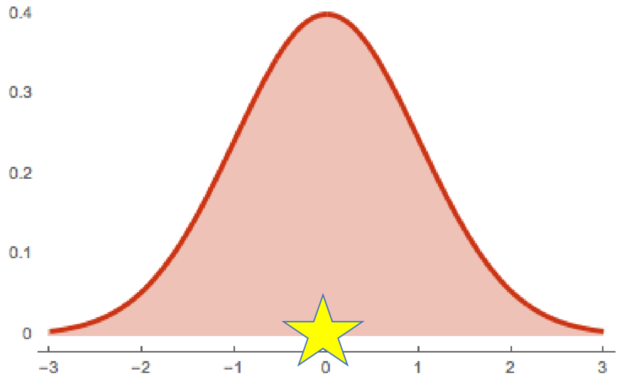
# Continuous space

- **Recombination**: $x_i$ for offspring = (weighted) mean $m_i$ over the group $\rho$ from the fittest individuals
- **Mutation**: for every offspring, sample each parameter $x_i \sim \mathcal{N}(m_i\ , Var_i)$

**Recombination**



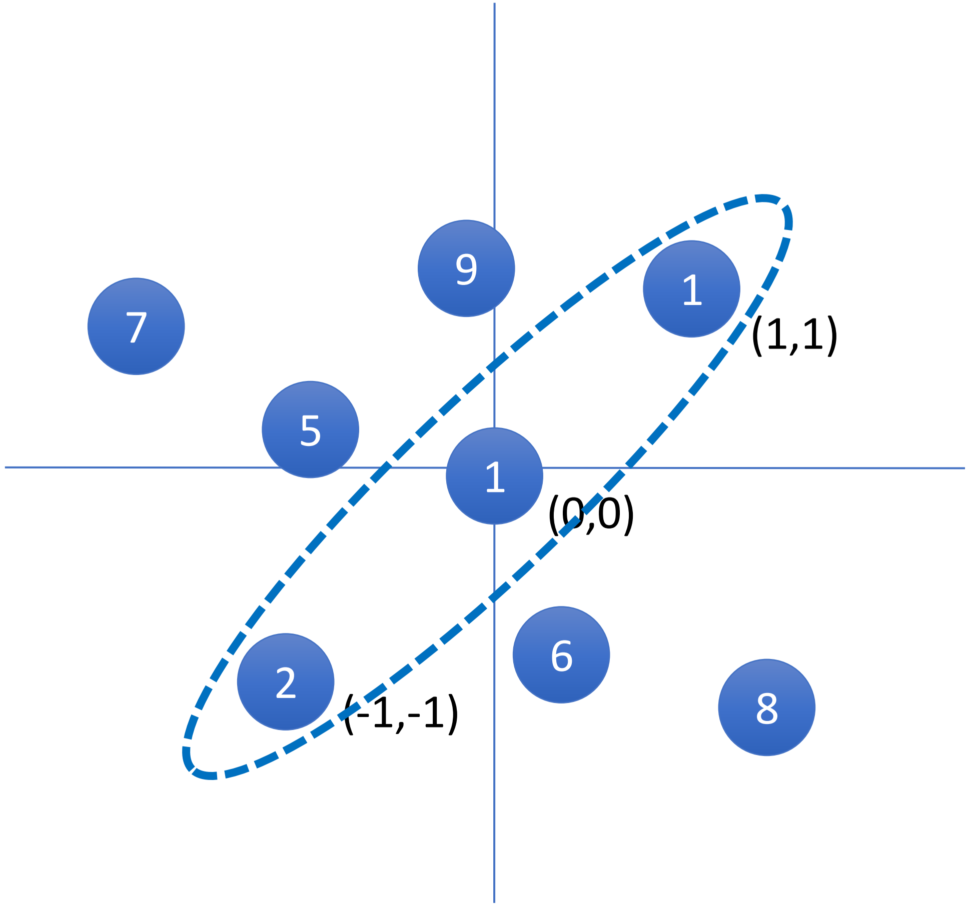
**Mutation**



# Recombination

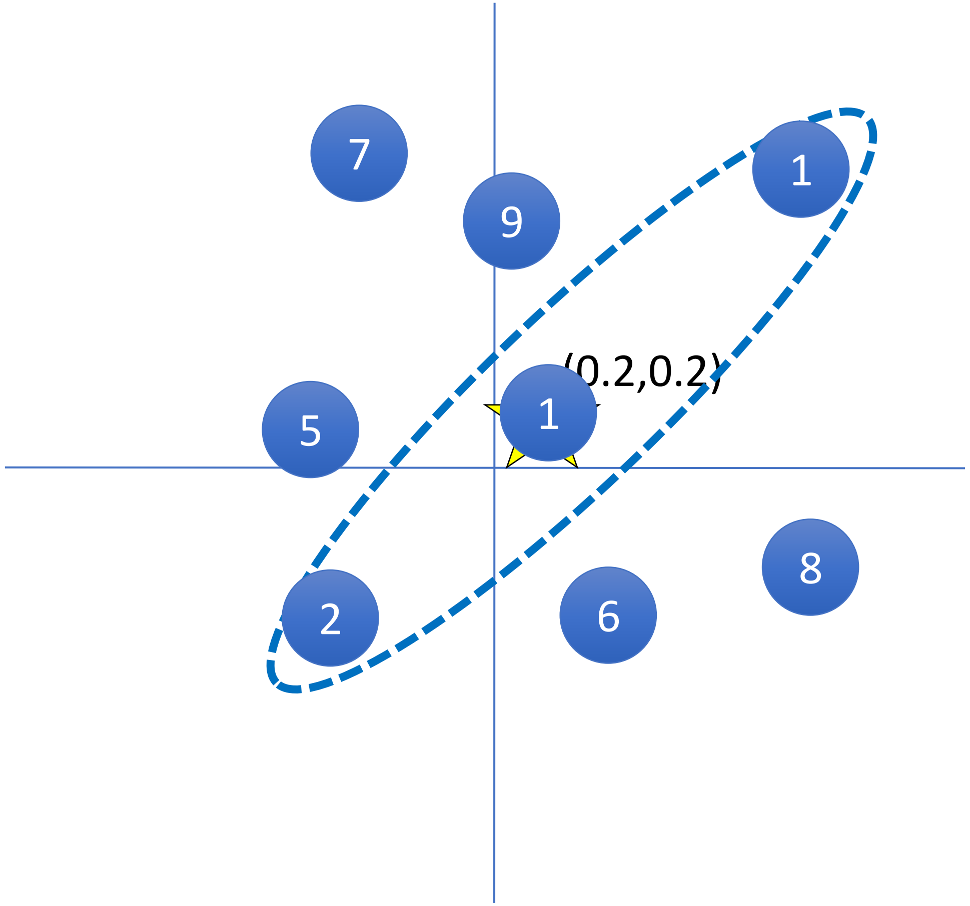
- Assume $(3/3_W, 8)$-ES
- Select 3 fittest parents
- Recombine (weighted) groups of size 3
- Generate 8 offsprings
- Update:
  - $m_1 =$
  - $m_2 =$
  - $Var_1 =$
  - $Var_2 =$

$$\text{sample var} = \frac{\sum (x_i - \bar{x})^2}{n-1}$$



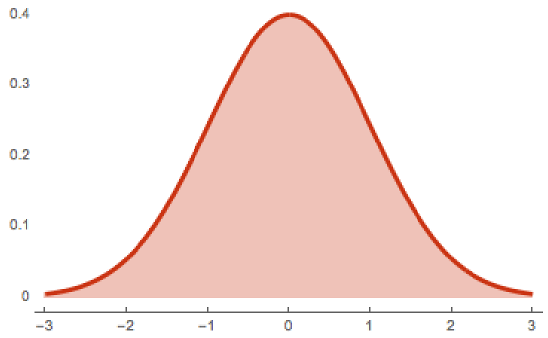
# Mutation

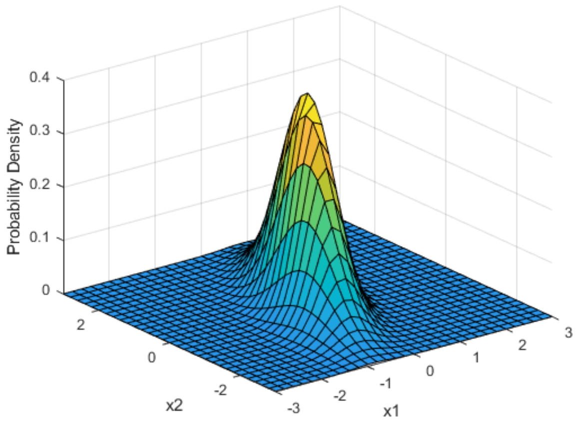
- generate 8 offsprings
  - $x_1 \sim \mathcal{N}(0.2\,, 0.7)$
  - $x_2 \sim \mathcal{N}(0.2\,, 0.7)$
- Ooops! What went wrong?
  - We failed to enforce the variables' co-dependencies
  - Instead of $Var_i$ consider the covariance matrix $(C)$

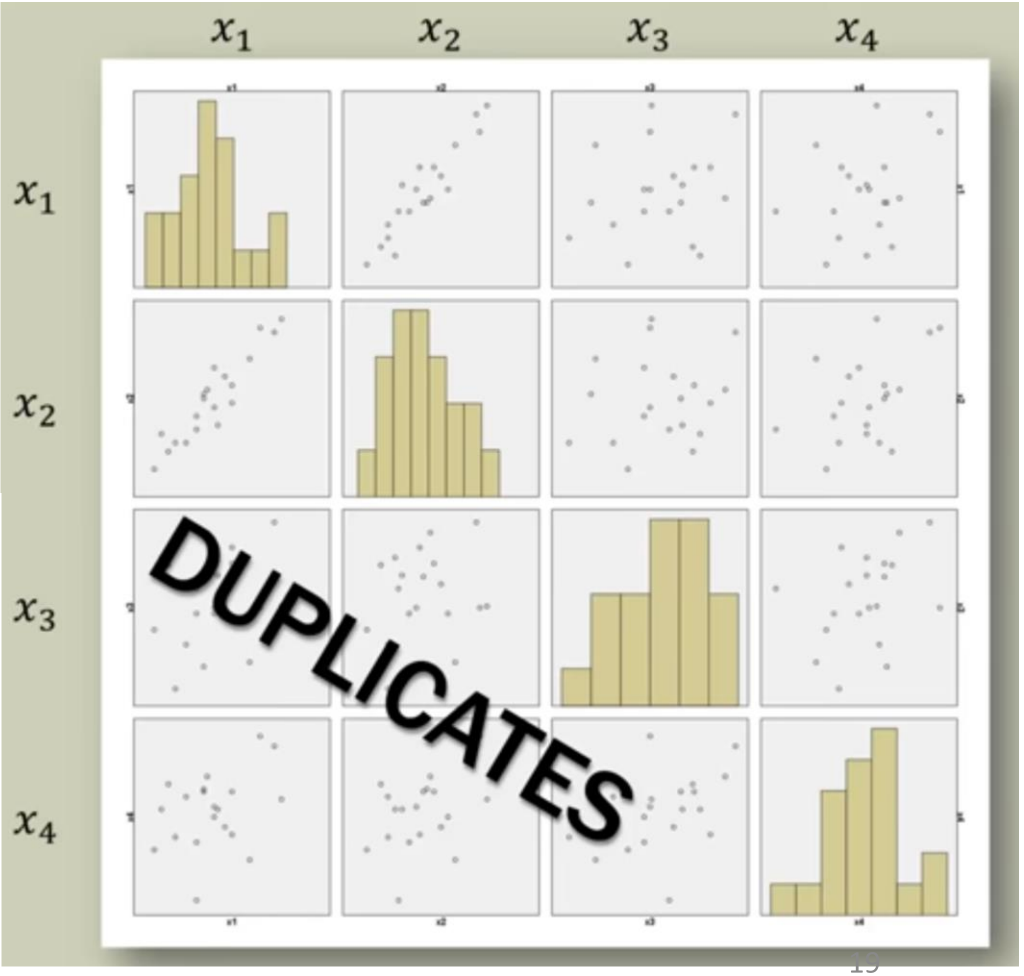


# Covariance matrix

- Capture dependencies between the dimensions
- Enforce such dependencies when spawning offsprings
- $X \sim \mathcal{N}(m, C)$

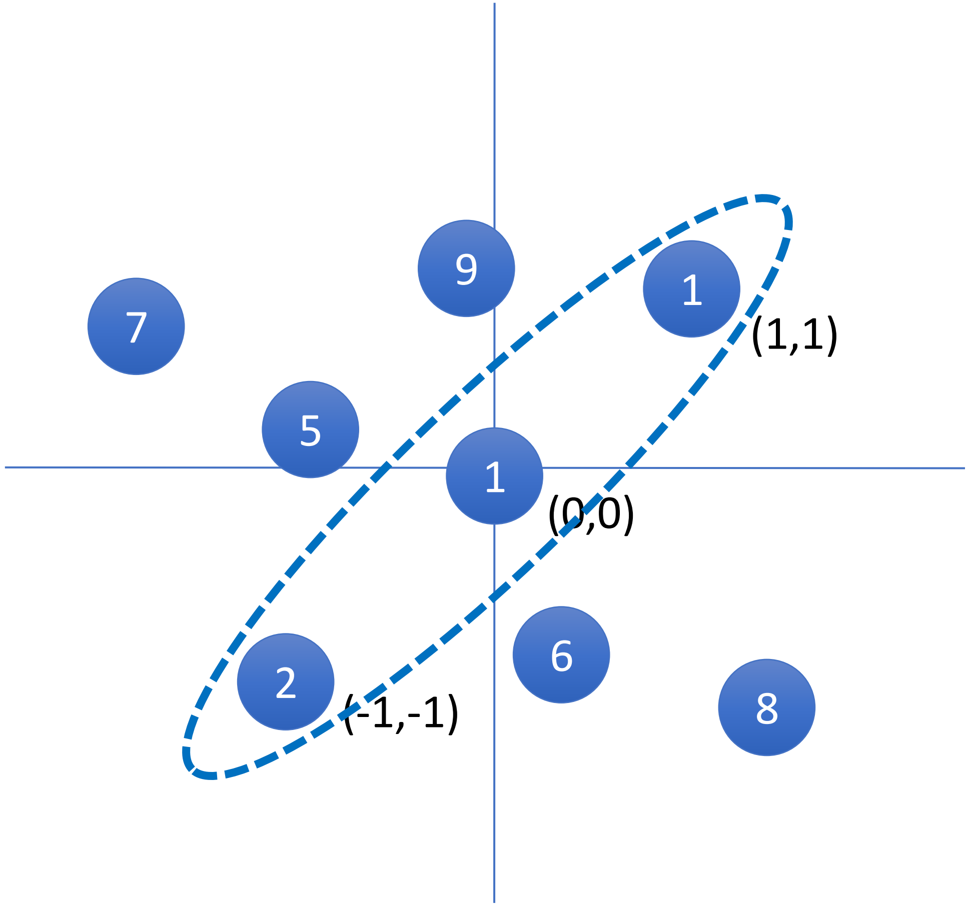


# Lets try again…

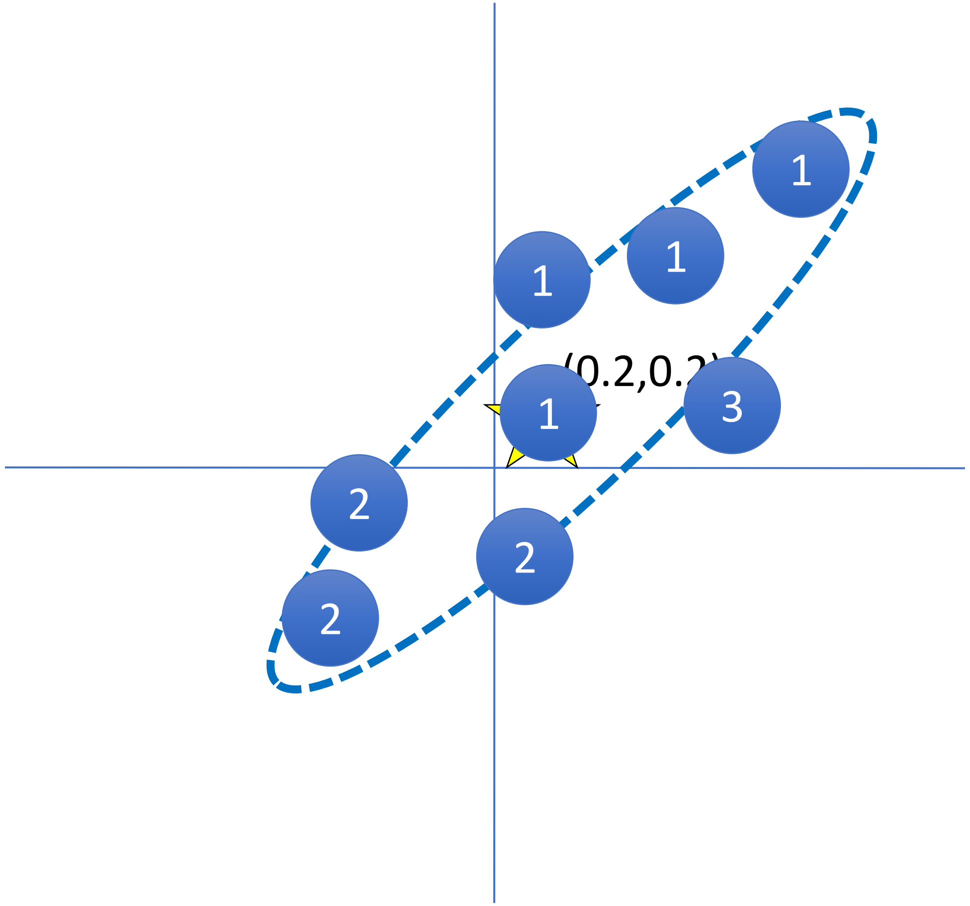
- Assume $(3/3_W, 8)$CM-ES
- Select 3 fittest parents
- Update mean and covariance
  - $m =$
  - $C =$




$$^{*}C_{ij} = \mathbb{E}[(x_i - m_i)(x_j - m_j)] = \frac{1}{n-1}\sum_{k=1}^{M}(x_i^k - m_i)(x_j^k - m_j)$$

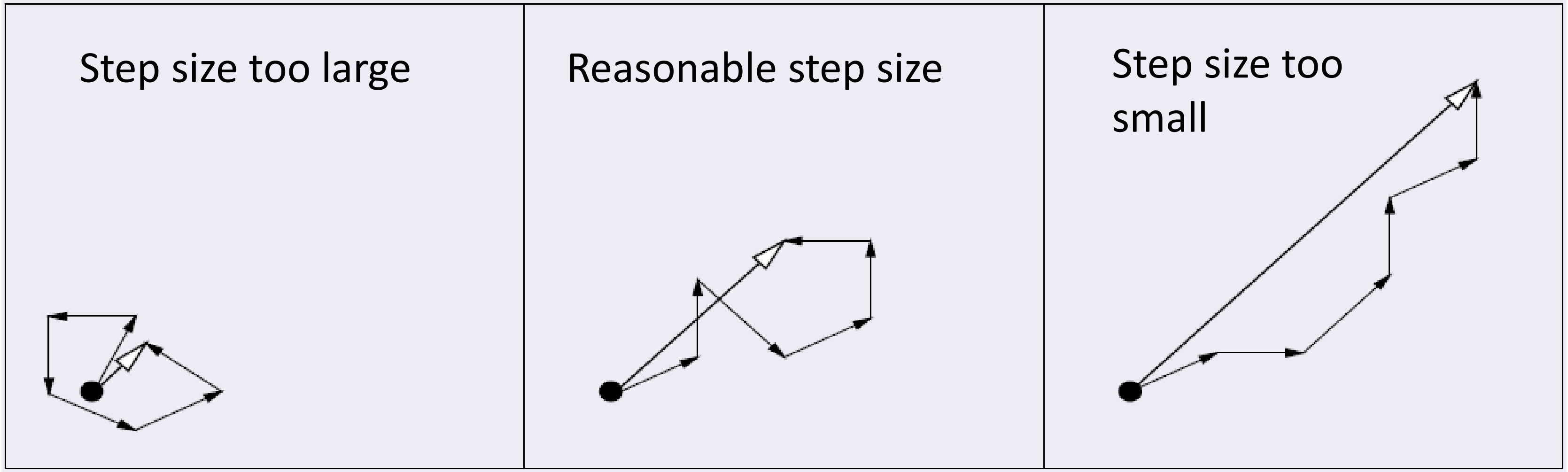
# Mutation

- Create 8 offsprings
  - $X \sim \mathcal{N}\left(\begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}, \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix}\right)$
- Much better!

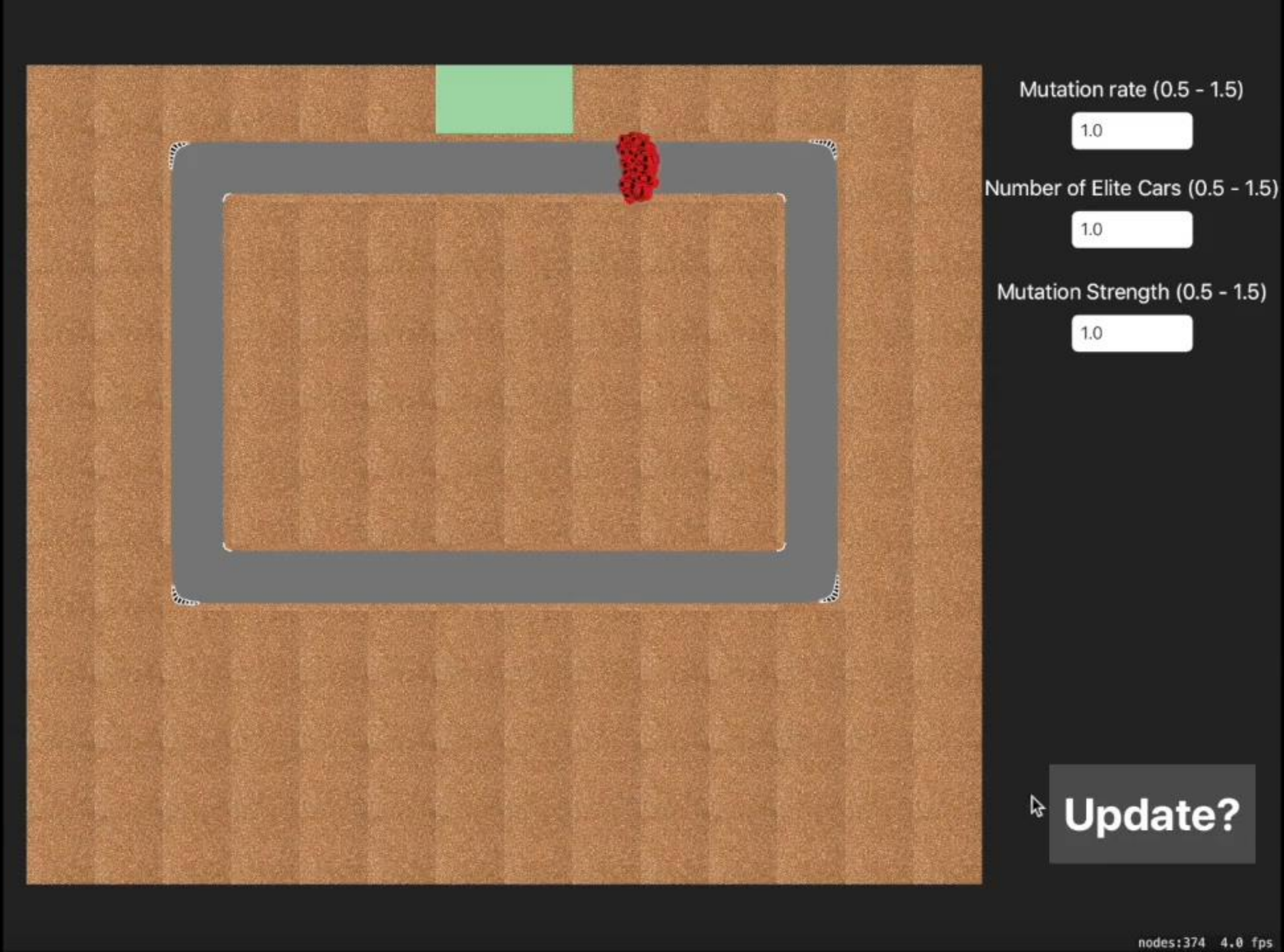


# Step-size control

- For updating the mean and covariance matrix between generations

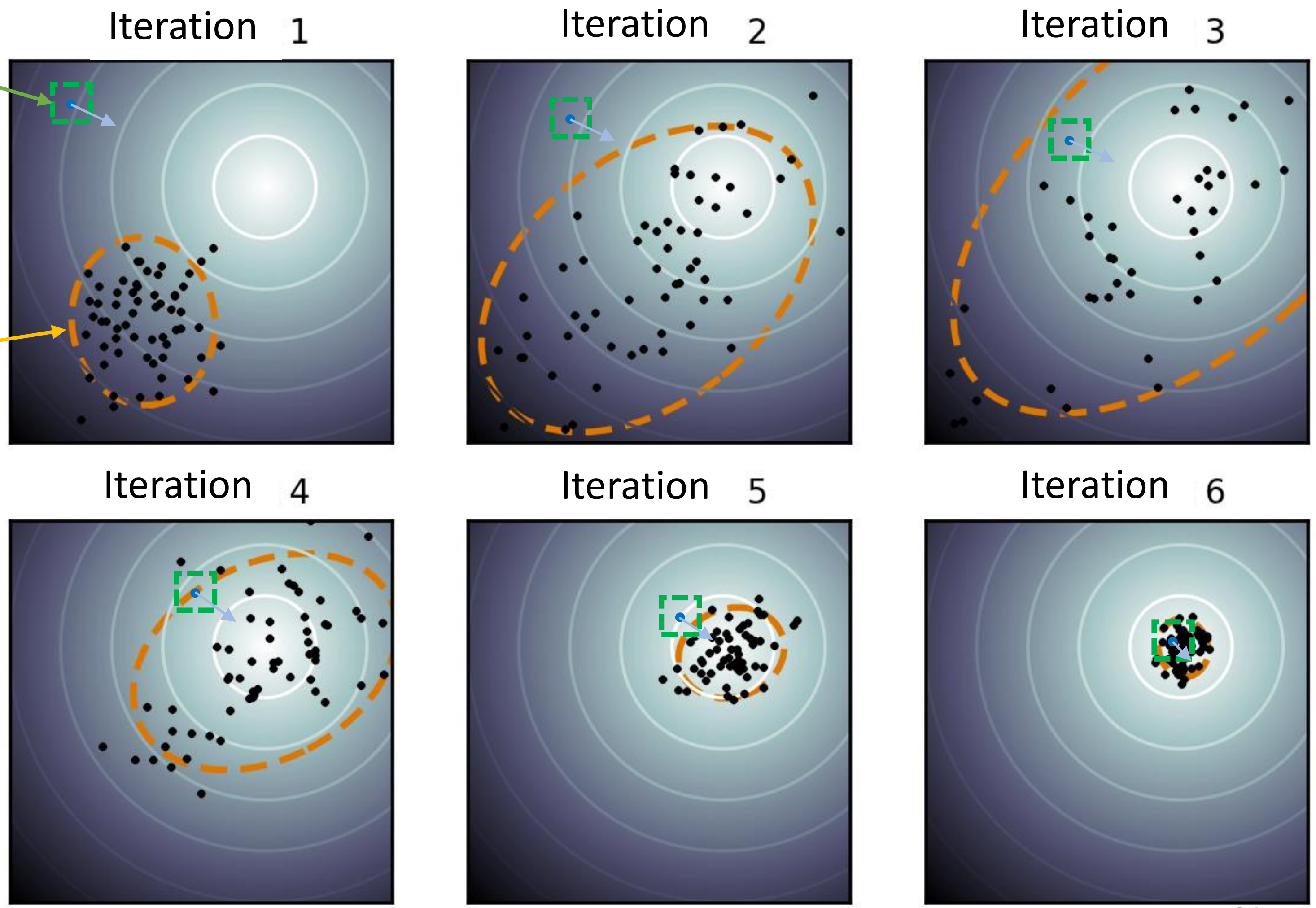



if several updates go to the same/similar direction the step-size is increased
We've seen this before (momentum)


Mutation rate (0.5 - 1.5)
1.0
Number of Elite Cars (0.5 - 1.5)
1.0
Mutation Strength (0.5 - 1.5)
1.0
Update?
nodes:374 4.0 fps

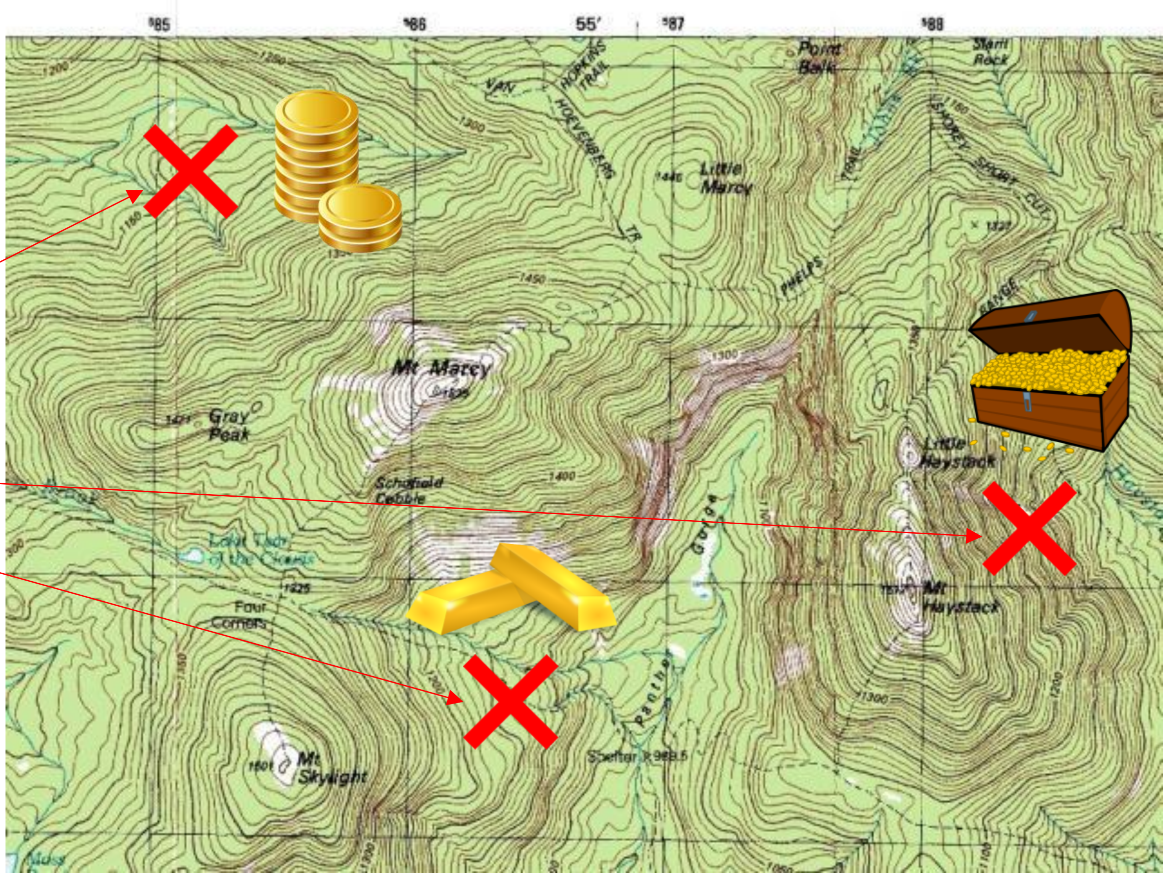
# Online parameter tuning

- Finite difference
  - Less exploration
  - Safer
  - Harder to escape a local minima
- CMA-ES
  - More exploration
  - Less safe
  - Easier to escape a local minima



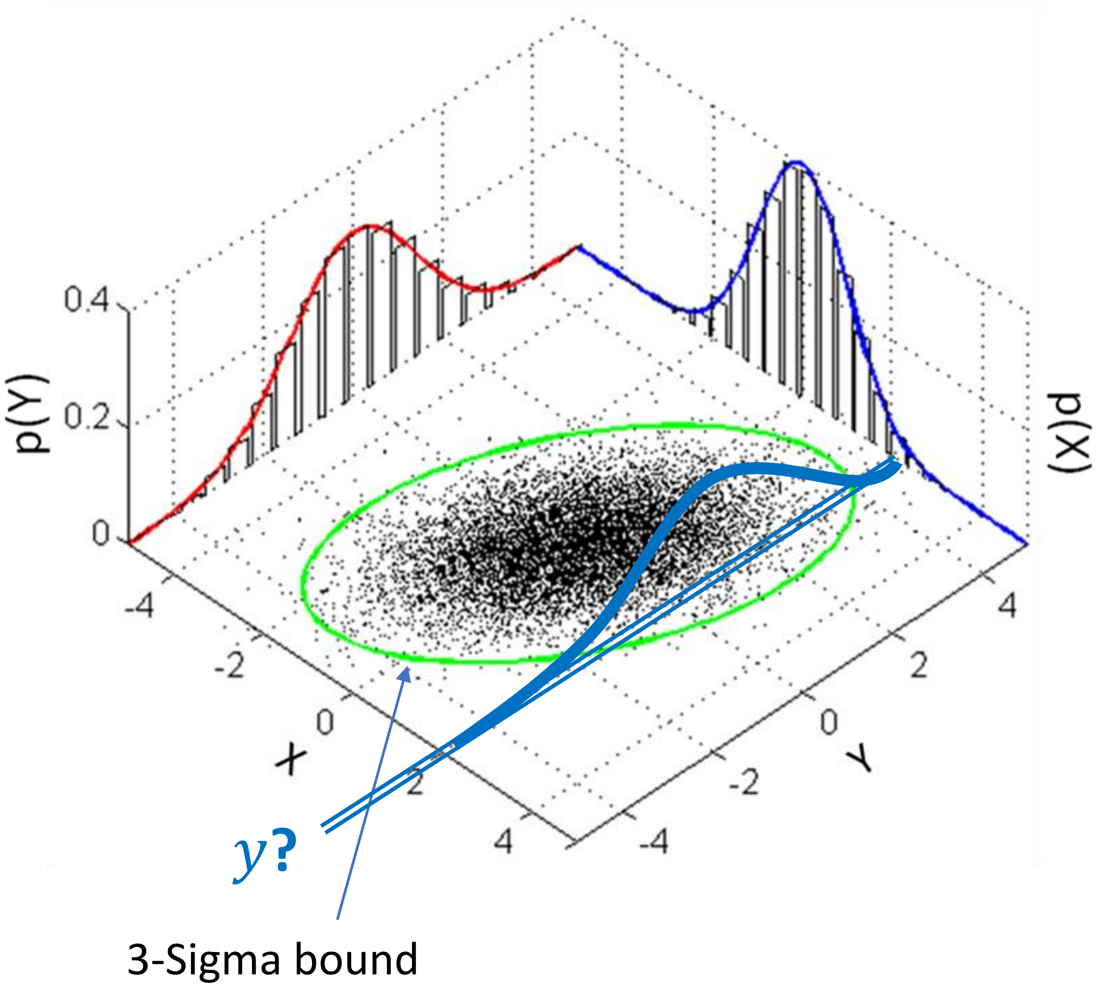
# Looking for gold

- Daniel Krige - South African mining engineer, 1960
- Find the point richest in gold
- Sampling the soil is expensive
- Given current samples
  - Where should the next sample be taken from?



# Conditional distribution

- Multivariate normal distribution
- $\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix}\right)$
- What is the distribution over $y$ given $x = 2$ ?



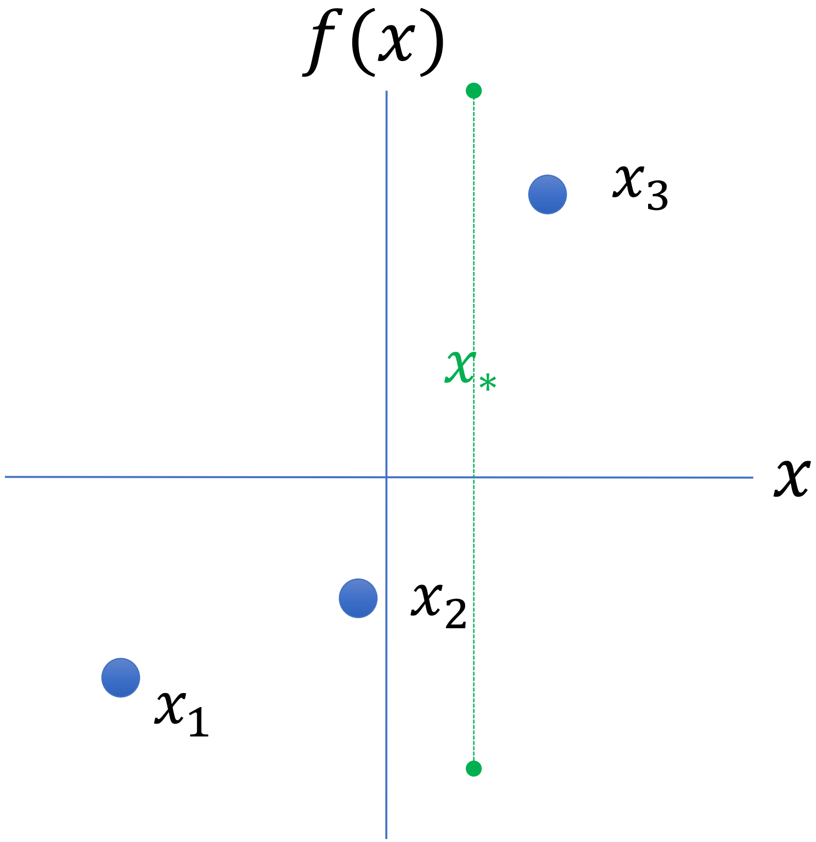
# Posterior conditionals of an MVN

- Theorem 4.3.1, "Machine Learning: a Probabilistic Perspective" by Kevin Patrick Murphy
- $\mu_{x|y} = \mu_x + C_{xy} C_{yy}^{-1} (y - \mu_y)$
- $C_{x|y} = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx}$
- $\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix} \right)$
- $\mu_{x|y} =$
- $C_{x|y} =$

In the example x and y are a single parameter but they can represent high-dimensional vectors (the general case). Hence the matrix notation.

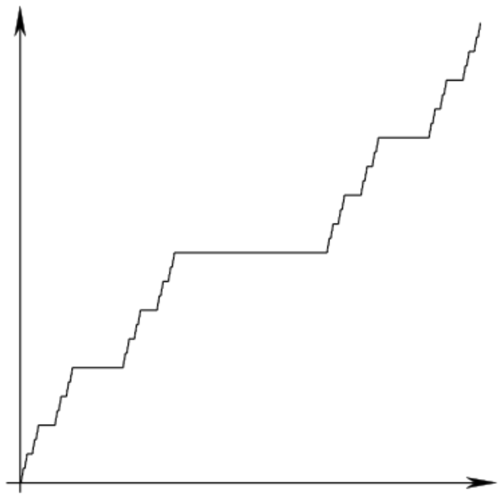
# Gaussian Prediction

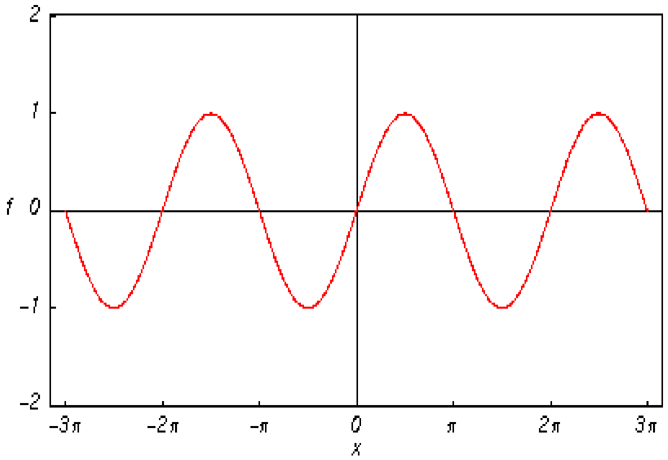
- Given training data $\{X, f_i\}$ and $x_*$ predict the mean and variance for $f_*$
- "The key idea is that if $x_i$ and $x_j$ are deemed by the kernel to be similar, then we expect the output of the function at those points to be similar, too" Machine Learning: a Probabilistic Perspective



# Kernels

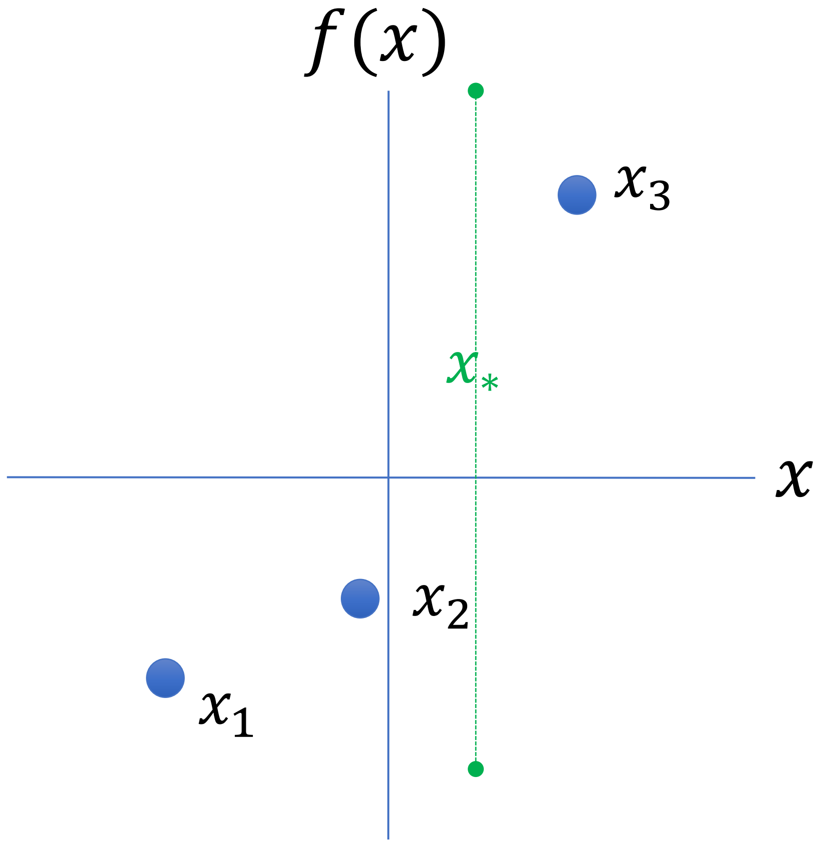
- Kernel functions define the co-variance between two points in some (user defined) space
- The chosen space should fit the approximated function/domain
- E.g., The **Squared Exponential Kernel**
- $k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2)$
  - $= 0, \quad \text{when } \|x_1 - x_2\| \to \infty$
  - $= 1, \quad \text{when } \|x_1 - x_2\| \to 0$





# Gaussian Prediction

- $k = \begin{bmatrix} 1 & 0.7 & 0.2 \\ 0.7 & 1 & 0.6 \\ 0.2 & 0.6 & 1 \end{bmatrix}$
- Assume: $f \sim \mathcal{N}(\mu, k)$
- Assume: $f_* \sim \mathcal{N}\left(\mu, k(x_*, x_*)\right)$
  - $k(x_*, x_*) = 1$
    - $\mathcal{N}(\mu, 1)$ is not very helpful
- We assume that $f$ and $f_*$ are jointly Gaussian

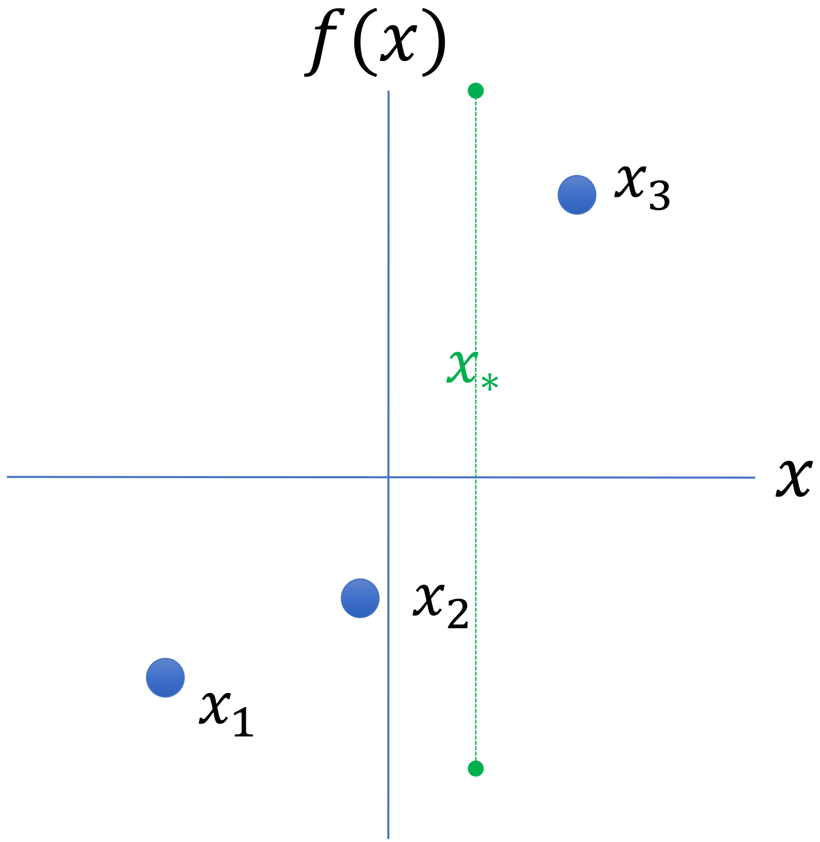


# Gaussian Process

- $\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mu, \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{1*} \\ k_{21} & k_{22} & k_{23} & k_{2*} \\ k_{31} & k_{32} & k_{33} & k_{3*} \\ k_{*1} & k_{*2} & k_{*3} & k_{**} \end{bmatrix}\right)$
- $k_{**} = k(x_*, x_*)$
- $k_{1*} = k(x_1, x_*)$

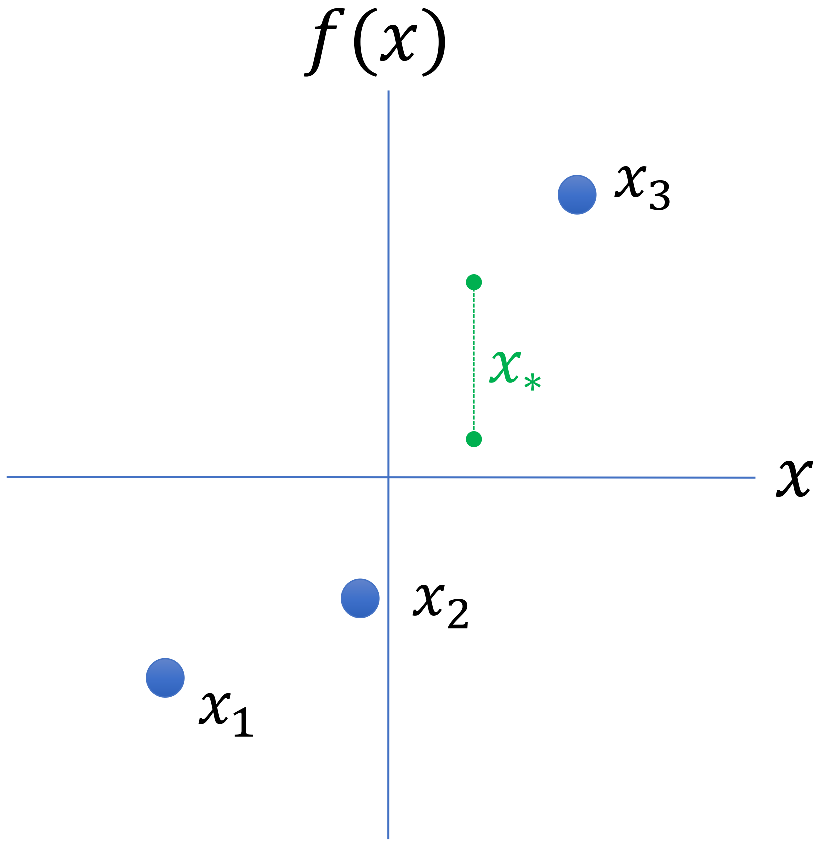
$\boldsymbol{k}$ $\quad$ $\boldsymbol{k}_*$ $\quad$ $\boldsymbol{k}_{**}$

- How can we determine the mean and variance for $f_*$ ?



# Gaussian Process

- Posterior conditionals of a MVN!
  - $\mu_{x|y} = \mu_x + C_{xy} C_{yy}^{-1} (y - \mu_y)$
  - $C_{x|y} = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx}$
- $\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k & k_* \\ k_*^T & k_{**} \end{bmatrix}\right)$
  - $E[f_*]|f = k_*^T k^{-1} f$
  - $c_*|f = k_{**} - k_*^T k^{-1} k_*$

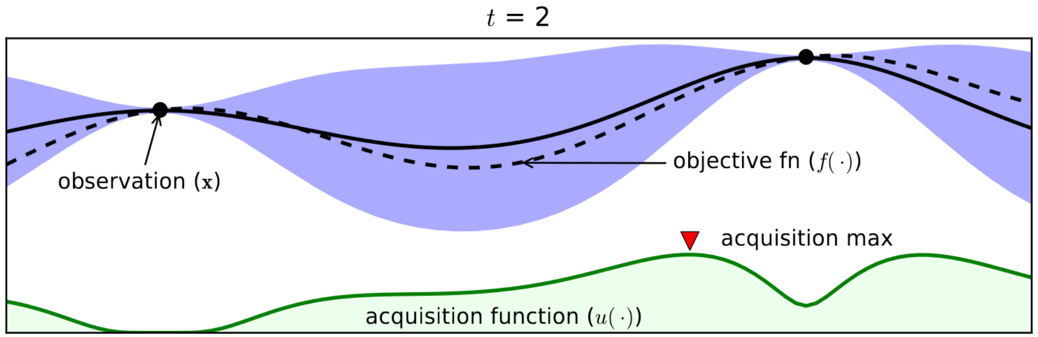


# Bayesian optimization

- A useful tool when:
  - $f$ (the objective function) is “expensive to evaluate”
    - Simulate a day’s traffic, car crash outcome, drill a hole,  human subjects
    - Bounded number of samples
  - $f$ is continuous
  - $f$ lacks known special structure like concavity or linearity
  - No known first- or second-order derivatives
  - Global optimum is required

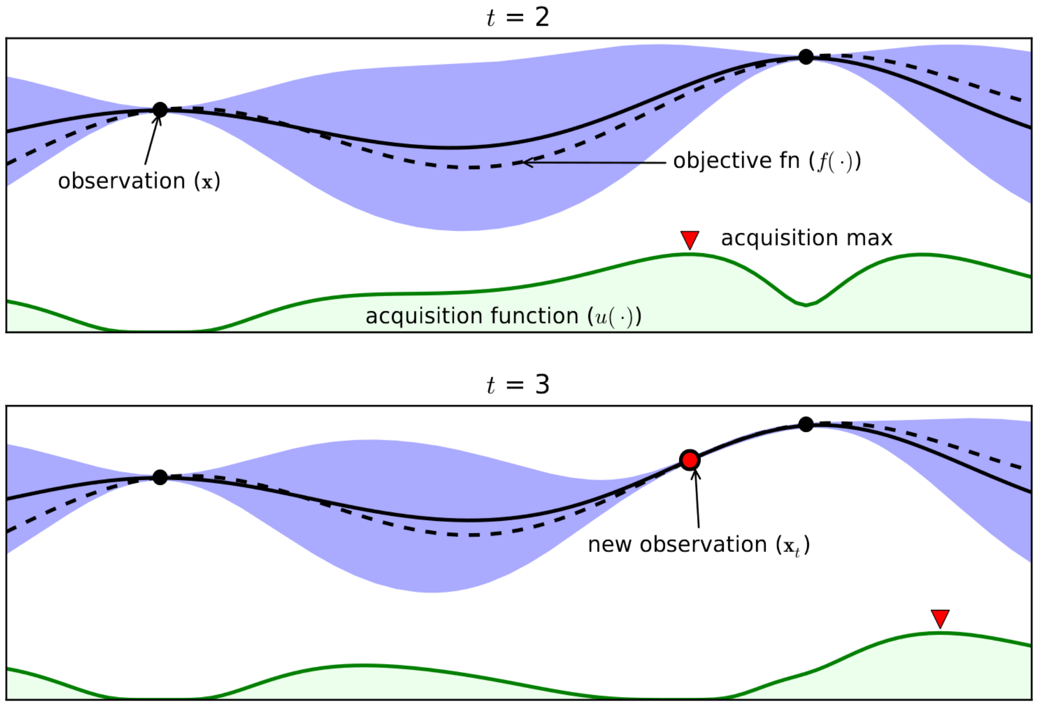
# Bayesian optimization

1. For $t = 1$ until $n$
    2. Get the best sampling candidate: $x_t = argmax_x u(x|\mathcal{D}_{1:t-1})$
    3. Sample the objective function: $y_t = f(x_t)$
    4. Update the GP according to: $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (x_t, y_t)\}$



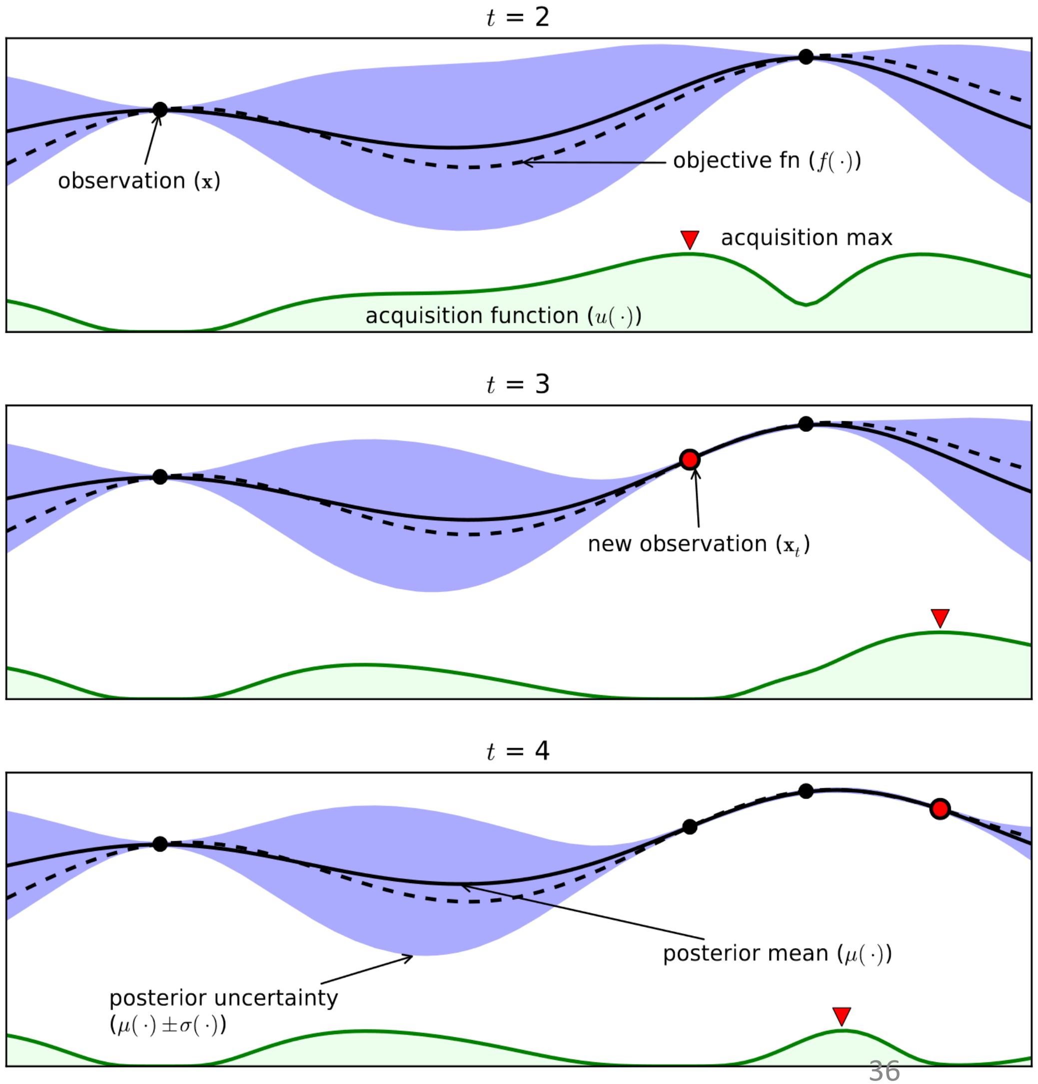
# Bayesian optimization

1. For $t = 1$ until $n$
   2. Get the best sampling candidate: $x_t = argmax_x u(x|\mathcal{D}_{1:t-1})$
   3. Sample the objective function: $y_t = f(x_t)$
   4. Update the GP according to: $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (x_t, y_t)\}$



# Bayesian optimization

1. For $t = 1$ until $n$
   2. Get the best sampling candidate: $x_t = argmax_x u(x|\mathcal{D}_{1:t-1})$
   3. Sample the objective function: $y_t = f(x_t)$
   4. Update the GP according to: $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (x_t, y_t)\}$

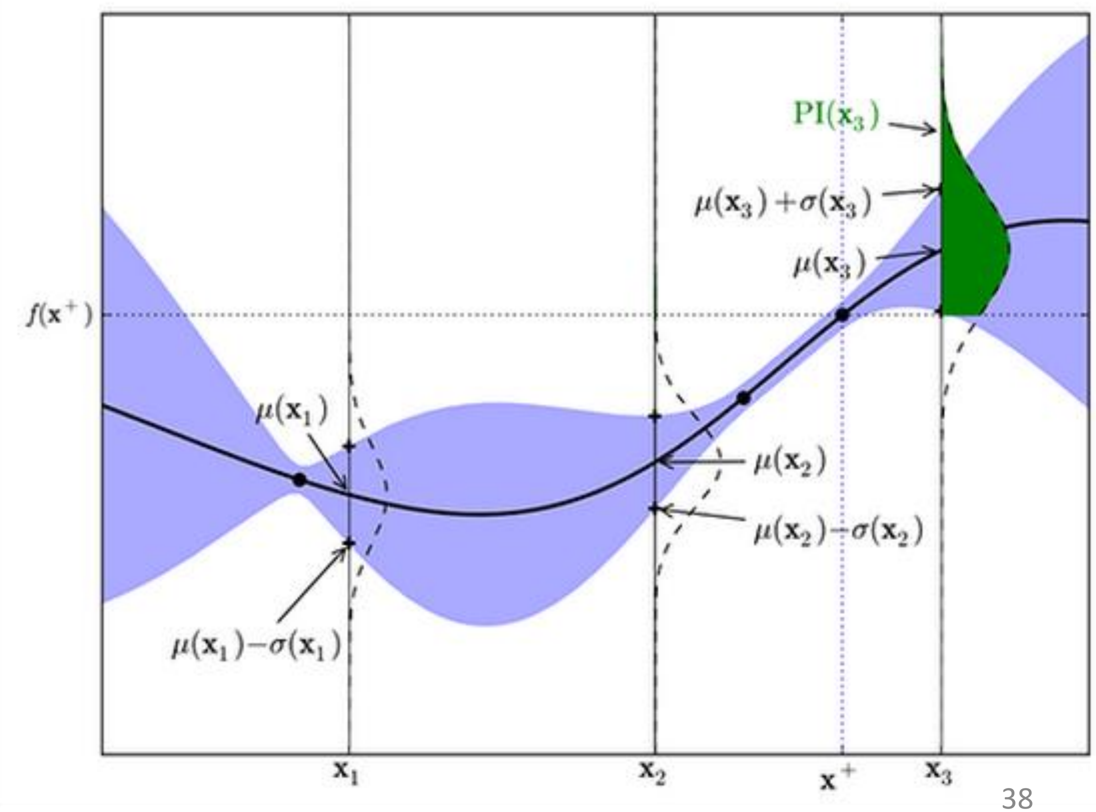


# Acquisition function

- **Exploit**: prefer the points with high expected mean value
- **Explore**: prefer the points with high variance
- The acquisition function balances between the two
    1. Probability of Improvement
    2. Expected Improvement
    3. Gaussian Process Upper Confidence Bound (GP-UCB)
    4. Thompson sampling

# Probability of Improvement [Kushner, 1964]

- $PI(x_*) = P(f(x_*) > \mu^+ + \xi) = \Phi\left(\frac{\mu_* - \mu^+ - \xi}{Var_*}\right)$
  - $\Phi$ is the normal CDF
- The probability that $x_*$ is better than the best known point ($\mu^+$)
- $\xi$ is required so to bias against previously observed points
- PI is very useful if the maximal $f$ value is known a priory



# Expected Improvement [Mockus 1978]

- $x_* = argmax_x \mathbb{E}(\max\{0, f_{n+1}(x) - f^{max}\}|\mathcal{D}_n)$
  - $f^{max} = \max(\mu^+ + \xi)$
- If $X$ is a random variable whose cumulative distribution function admits a density $f(x)$ then: $\mathbb{E}(x)$=$\int xf(x)dx$
- $EI(x_*) = \begin{cases} (\mu_* - \mu^+ - \xi)\Phi(Z) + Var_*\phi(Z) & Var_* > 0 \\ 0 & Var_* = 0 \end{cases}$
  - $Z = \frac{\mu_* - \mu^+ - \xi}{Var_*}$
  - $\phi$ is the normal PDF

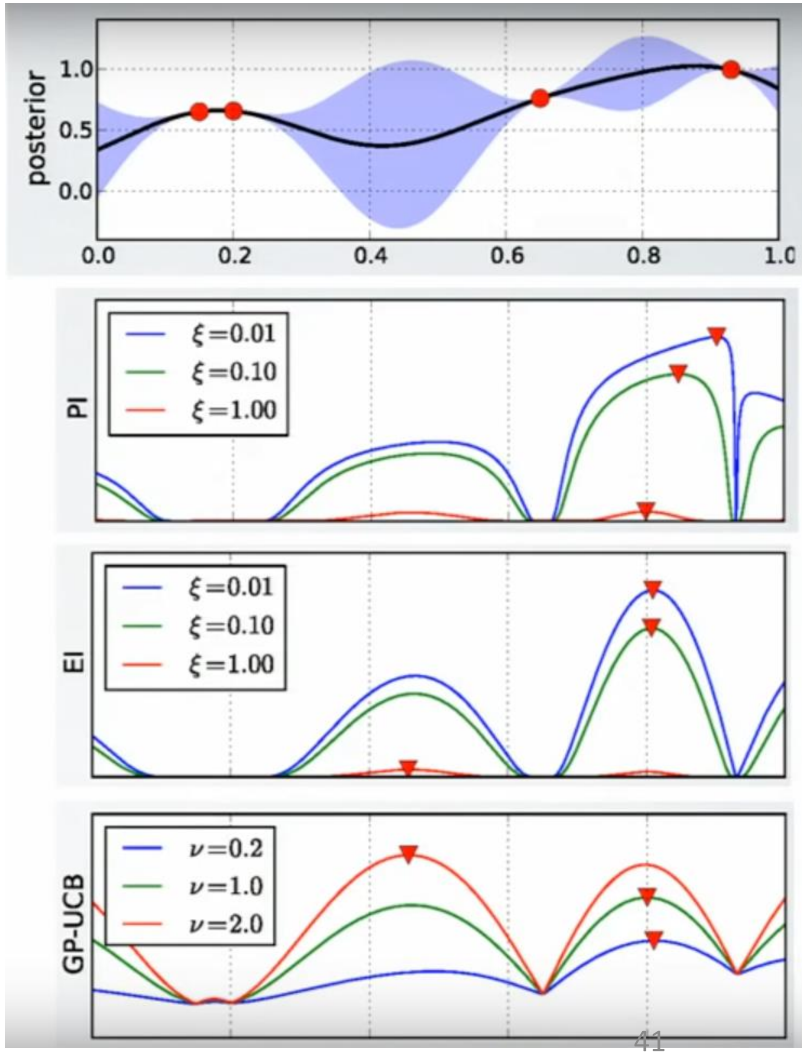
# GP-UCB [Srinivas et al. 2010]

- $GPUCB(x_*) = \mu_* + \sqrt{v\beta_t}Var_*$
- Linear combination of the mean and variance
- Setting $v = 1$ and $\beta_t = 2\log\left(\frac{t^{d/2+2}\pi^2}{3\delta}\right)$ leads to no regret:

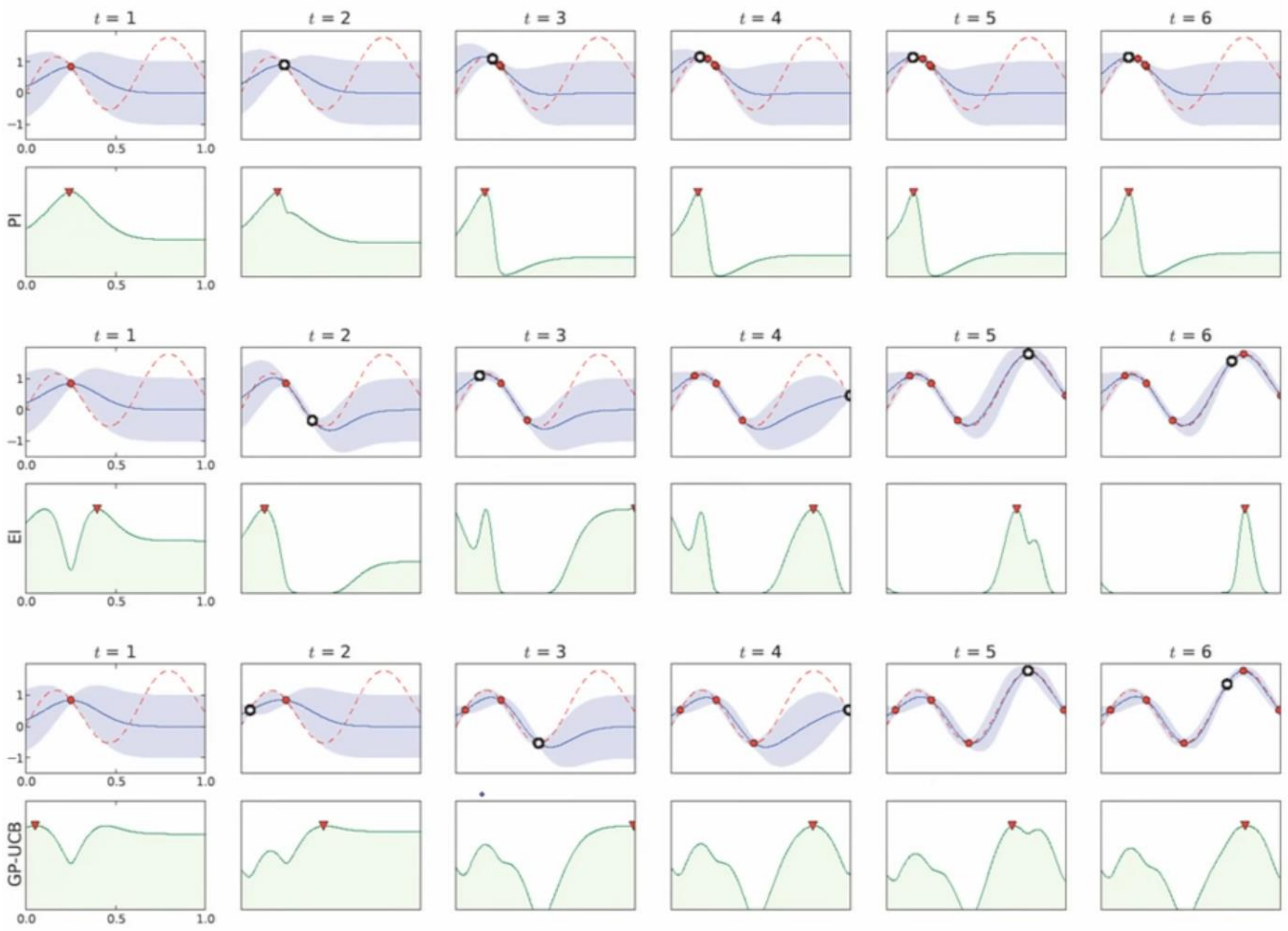
  $\lim_{T\to\infty} \frac{R_T}{T} = 0$
  - $R_T = r(x_1) + \cdots + r(x_T)$
  - $r(x) = f(x^*) - f(x)$

# Acquisition functions

- **PI**: Probability improvement
- **EI**: Expected improvement
- **GP-UCB**: Gaussian process upper confidence bound

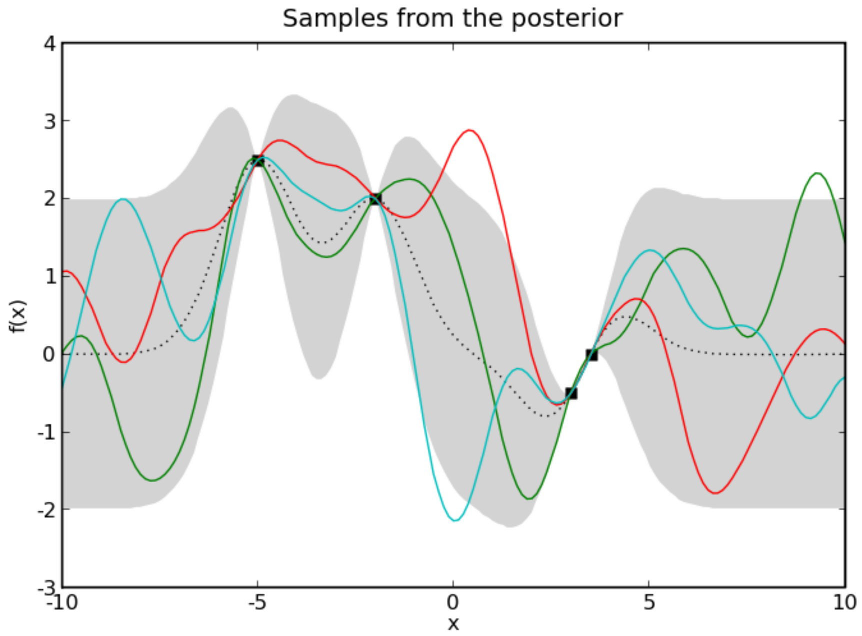


# Acquisition functions



# Thompson Sampling

- Sample a function from the Gaussian Process
- The function is sampled as a set of $f_*$ values from the distribution
$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left( \begin{matrix} \mu \\ \mu_* \end{matrix}, \begin{bmatrix} k & k_* \\ k_*^T & k_{**} \end{bmatrix} \right) \text{ given } f$$
- Chosen point is the optimum of the sampled function



# Extra reading

- https://www.youtube.com/watch?v=SQtOI9jsrJ0&list=PLCJPYIcPhgPAiiBjjVxi5St_lC_cXHFCK&index=9
- https://www.youtube.com/watch?v=4vGiHC35j9s

# What next?

- **Lecture**: Curriculum Learning
- **Assignments**:
  - DDPG, by No. 25, EoD
- **Quiz (on Canvas)**:
  - Imitation Learning, by Nov 25, EoD
- **Project**:
  - Final Report, by Dec. 2, EoD