Sensitivity Analysis of Deep Learning and Optimization Algorithms

Dr Varun Ojha

Department of Computer Science
University of Reading, UK
v.k.ojha@reading.ac.uk; vkojha@ieee.org
Github: https://github.com/vojha-code

at

LOD2022

The 8th International Conference on Machine Learning, Optimization, and Data Science

September 18 – 22, 2022

Siena – Tuscany, Italy

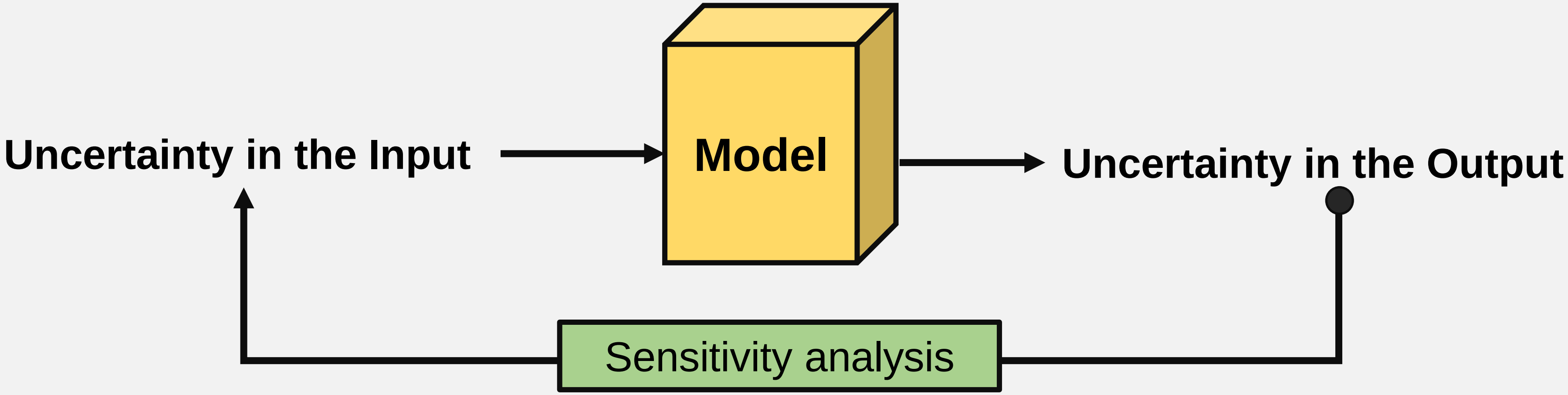
Content

- **Part 1: Sensitivity Analysis**
  - Sensitivity analysis methods
  - Algorithm configuration problem
  - Configuration selection methods
- **Part 2: Sensitivity Analysis of Deep Learning Algorithms**
  - Deep learning algorithms
  - Deep learning algorithm configuration space
  - Results of the analysis
- **Part 3: Sensitivity Analysis of Optimization Algorithms**
  - Optimization algorithms
  - Optimization algorithm configuration space
  - Results of the analysis
- Resources

Part 1
Sensitivity Analysis

Sensitivity analysis

The study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input (Saltelli et al., 2004)



Simple Example: linear model

$$\boldsymbol{Y} = \sum_{\boldsymbol{i}}^{\boldsymbol{n}} \boldsymbol{W_i} \boldsymbol{X_i}$$

where input factors are $\Omega = (W_1, W_2, \dots, W_n, X_1, X_2, \dots, X_n)$.

If we the **coefficients** $(W_1, W_2, \dots, W_n)$ **are fixed** then the model has variables $(X_1, X_2, \dots, X_n)$ are the only active factors.

Therefore, model outputs Y are sensitive to model inputs X.

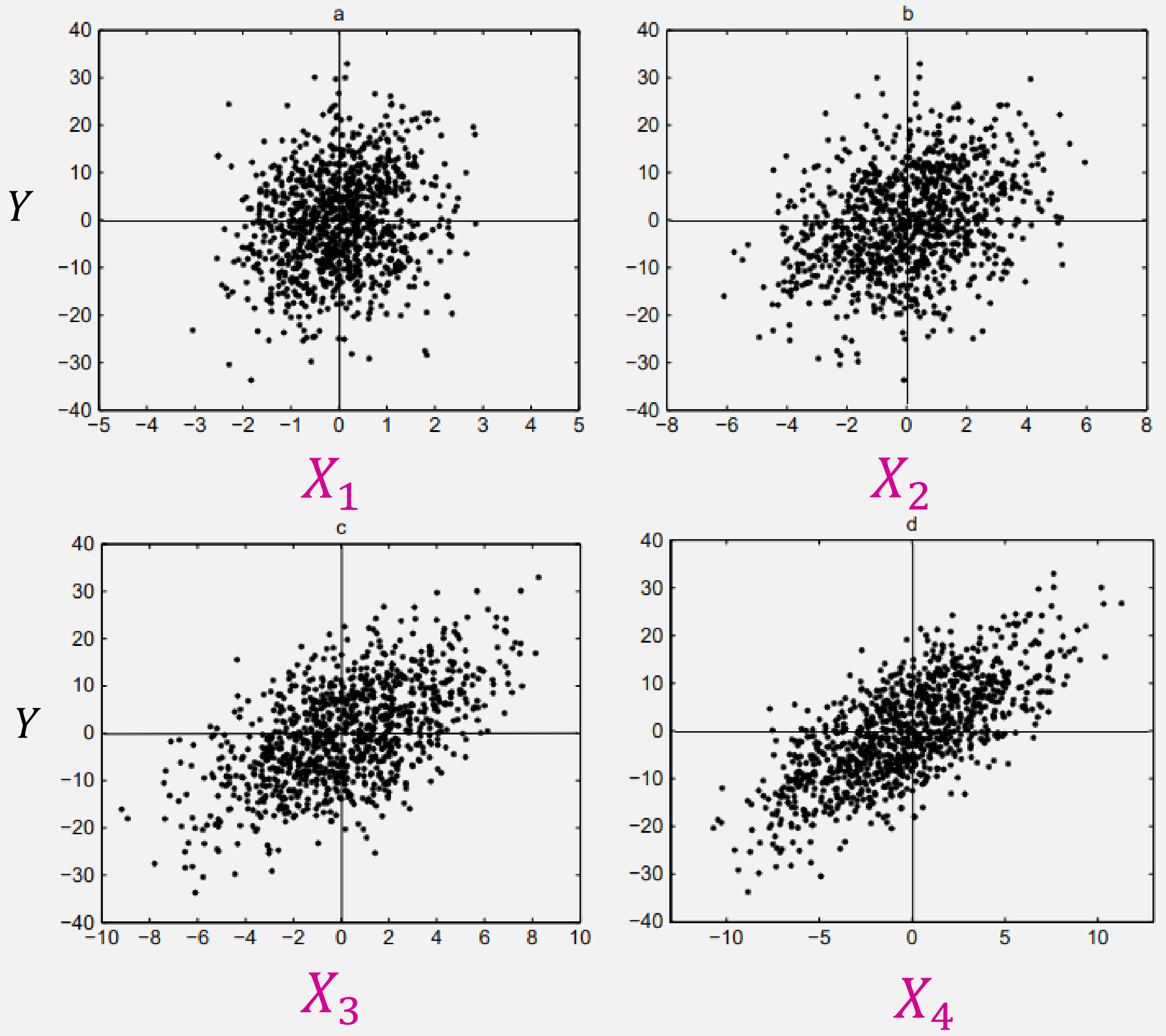
Which variable $(X_1, X_2, \dots, X_n)$ is the most influential?

Which is the most influential factor?

- Scatterplots of Y versus (X_1, X_2, X_3, X_4)

- The scatterplots show that Y is more sensitive to X_4 than it is to X_3, and that the ordering of the input factors by their influence on Y is

$$X_4 < X_3 < X_2 < X_1$$



Conditional Variances (First Order measure)

- For a model

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_n})$$

we wish to determine what would happen to the uncertainty of Y if we could fix a factor X_i at a value x_i^*.

We would imagine that the resulting variance $V_{X_{\sim i}}(Y \mid X_i = x_i^*)$ will be less than the total or unconditional variance $V(Y)$.

Limitation: the sensitivity measure depend on a value x_i^*.

Conditional Variances

Avoid the the sensitivity measure dependence on a value x_i^**.*

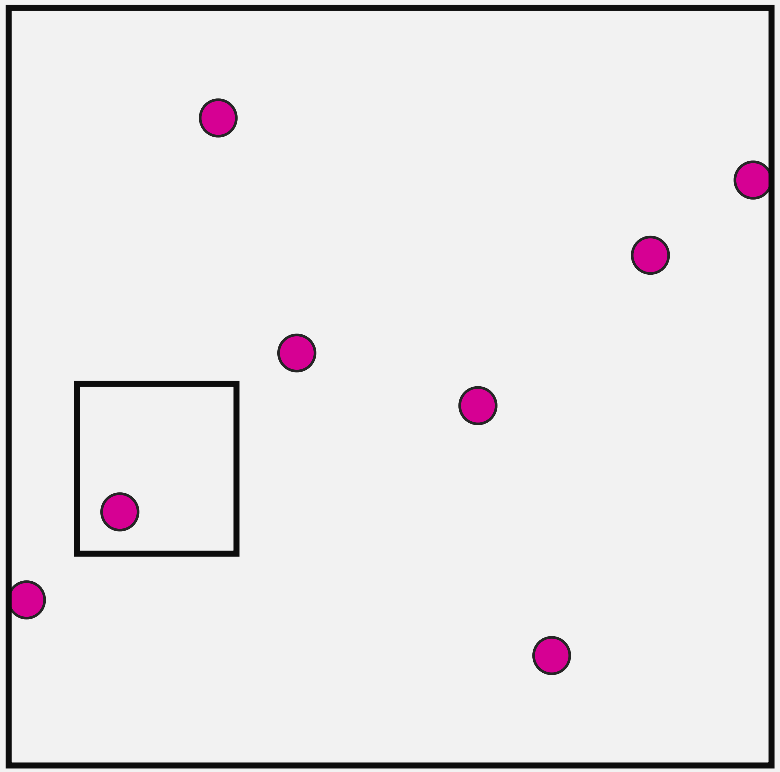
We take average over all values of values of X_i and NOT just *a fixed value* x_i^* : $E_X\ (V_{X\sim i}(Y\ |X_i))$. And we have averaging over all-but-X_i as $E_{X\sim i}(Y\ |X_i)$.

Therefore, the conditional variance $V_{X_i}\ (E_{X\sim i}(Y\ |X_i)) \leq V(Y)$, i.e., the conditional variance is less than the variance of model on all total or unconditional variance $V(Y)$.

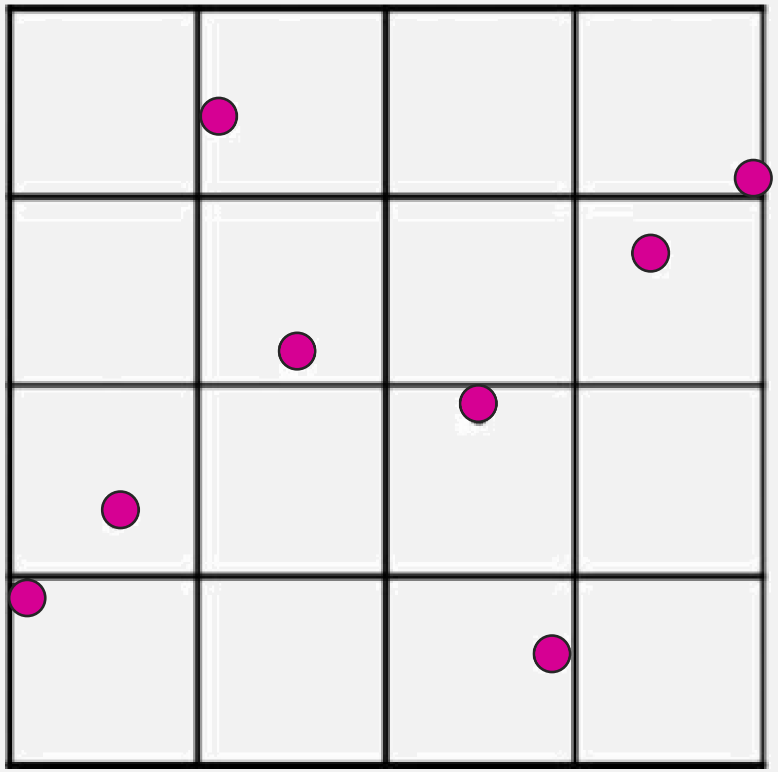
This gives us the sensitivity measure S_i of variable X_i as

$$S_i = \frac{V_{X_i}\ (E_{X\sim i}(Y\ |X_i))}{V(Y)}$$

How to sample values of variable X



random sampling



gird sampling / One at a time (OTA) sampling

Sensitivity Analysis: Elementary Effect (EE)

$$EE_i = \frac{[Y(X_1, \dots, X_i + \Delta, \dots X_k) - Y(X_1, \dots X_k)]}{\Delta}$$

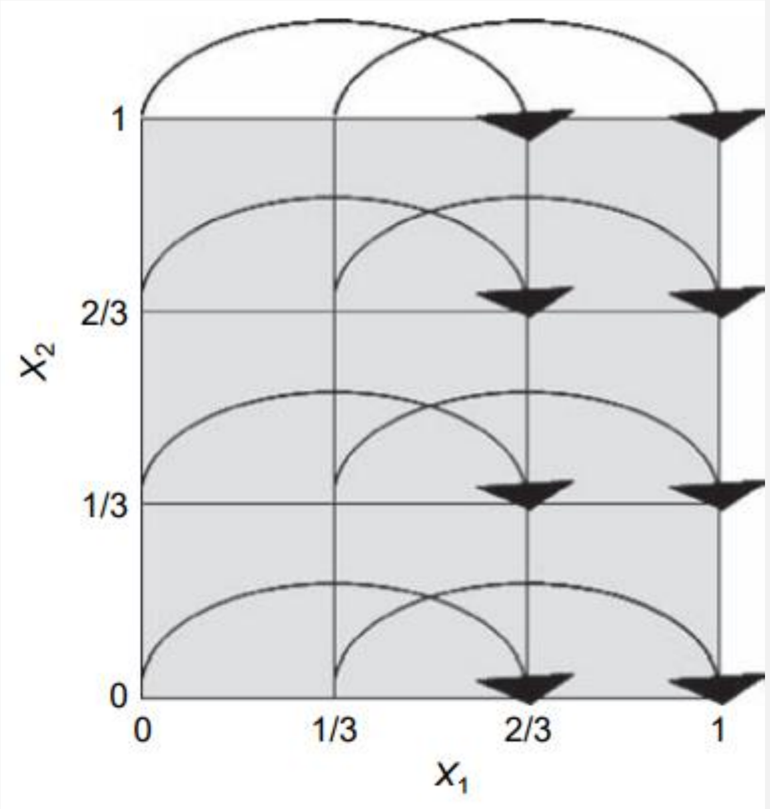
For r sample points, the sensitivity measures are:

Means μ of EE

$$\mu_i = \frac{1}{r}\sum_{j}^{r} EE_i^j$$

Standard deviation σ of EE

$$\sigma = \sqrt{\frac{1}{r-1}\sum_{j}^{r}(EE_i^j - \mu_i)^2}$$



four-level grid ($p = 4$) in the two-dimensional input space ($k = 2$), $\Delta = p/(2(p-1))$

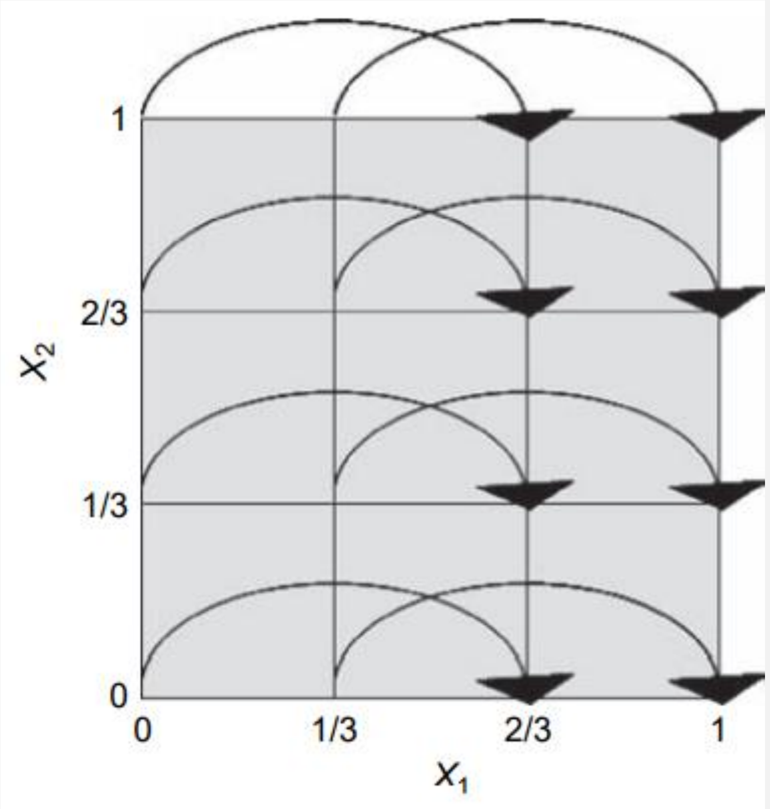
Sensitivity Analysis: Total Effect (EE)

First order Effect

$$S_i = \frac{V\left(E(Y|X_i)\right)}{V(Y)}$$

Total Effect

$$S_{T_i} = 1 - \frac{V\left(E(Y|X_{\sim i})\right)}{V(Y)}$$



four-level grid ($p = 4$) in the two-dimensional input space ($k = 2$), $\Delta = p/(2(p-1))$

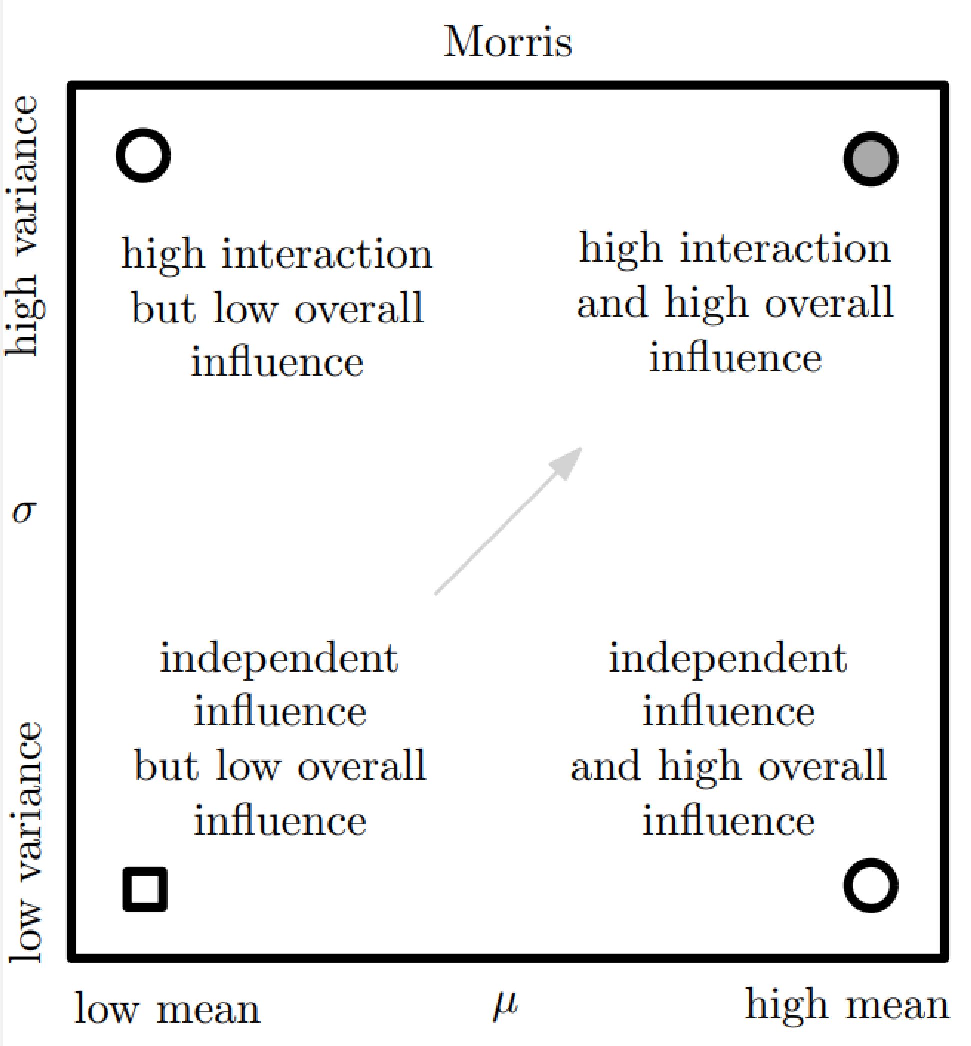
Sensitivity Analysis: Interpretation

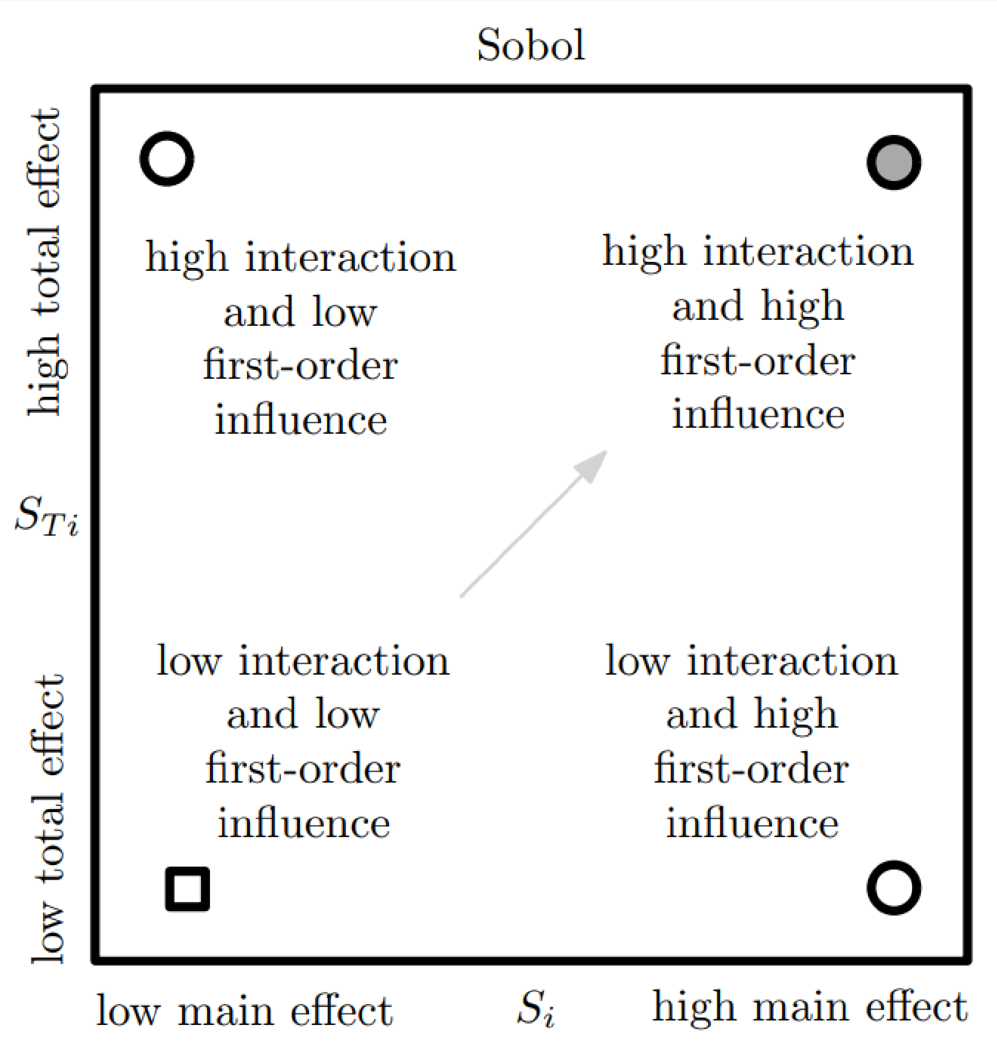
- Morris Method (Elementary Effect)
  - Mean μ
    - Low value – the variable X has low overall influence on Y
    - High value – the variable X has high overall influence on Y
  - Standard deviation σ
    - Low value - the variable X has low influence independently on Y
    - High value - the variable X has high interactive influence on Y

Sensitivity Analysis: Interpretation

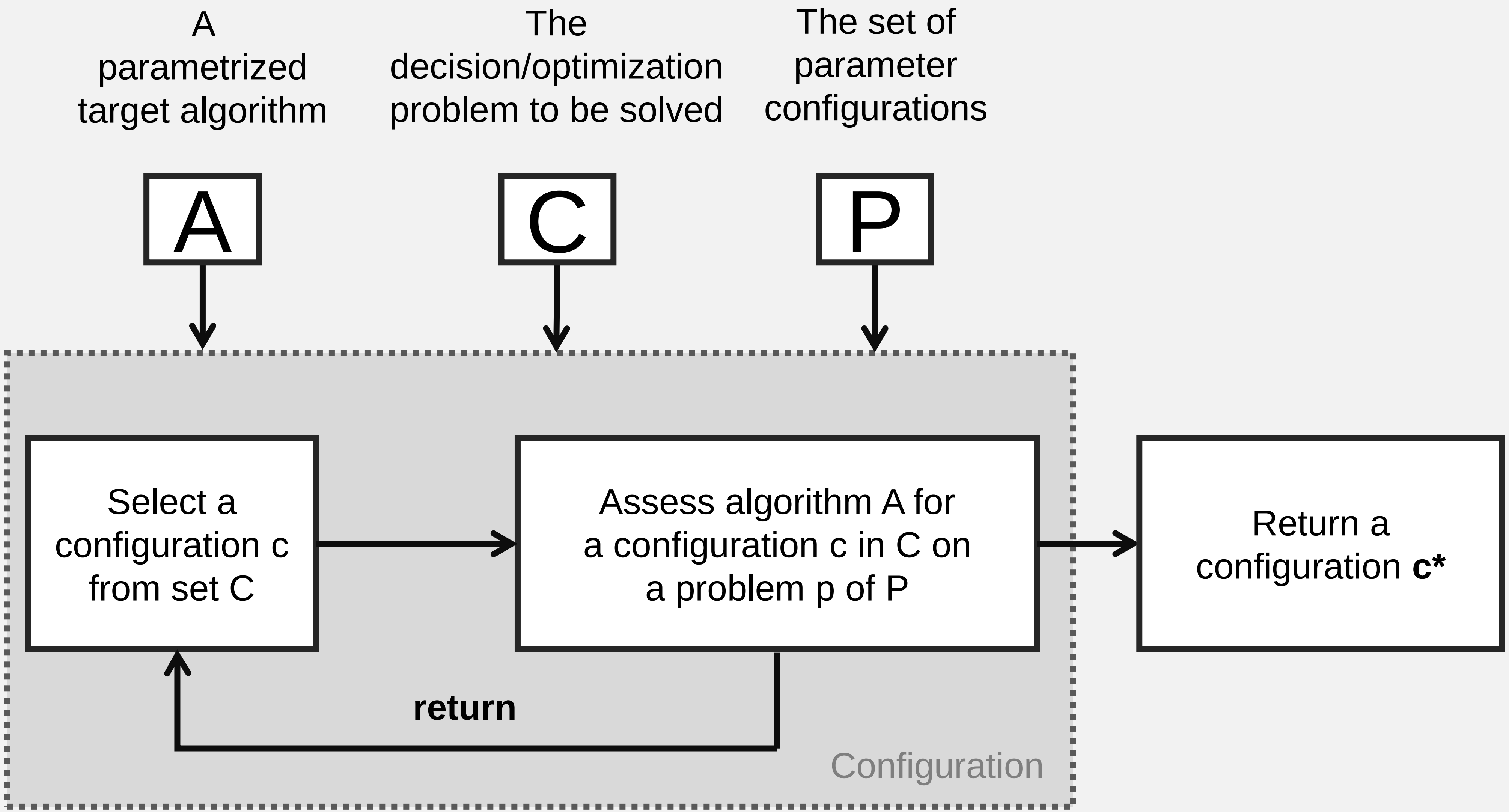
- Sobol Method (Variance Based / Total Effect)
  - First order effect
    - Low value – the variable X has low direct influence on Y
    - High value – the variable X has high direct influence on Y
  - Total effect
    - Low value - the variable X has low total influence on Y
    - High value - the variable X has high interactive influence on Y

Sensitivity Analysis: Interpretation

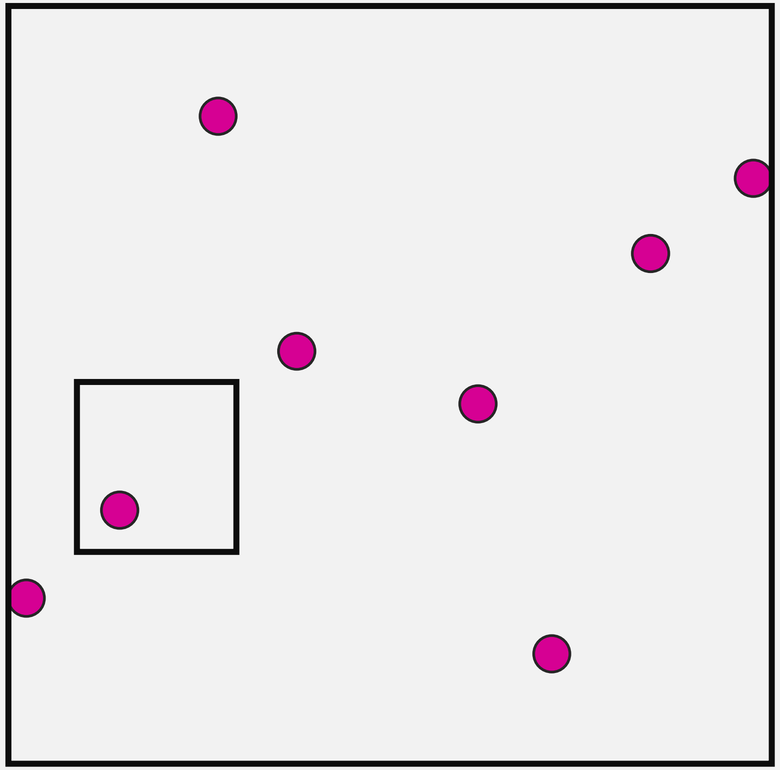




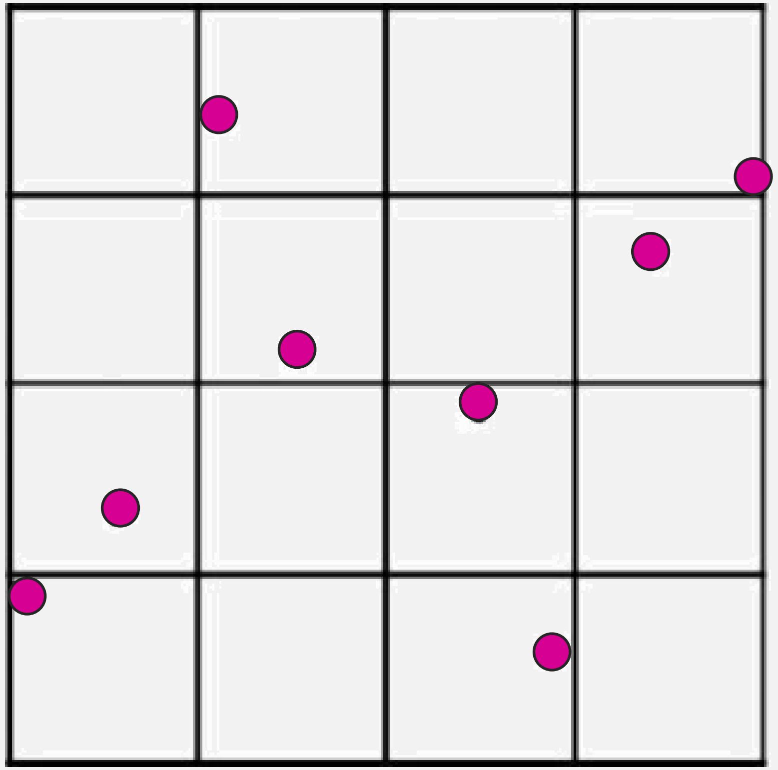
Algorithm Configuration Problem



Selection of configuration c from C



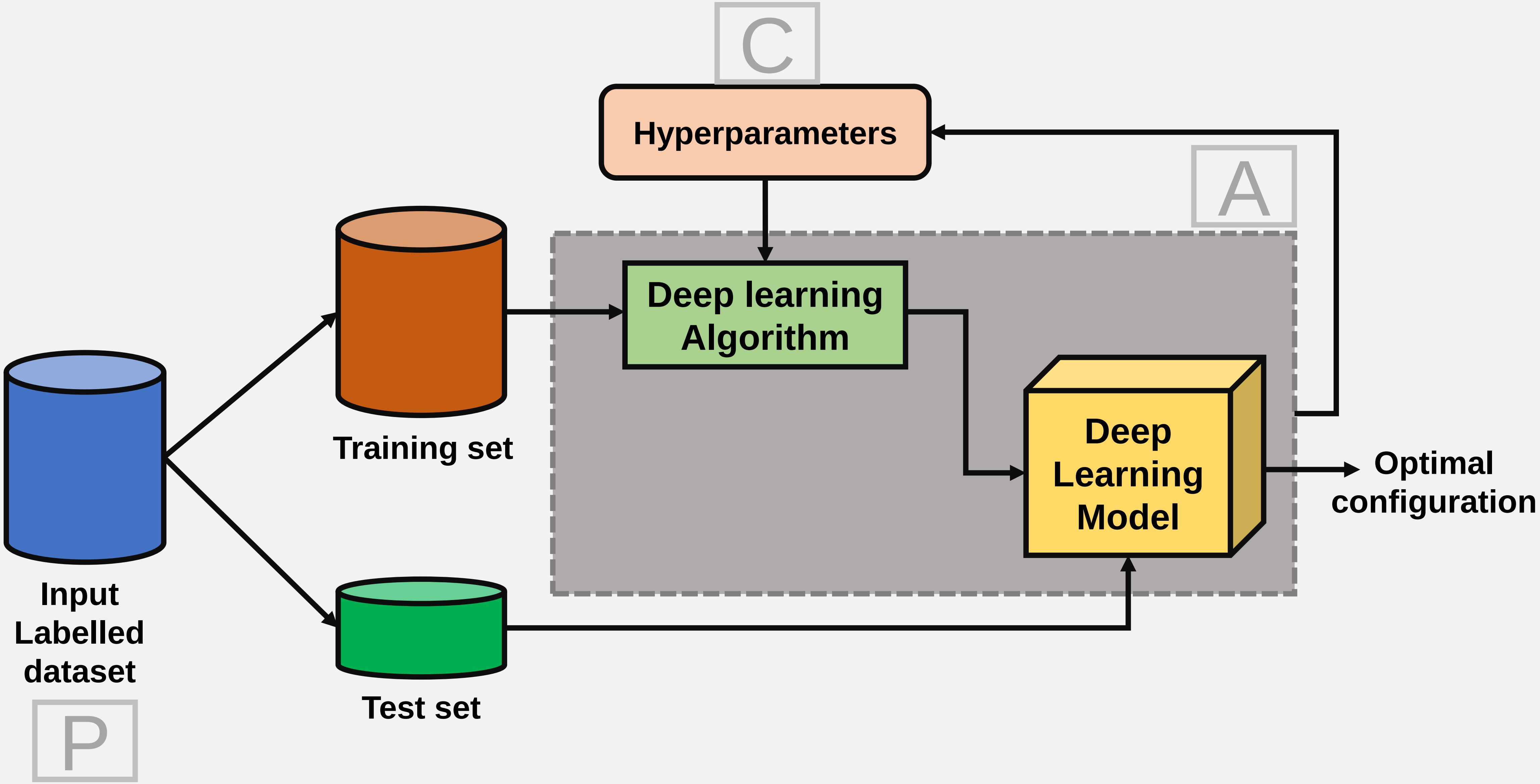
random sampling



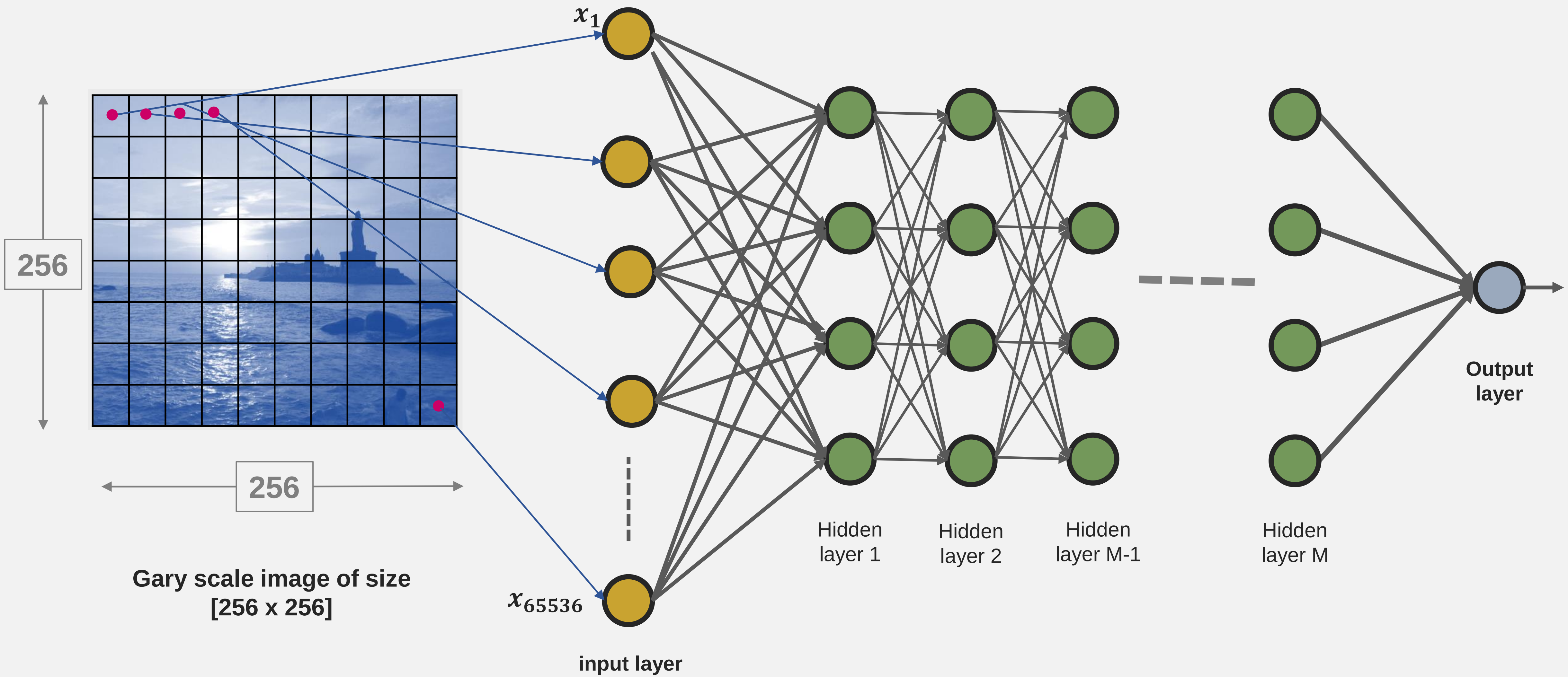
gird sampling / One at a time (OTA) sampling

Part 2
Sensitivity Analysis of Deep Neural Networks

Deep Learning Algorithm



Algorithm: Deep Learning

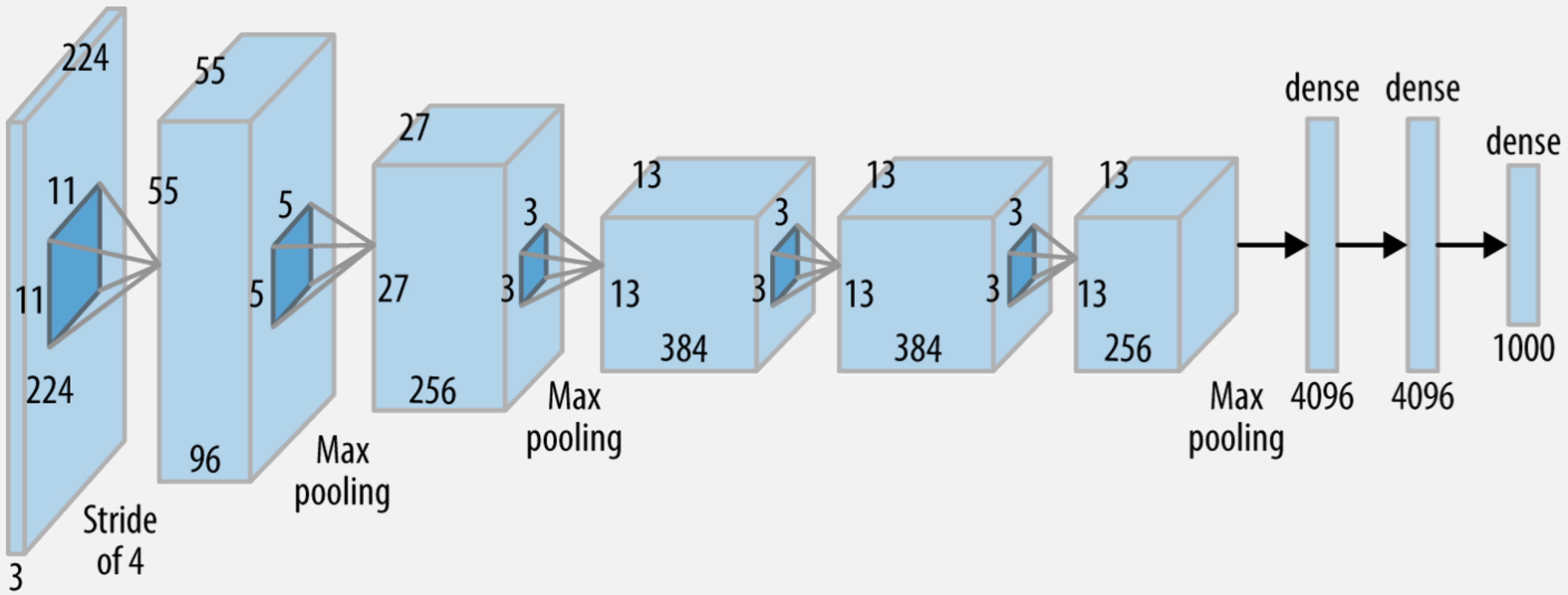


Configuration: Deep Learning

- **Network Architecture**
  - Number of layers
  - Number of nodes per layer
  - Type of layers
- **Activation functions**
  - Type of activation function
- **Learning algorithms**
  - Type of optimizers
  - Learning mode
  - Learning epochs
  - Hyperparameters of optimizers (e.g., Learning rate)

Algorithm: Deep Neural Network

- Deep Neural Network
- ResNet18
- AlexNet
- GoogleNet



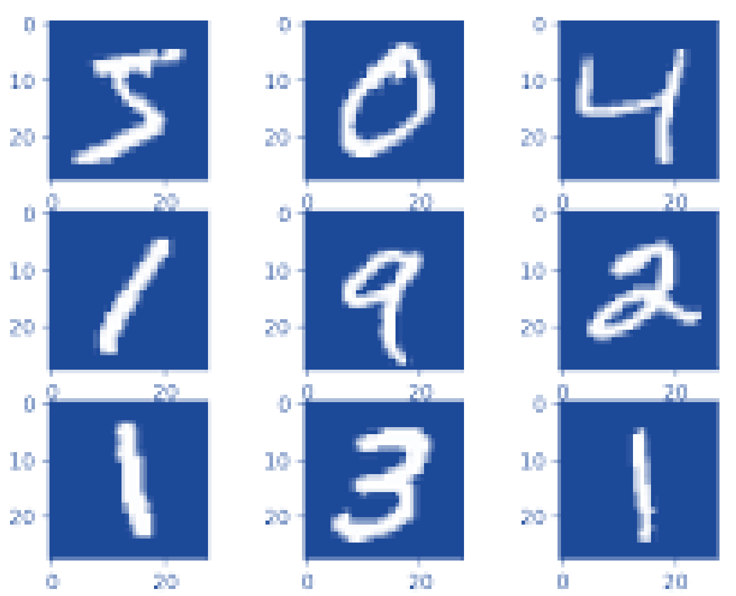
Example: AlexNet Block Diagram

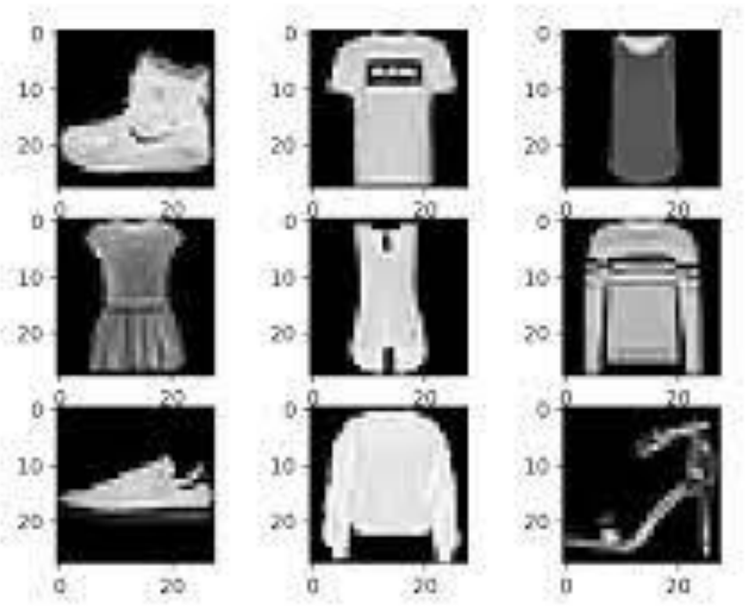
Configuration: Deep Learning

Parameter	Description	Range	Default
Optimiser	List of gradient descent (GD) algorithms.	Category*	Adam
Learning rate (α)	Initial GD step controller.	[1x10^{-7}, 0.5]	0.001
Momentum (β)	Acceleration factor for GD.	[0, 0.99	0.6
Learning rate decay (α_{decay})	Reduction rate of (α).	[0, 1]	0.9
Learning rate decay step (α_{d-step})	Number of epochs between Learning Rate Decay.	[1,100]	10
Batch size	Size of training subset for GD update.	Category*	32
Epochs	Number of training cycles.	[5, 1000]	100

Note: *Optimisers variations: Adam, SGD, RMSprop, ADAdelta, ADAgrad and ADAmax;*
Batch size variations : 1, 32, 64 and 128

Problems: Deep Learning





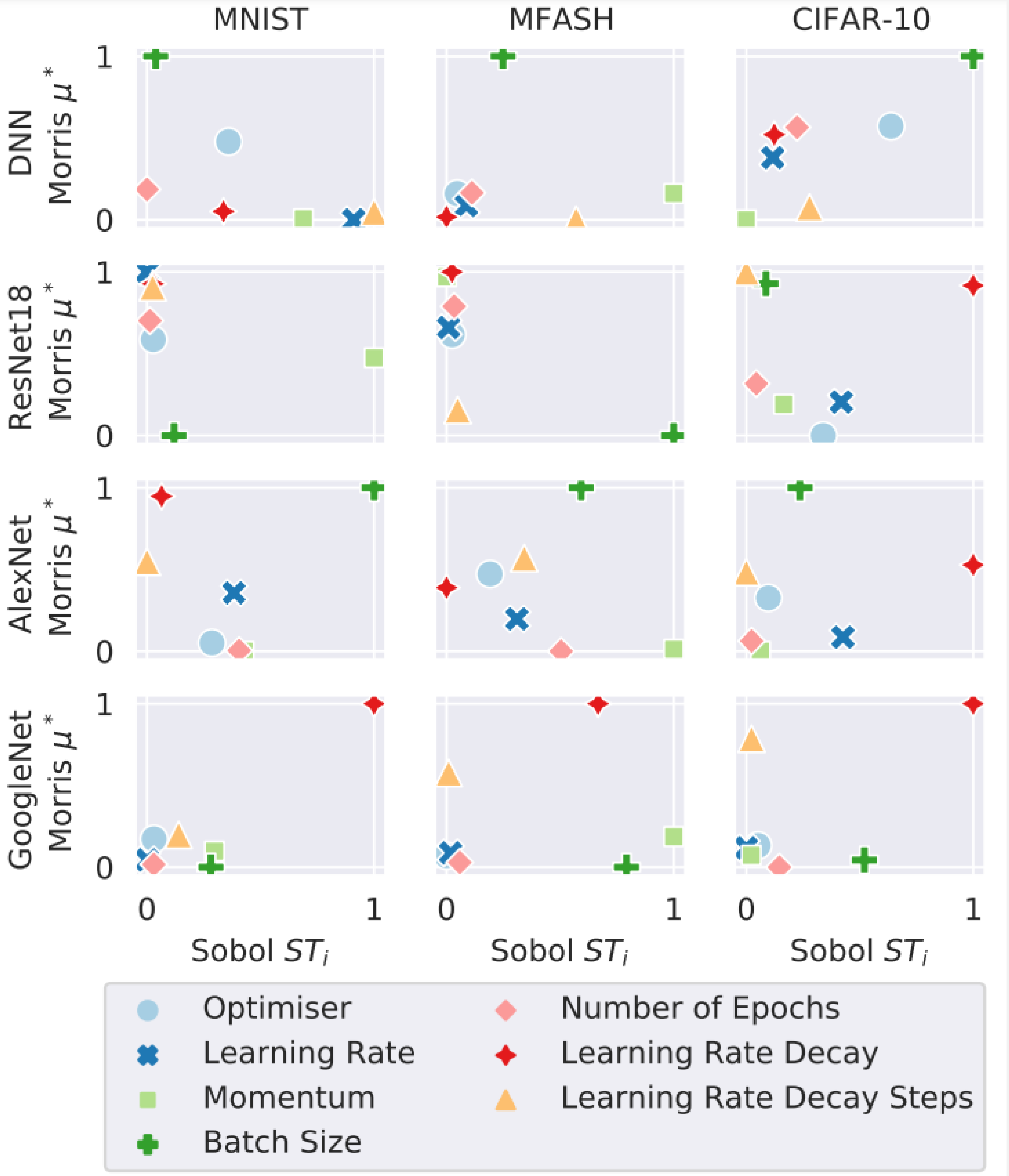
MNIST

Fashion MNIST

CIFAR-10

Sensitivity analysis summary

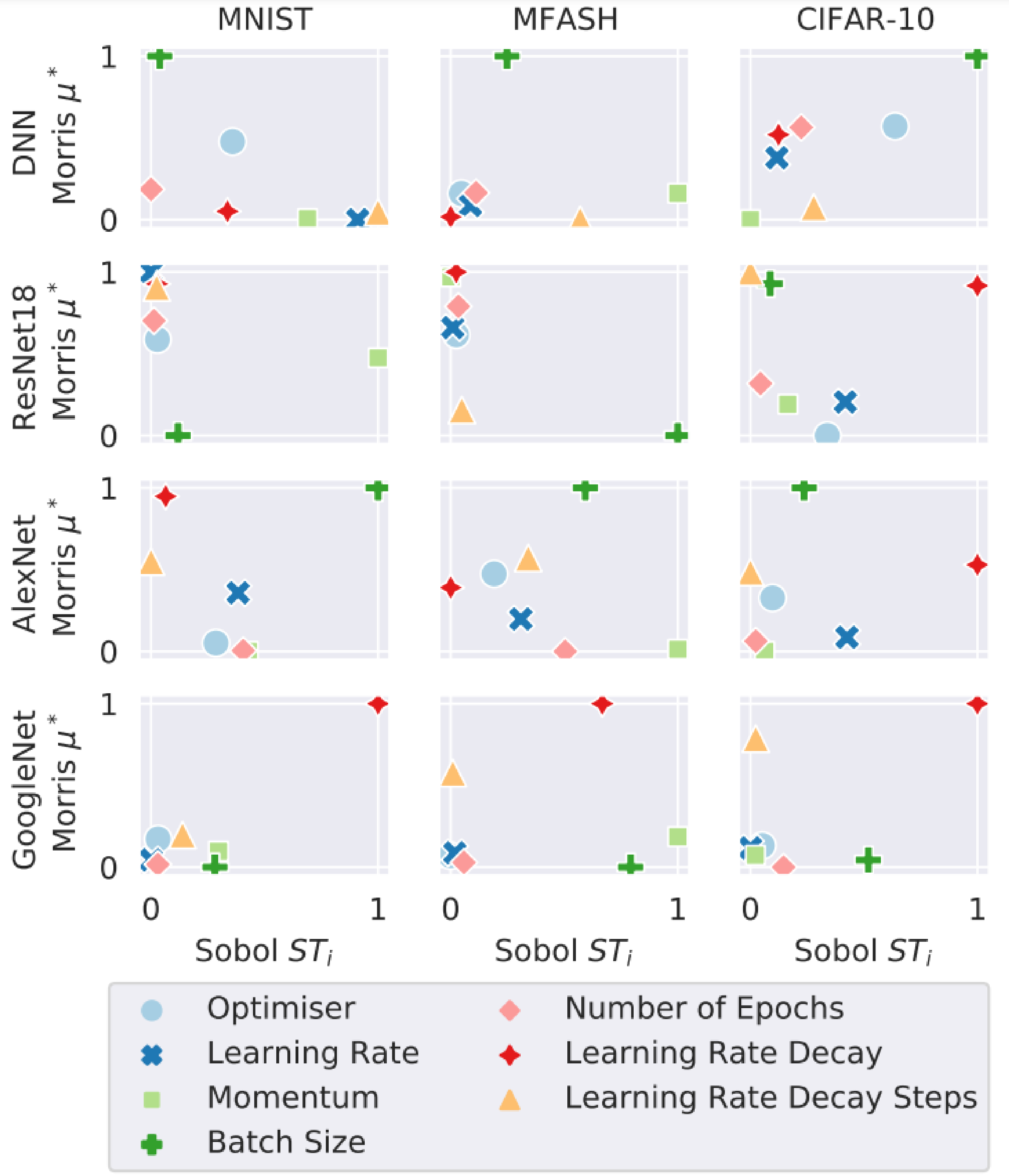
- Type of gradient decent optimizer is not a major factor on DNN
- Learning rate is not a major factor on DNN whereas the learning rate decay is.
- Number of epochs is relatively least influential



Sensitivity analysis summary

- Learning rate decay is the most influential for fixed network architecture models

- Batch size is the most influential for flexible network architecture model



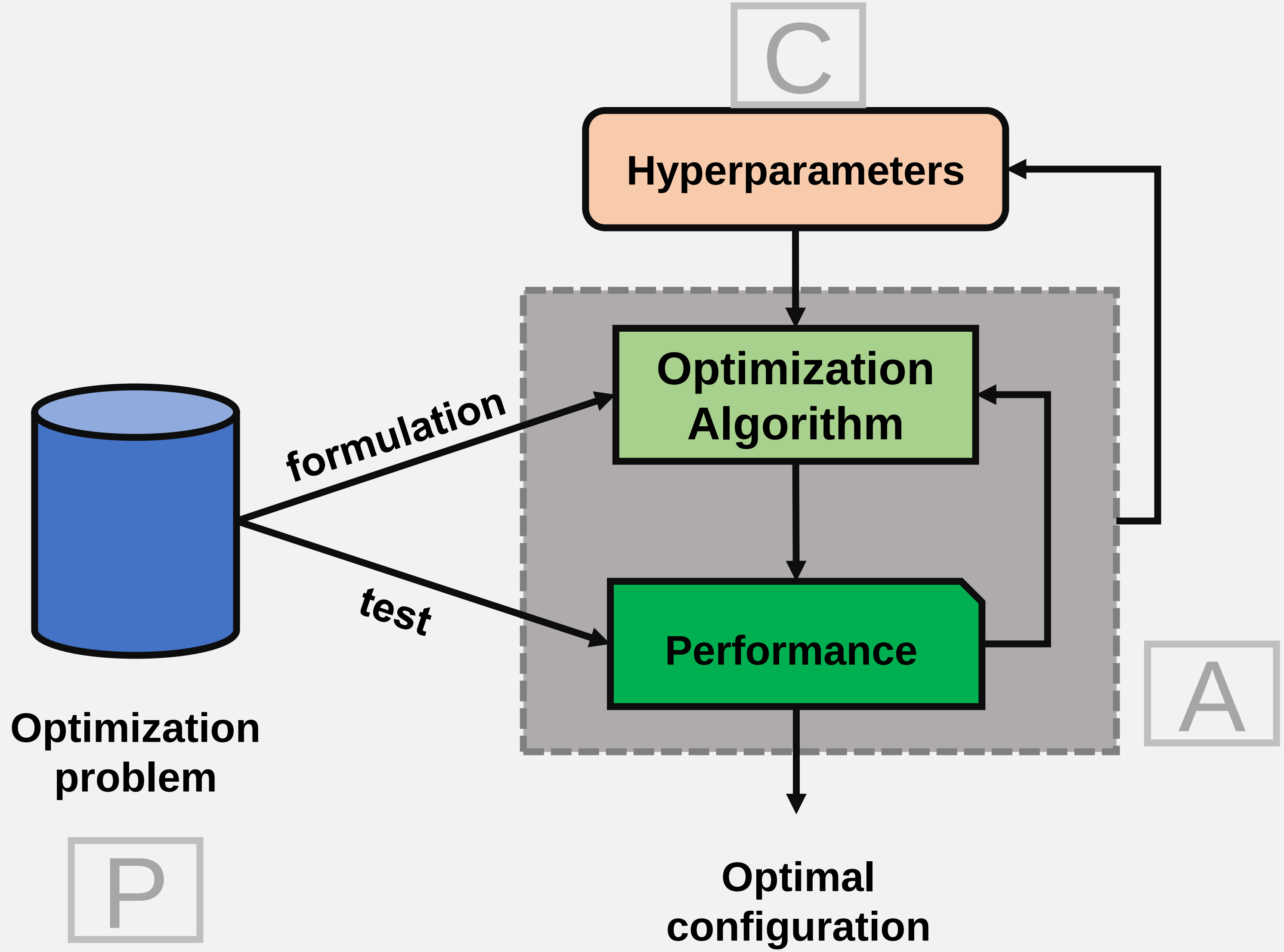
DNN Sensitivity analysis summary

	DNN			ResNet18			AlexNet			GoogleNet			
Parameter	M	MF	C	M	MF	C	M	MF	C	M	MF	C	Average
Learning Rate Decay	1.16	1.40	1.00	0.98	**0.98**	**0.08**	0.94	1.17	**0.47**	**0.00**	**0.33**	**0.00**	**0.71**
Batch Size	0.96	**0.75**	**0.00**	1.33	1.00	0.92	**0.00**	**0.41**	0.76	1.23	1.02	1.07	0.79
Learning Rate Decay Steps	0.95	1.09	1.17	0.98	1.27	1.00	1.10	0.79	1.12	1.18	1.08	1.00	1.06
Momentum	1.04	0.84	1.41	**0.52**	1.00	1.16	1.15	0.99	1.37	1.14	0.81	1.35	1.07
Optimiser	**0.83**	1.27	0.56	1.06	1.05	1.20	1.19	0.96	1.13	1.27	1.37	1.28	1.10
Learning Rate	1.00	1.29	1.08	1.00	1.05	0.98	0.89	1.06	1.08	1.38	1.34	1.33	1.12
Epochs	1.29	1.22	0.89	1.03	0.99	1.17	1.16	1.12	1.35	1.38	1.35	1.31	1.19

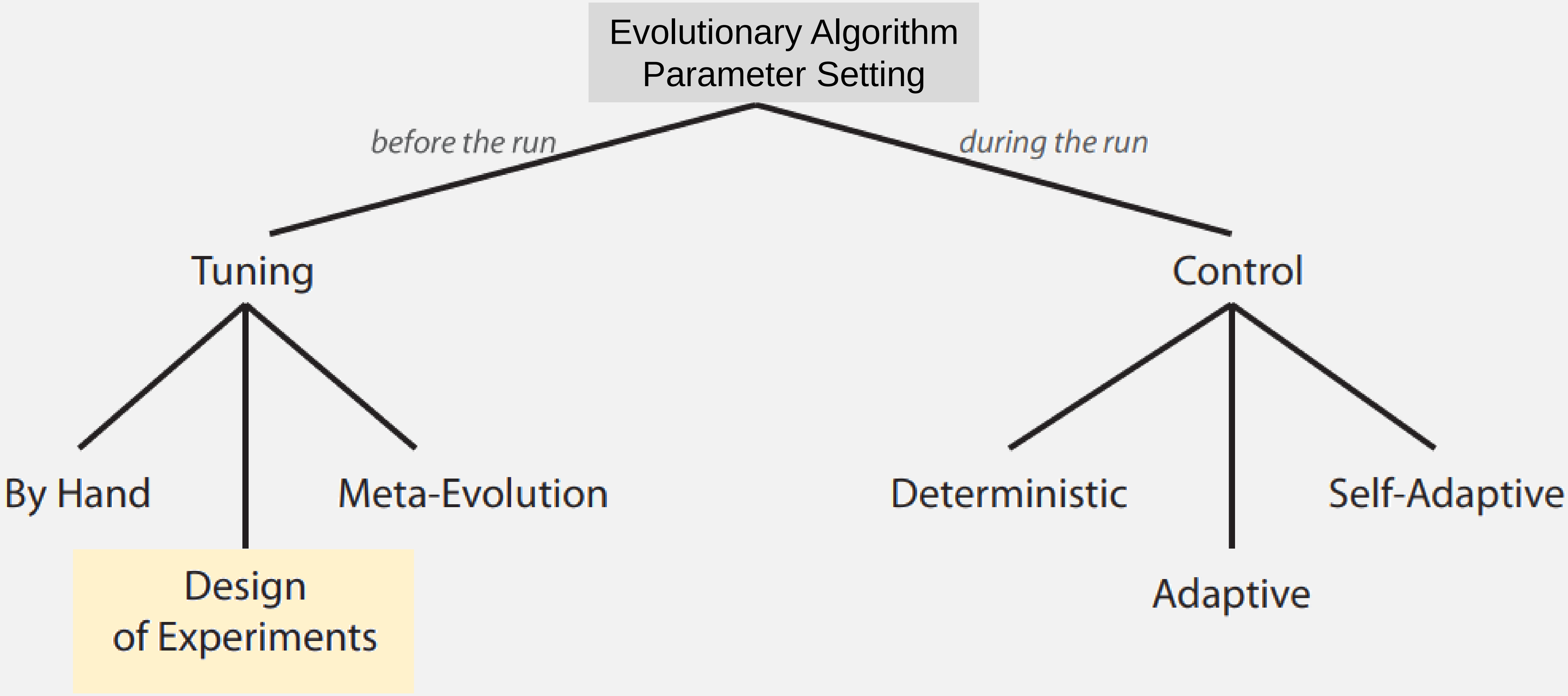
Note: Dataset names abbreviated in above table as M for MNIST, MF for MNIST Fashion and C for CIFAR-10.

Part 3
Sensitivity Analysis of Optimization Algorithms

Optimization Algorithms



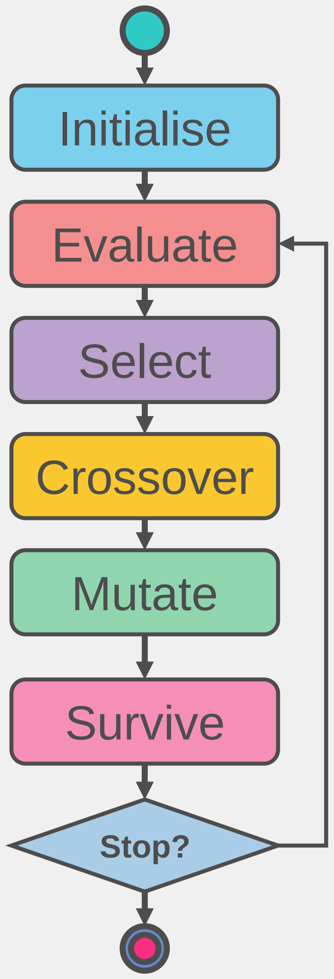
Optimization Algorithms



Source: Kramer, O.: Evolutionary self-adaptation: a survey of operators and strategy parameters. Evolutionary Intelligence 3, 51–65 (2010)

Evolutionary Algorithms (EAs) - STEPS

1. **t** := **0**; // Generation 0
2. Generate **Initial Population** **P**$^{(t)}$ at random;
3. **Evaluate the fitness** of each individual in **P**$^{(t)}$;
4. **Until** (termination condition **not** met) **do**
   1. **Select parents**, **Pa**$^{(t)}$ from **P**$^{(t)}$ based on their fitness in **P**$^{(t)}$;
   2. Apply **crossover (recombination)** to create offspring from parents: **Pa**$^{(t)}$ → **O**$^{(t)}$
   3. Apply **mutation** to the offspring: **O**$^{(t)}$ → **O**$^{(t)}$
   4. **Evaluate the fitness** of each individual in **O**$^{(t)}$;
   5. **Survive population** **P**$^{(t+1)}$ from current offspring **O**$^{(t)}$ and parents **P**$^{(t)}$;
   6. **t** := **t** + **1**; // Next generation
5. **end-do**



Versions of Evolutionary Algorithms

- Single objective EAs – solve only one objective

$$f: \quad \mathbb{R}^n \to \mathbb{R}$$
$$\mathbf{x} \mapsto f(\mathbf{x})$$

- Multi-objective EAs – solve only two or more objectives simultaneously

$$F(\mathbf{x}) \equiv (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})), \text{ i.e., } F: \mathbb{R}^n \to \mathbb{R}^k \text{ for } k \geq 2$$

- X is decision variable of the problem, k is objectives

Metric for Single Objective EA

$$
\begin{aligned}
f: \quad & \mathbb{R}^n \to \mathbb{R} \\
& \mathbf{x} \mapsto f(\mathbf{x})
\end{aligned}
$$

Optimal solution is the one that give global minimum value of the problem f, e.g., this could a value of 0.

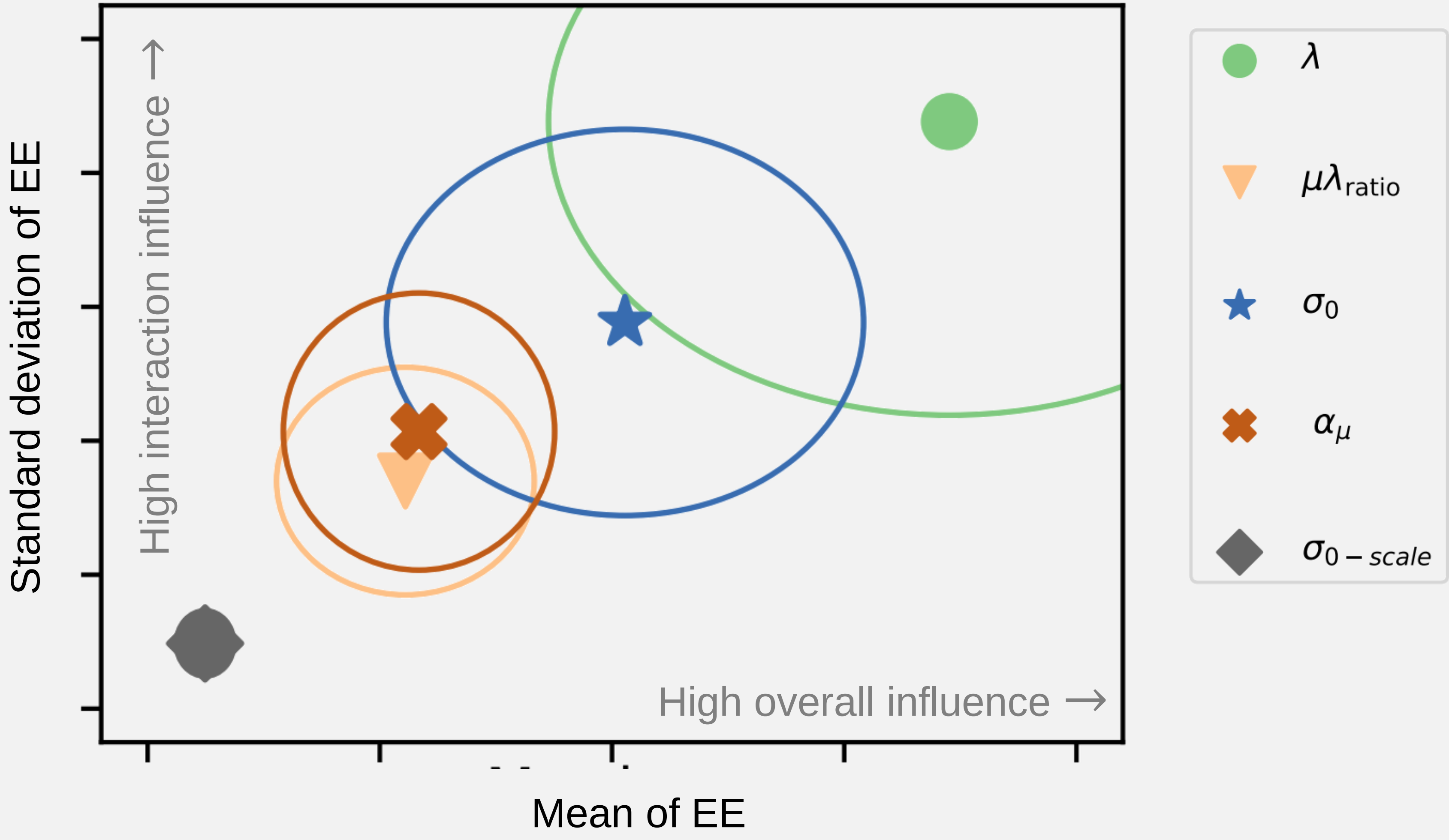
Most Popular Evolutionary Algorithms

- **Single objective EAs** – solve only one objective
  - Differential Evolution (DE)
  - Covariance Matrix Adaptation Evolution Strategies (CMA-ES)
- **Multi-objective EAs** – solve only two or more objectives simultaneously
  - Non-Dominated Sorting Genetic Algorithm–III (NSGA-III)
  - Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D)

Single Objective EAs - Hyperparameters

Algo	Params	Domain	Description
CMA-ES	λ	$[10, 1000]$	Population size
	α_μ	$[0, 4]$	Learning rate
	σ_0	$[0.1, 2]$	Initial step size
	$\sigma_{0-scale}$	{False, True}	Re-scaling of σ_0: convergence speed controller
	$\mu\lambda_{\text{ratio}}$	$[0.1, 1]$	Percentage of population's elements usage in co-variance matrix estimation and update
DE	λ	$[10, 1000]$	Population size
	X	{bin, exp}	Crossover methods: Binomial and Exponential
	$P[\text{X}]$	$[0, 1]$	Crossover probability
	β_{min}	$[0, 1]$	Minimum Acceleration coefficient
	β_{max}	$[0, 2]$	Maximum Acceleration coefficient, $\beta_{\text{max}} = \beta_{\text{min}} + \beta_{\text{max}}$
	$\mathbf{b}_{\text{type}}$	{"best," "target-to-best," "rand-to-best," "rand"}	Base vector selection methods (mutation type or DE algorithm version)
	$\mathbf{b}\lambda_{\text{ratio}}$	$[0.01, 0.5]$	Percentage of base vectors (solution) to be used for difference vectors computation

Covariance Matrix Adaptation Evolution Strategies Sensitivity to its Hyperparameters



Population size

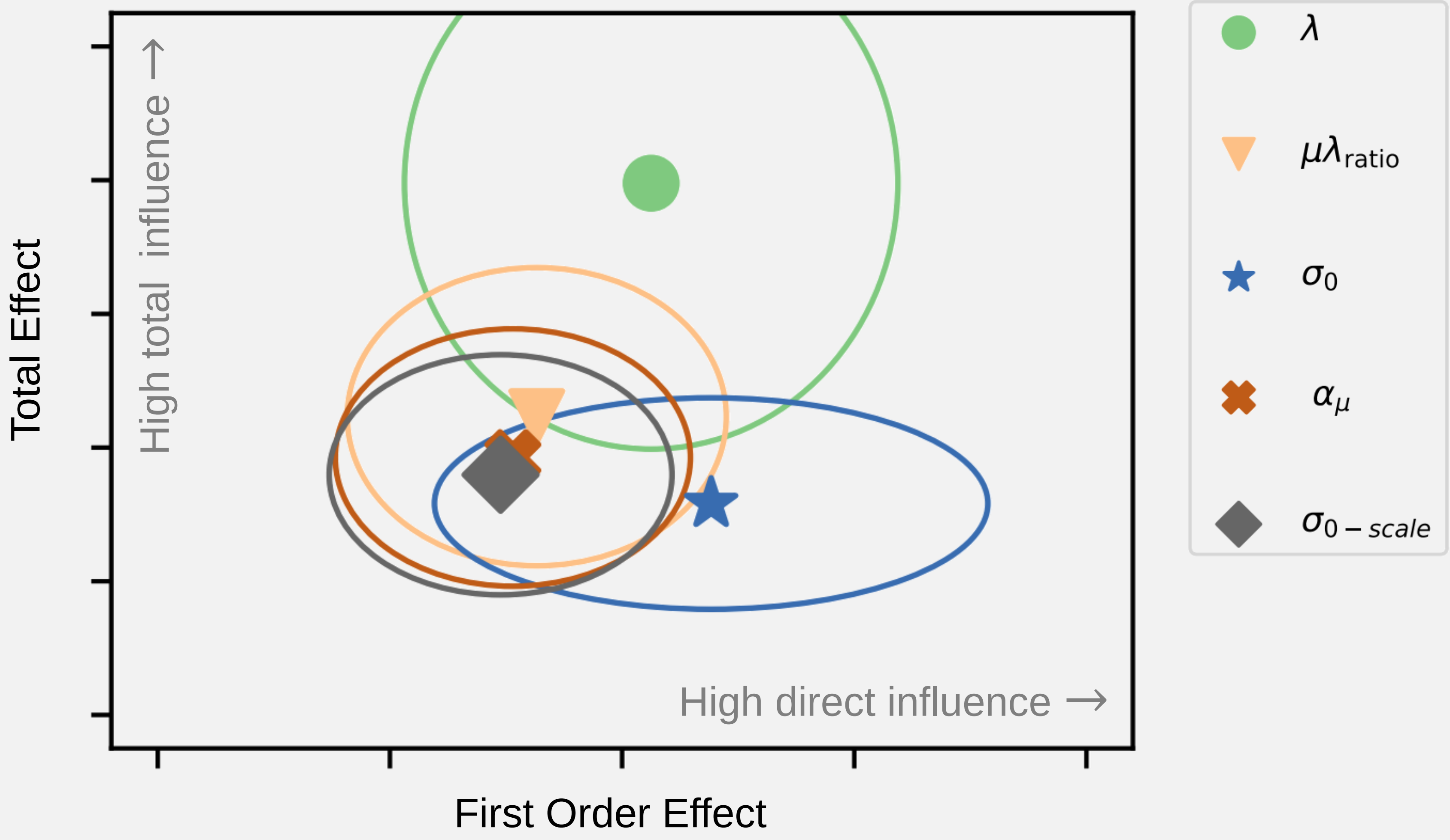
Vectors for co-variance matrix estimation

Initial step size

Learning rate

convergence speed controller

Covariance Matrix Adaptation Evolution Strategies Sensitivity to its Hyperparameters



Population size

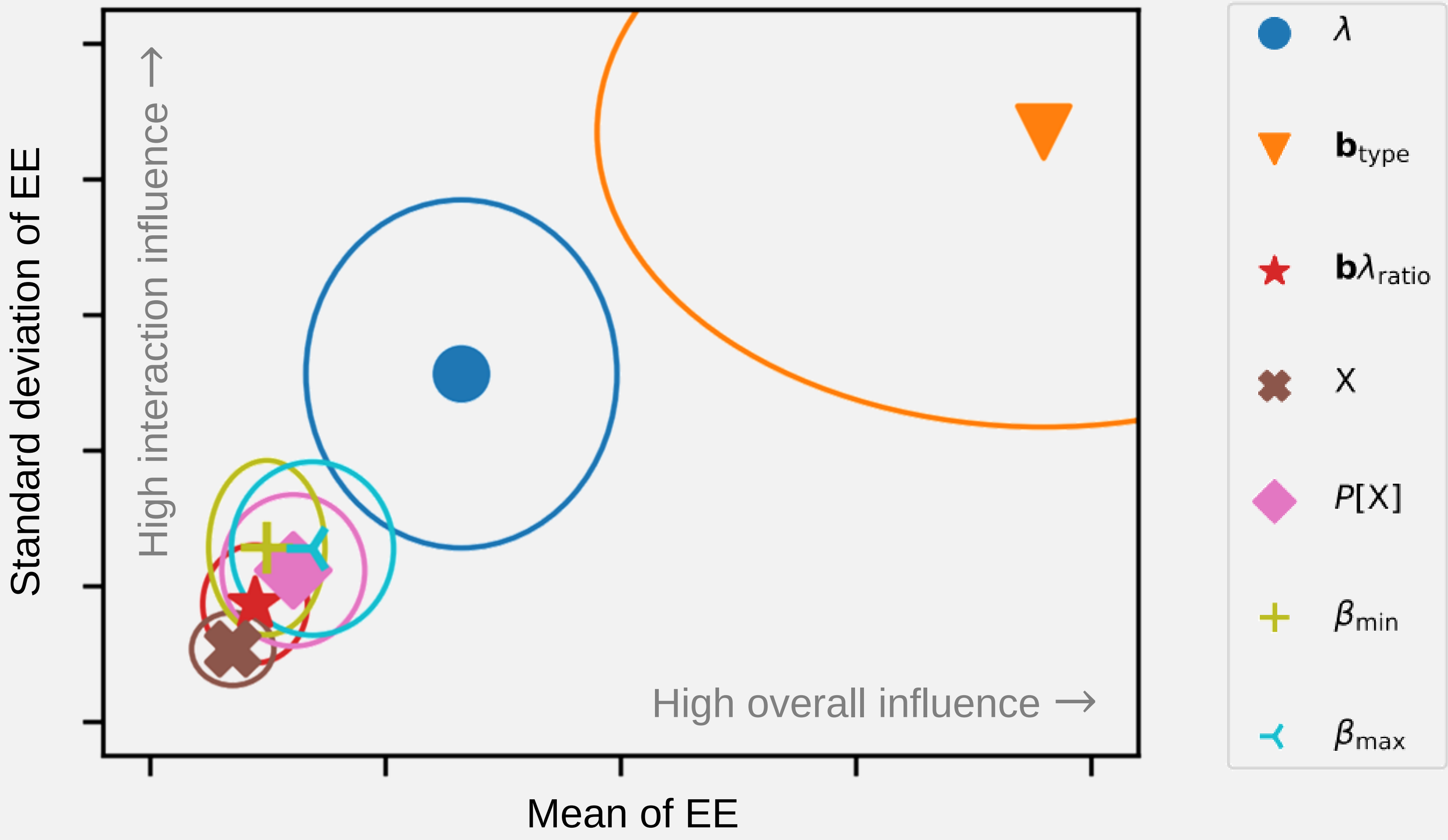
Vectors for co-variance matrix estimation

Initial step size

Learning rate

convergence speed controller

Differential Evolution (DE) Sensitivity to its Hyperparameters



Population size

Mutation type

Size of base vectors

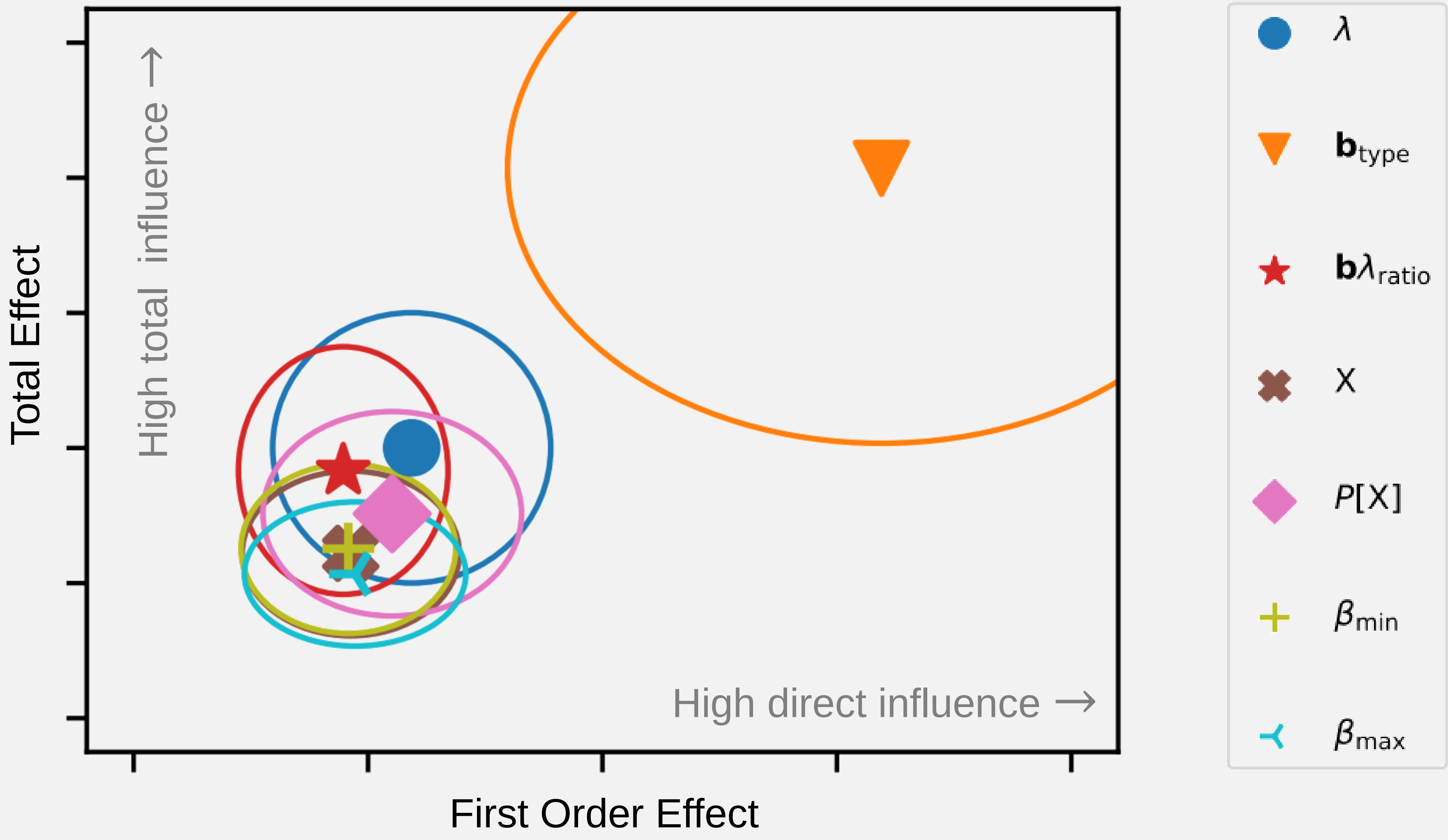
Crossover type

Crossover probability

Min Acceleration coefficient

Max Acceleration coefficient

Differential Evolution (DE) Sensitivity to its Hyperparameters



Population size

Mutation type

Size of base vectors

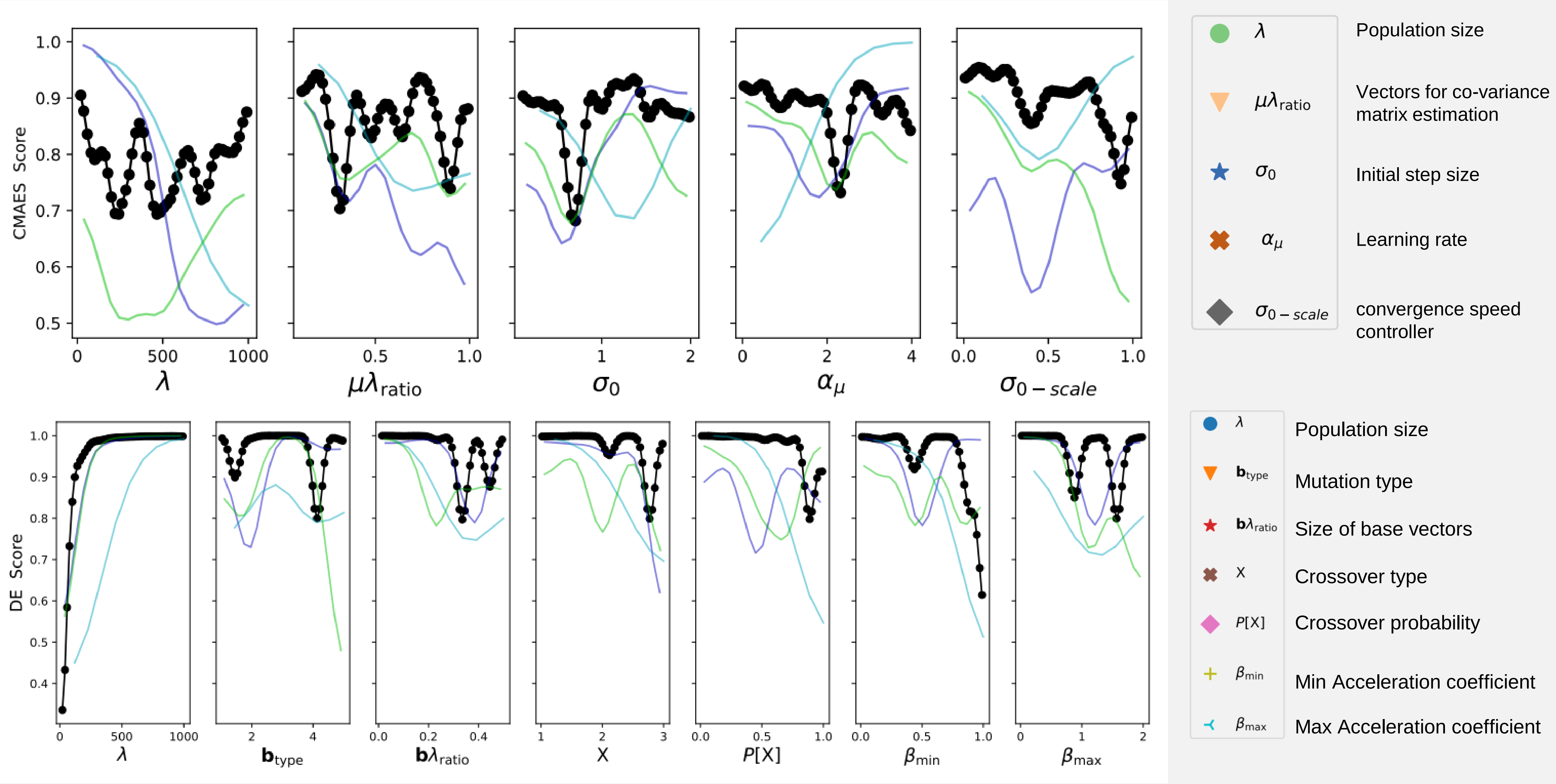
Crossover type

Crossover probability

Min Acceleration coefficient

Max Acceleration coefficient

Hyperparameter Influence Summary

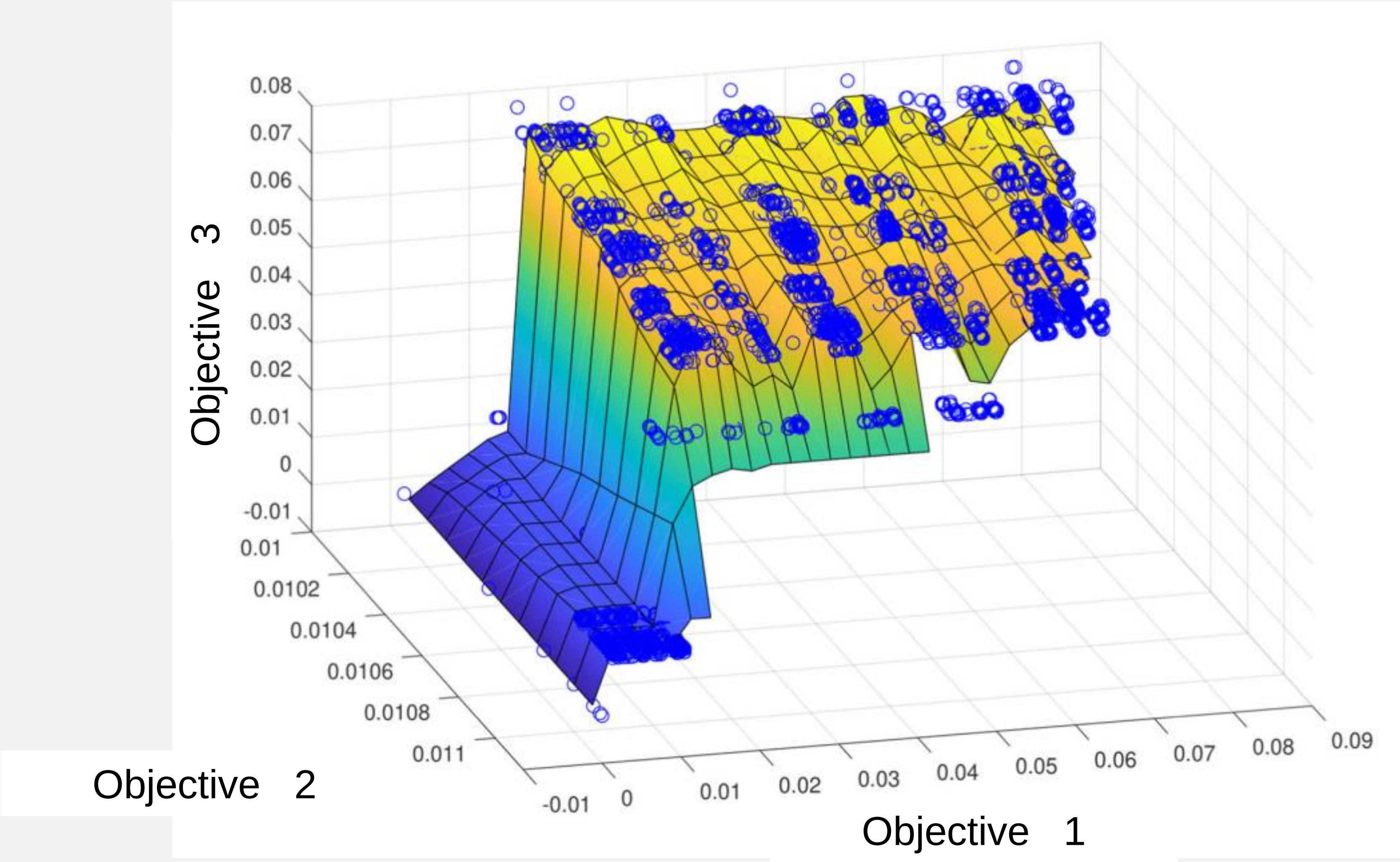


Order of Turning: Single Objective EAs

- Covariance Matrix Adaptation Evolution Strategies
  - Population size
  - Size of covariance metrics
  - Initial step size
  - Learning rate
  - Convergence speed controller

- Differential Evolution (DE)
  - Mutation type
  - Population size
  - Probability of crossover
  - Base vector size
  - Acceleration coefficient settings
  - Crossover type

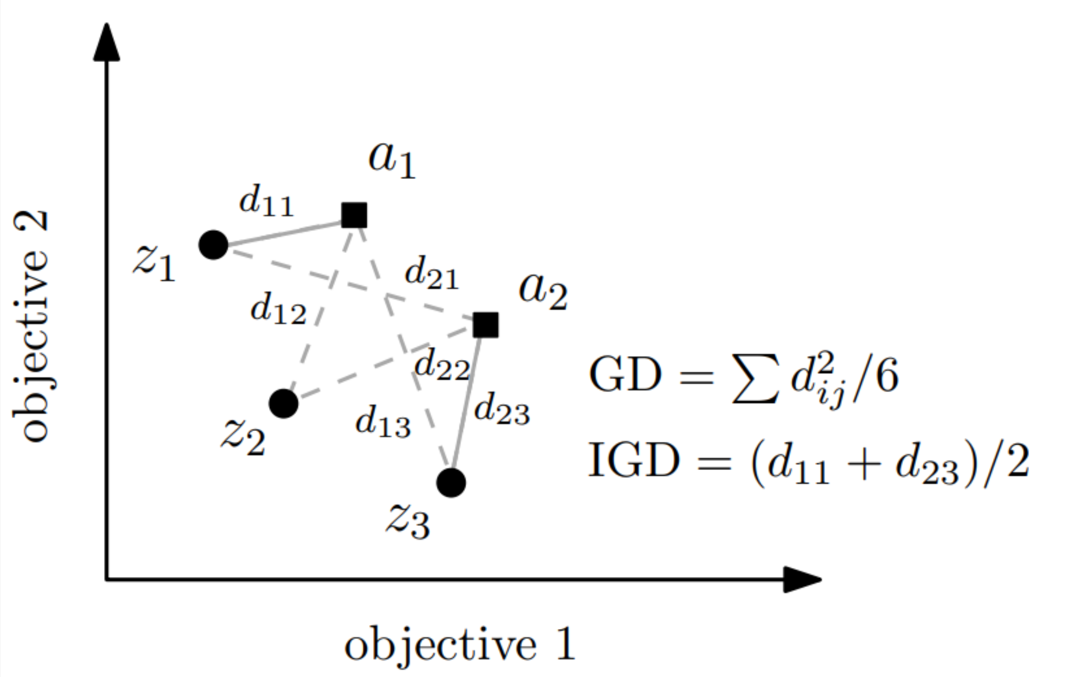
Multi-Objective Evolutionary Algorithms



Metric for Multi-Objective EAs

current Pareto front is A = {a1, a2}

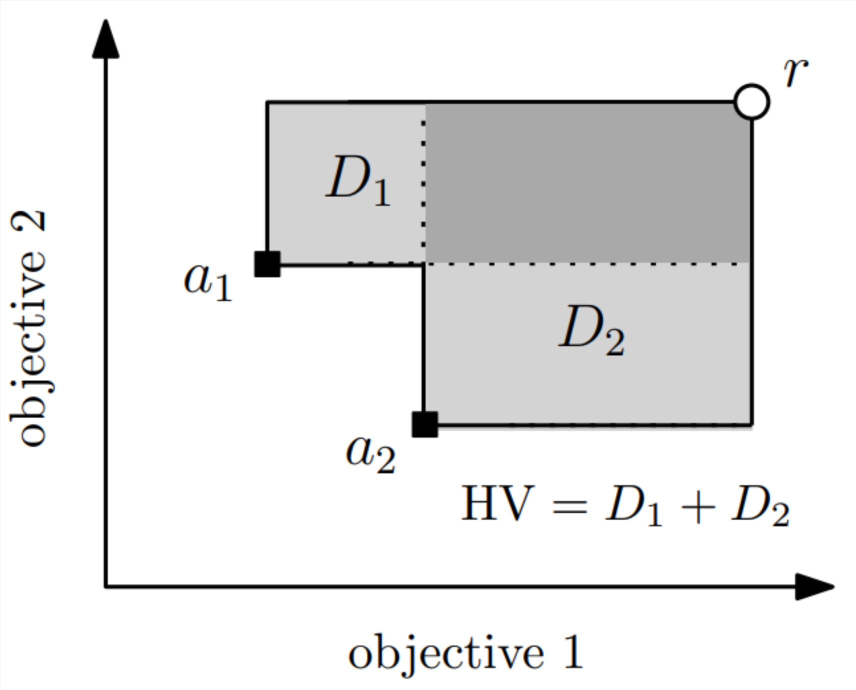
true Pareto front is Z = {z1, z2, z2}



Generational Distance (IGD) and
Inverse Generational Distance (IGD).

current Pareto front is A = {a1, a2}

a reference point r

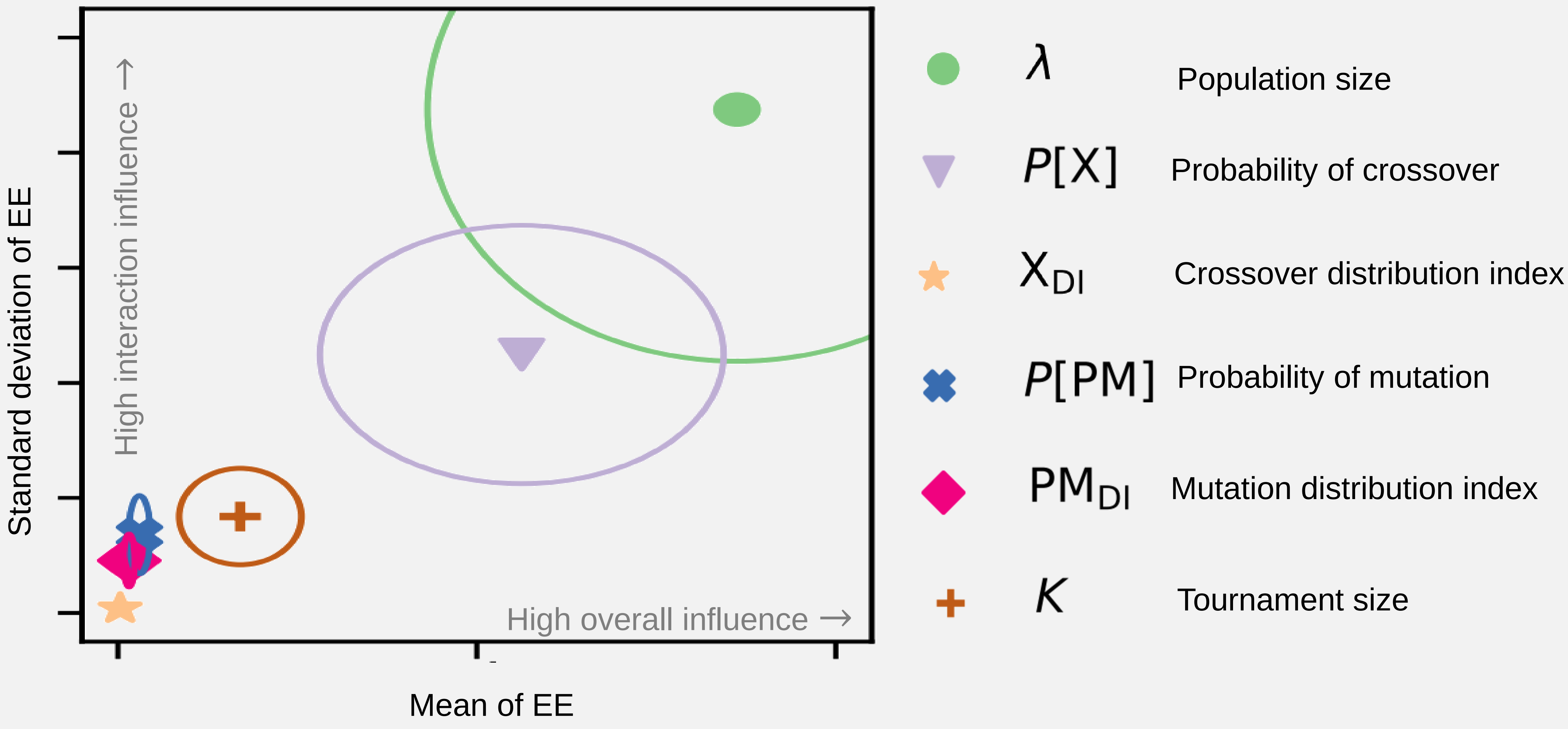


Hypervolume Indicator (HV)

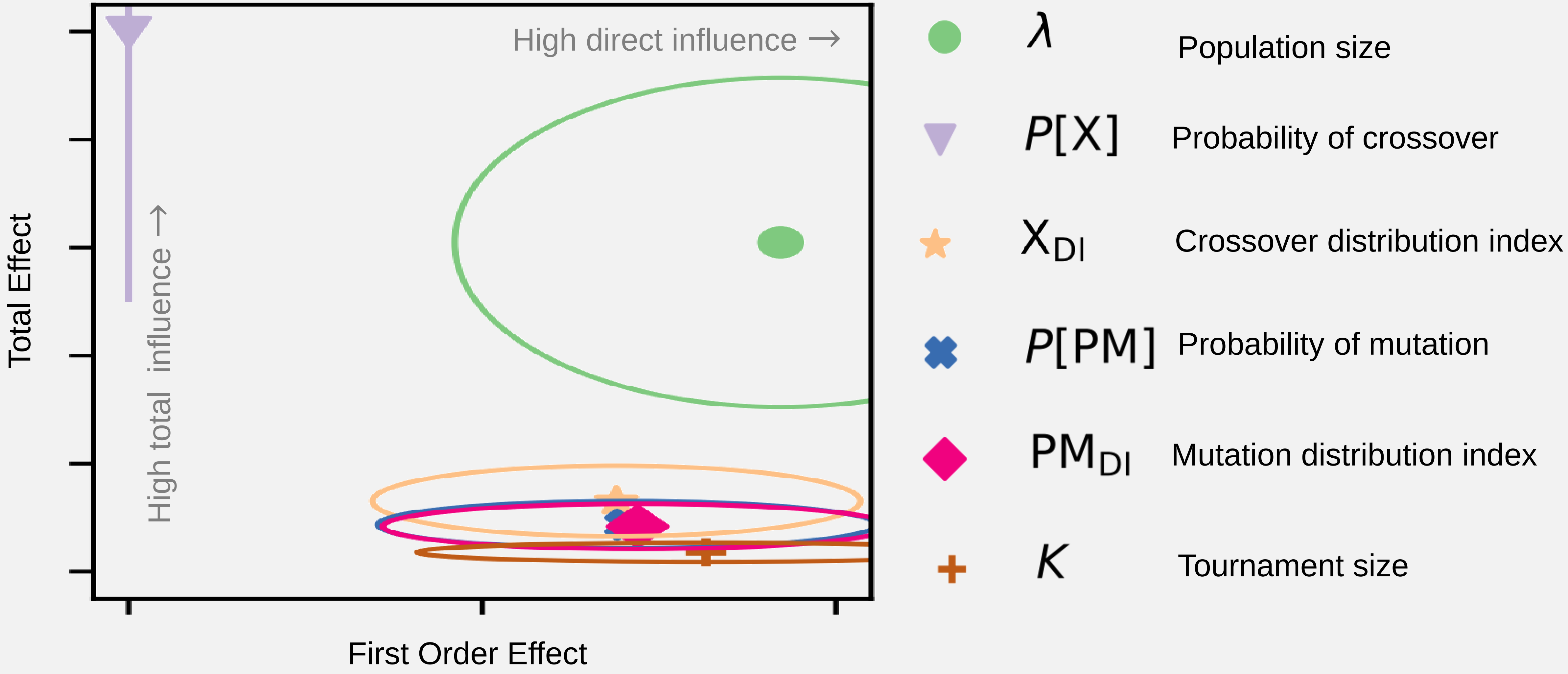
Multi-Objective EAs - Hyperparameters

Algo	Params	Domain	Description
Common	λ	$[10, 1000]$	Population size.
	$P[\mathrm{X}]$	$[0, 1]$	Simulated binary crossover (SBX) probability
	X_{DI}	$[1, 200]$	SBX distribution index
	$P[\mathrm{PM}]$	$[0, 1]$	Polynomial mutation (PM) probability
	PM_{DI}	$[1, 200]$	PM distribution index
NSGA-III	K	$[2, 10]$	Tournament size
	Selection	Tournament	Parents selection for offspring generation
MOEA/D	*Mode*	{“penalty based boundary intersection (PBI),” “Tchebycheff,” “Tchebycheff with normalization,” “modified Tchebycheff”}	Method for MOO decomposition into many SOO subproblems
	ϵ_N	$[0.05, 0.5]$	Neighbors: percentage of the population considered as neighbors for each sub-problem generation

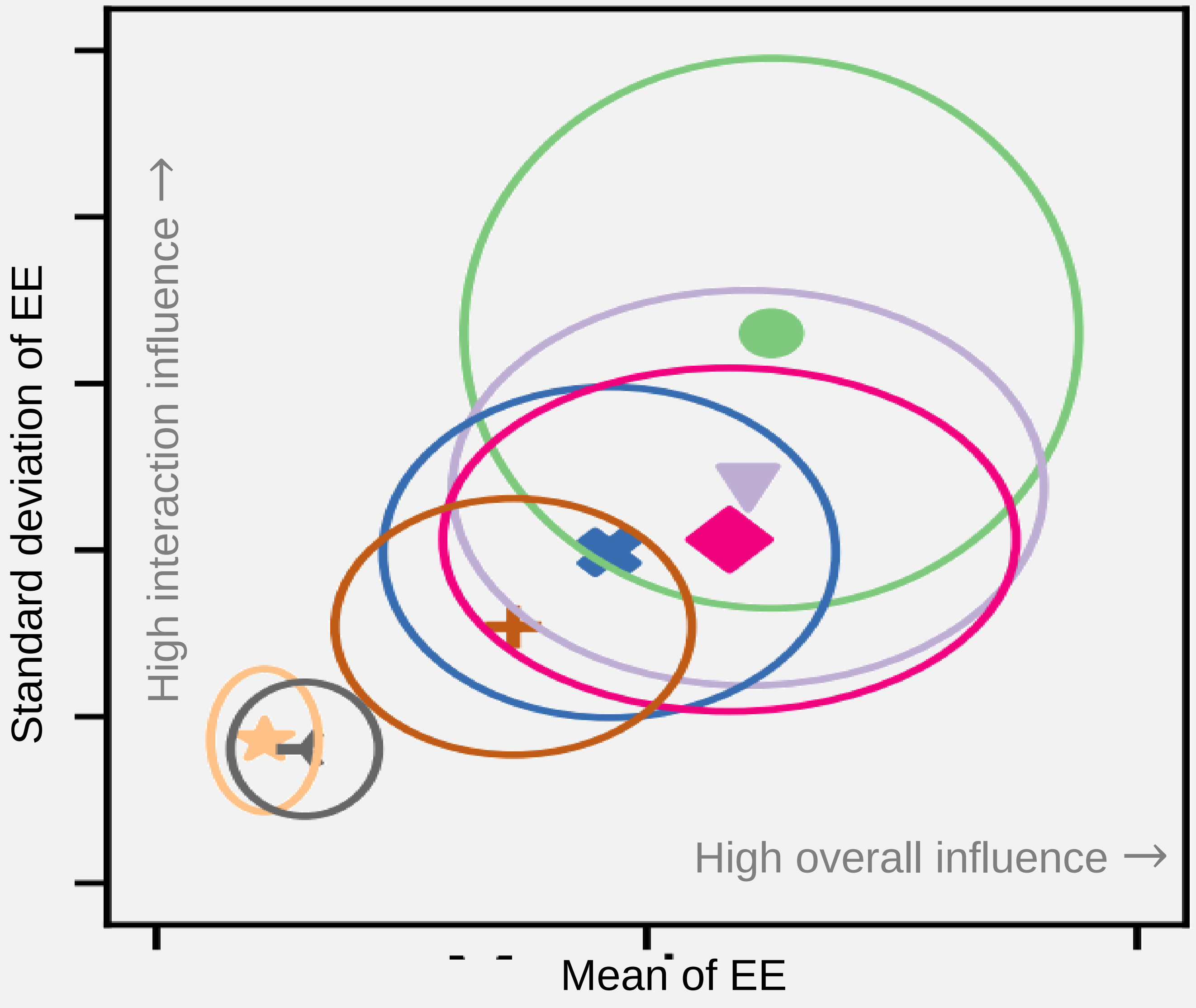
NSGA-III Sensitivity to its Hyperparameters



NSGA-III Sensitivity to its Hyperparameters

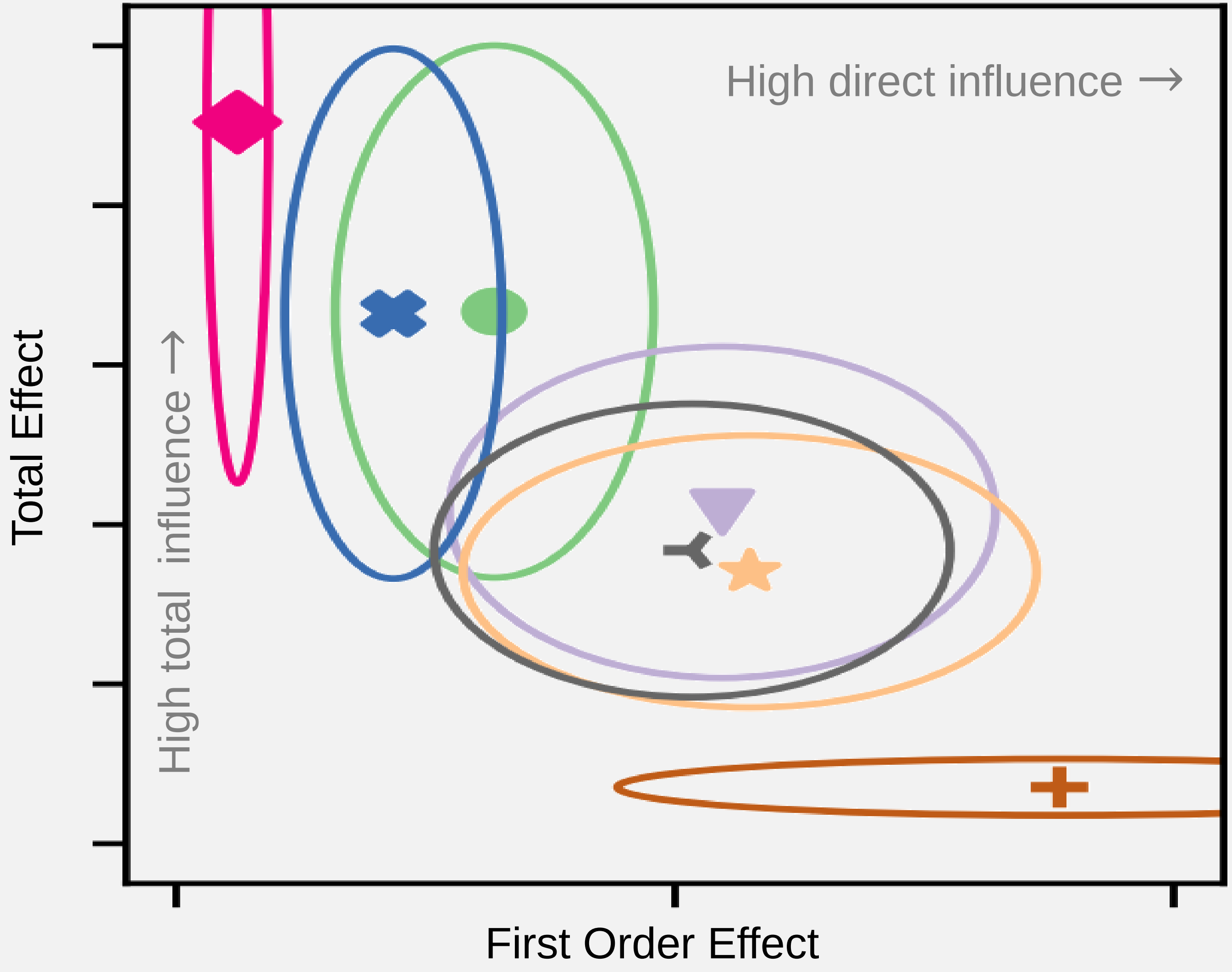


MOEA/D Sensitivity to its Hyperparameters



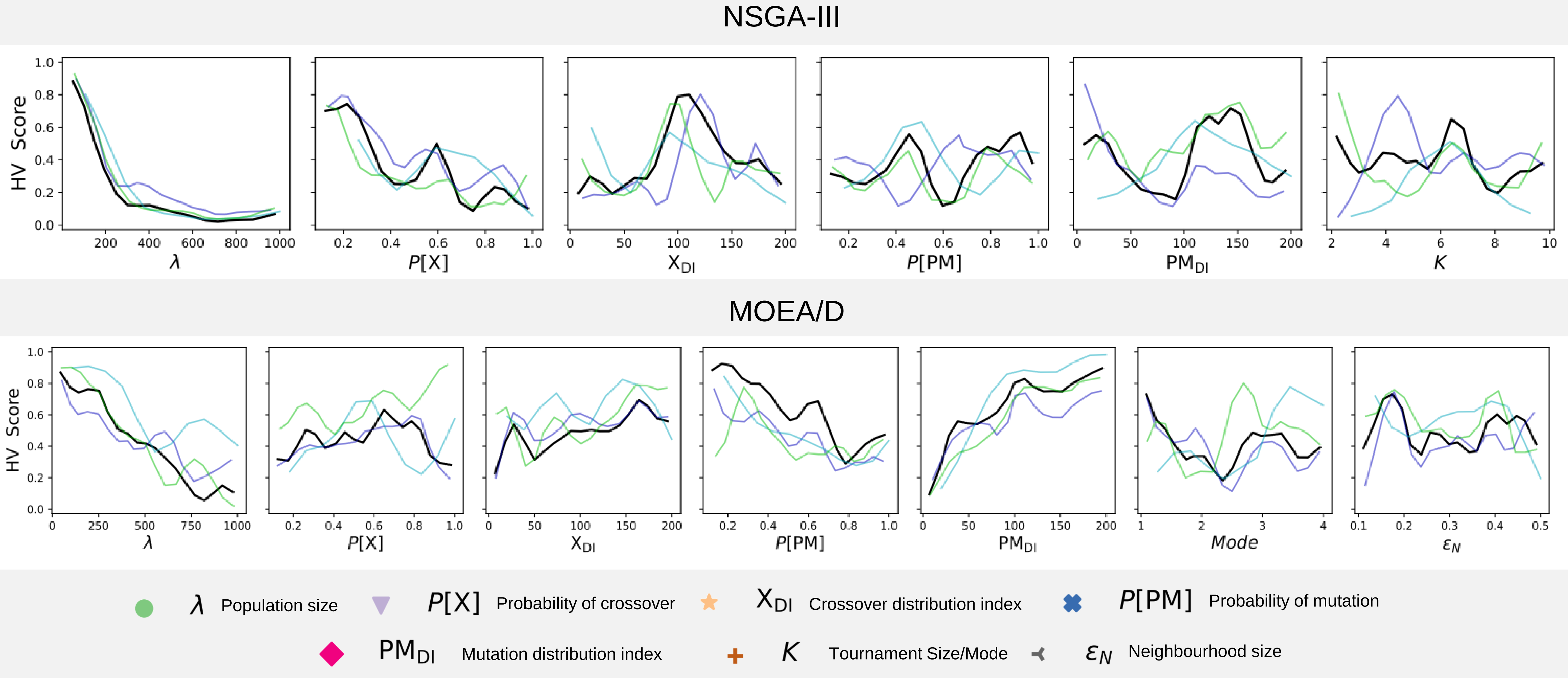
●	λ	Population size
▼	$P[X]$	Probability of crossover
★	X_{DI}	Crossover distribution index
✖	$P[PM]$	Probability of mutation
◆	PM_{DI}	Mutation distribution index
+	K	Mode
⤙	ε_N	Neighbourhood size

MOEA/D Sensitivity to its Hyperparameters



●	λ	Population size
▼	$P[X]$	Probability of crossover
★	X_{DI}	Crossover distribution index
✖	$P[PM]$	Probability of mutation
◆	PM_{DI}	Mutation distribution index
✚	K	Mode
⤙	ε_N	Neighbourhood size

Hyperparameter Influence Summary



Order of Turning: Single Objective EAs

- Non-dominated Sorting Genetic Algorithm –III (NSGA-III)
  - Population size
  - Crossover Probability
  - Crossover distribution index
  - Tournament size
  - Mutation Probability
  - Mutation distribution index

- Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D)
  - Population size
  - Mode of decomposition
  - Mutation distribution index
  - Mutation Probability
  - Crossover Probability
  - Neighbourhood size
  - Crossover distribution index

References

- V Ojha, J Timmis, G Nicosia (2022) Assessing ranking and effectiveness of evolutionary algorithm hyperparameters using global sensitivity analysis methodologies Swarm and Evolutionary Computation 74, 101130. URL: https://arxiv.org/abs/2207.04820

  Code: https://github.com/vojha-code/saofeas

- R Taylor, V Ojha, I Martino, G Nicosia (2021) Sensitivity analysis for deep learning: ranking hyper-parameter influence. IEEE 33rd 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI) URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9643336

Dr Varun Ojha

v.k.ojha@reading.ac.uk; vkojha@ieee.org

Github: https://github.com/vojha-code