Efficient Performance Trade-off Modeling for Analog Circuit based on Bayesian Neural Network

Zhengqi Gao[1], Jun Tao[1], Fan Yang[1], Yangfeng Su[2], Dian Zhou[1,3], Xuan Zeng[1]
[1] ASIC& System State Key Lab, School of Microelectronics, Fudan University, Shanghai, China
[2] School of Mathematical Sciences, Fudan University, Shanghai, China
[3] University of Texas at Dallas, Dallas, USA

Hierarchical Optimization for Analog System

Level 1 (System Level):
e.g., wireless communication system

(constraint mapping)

Level 2

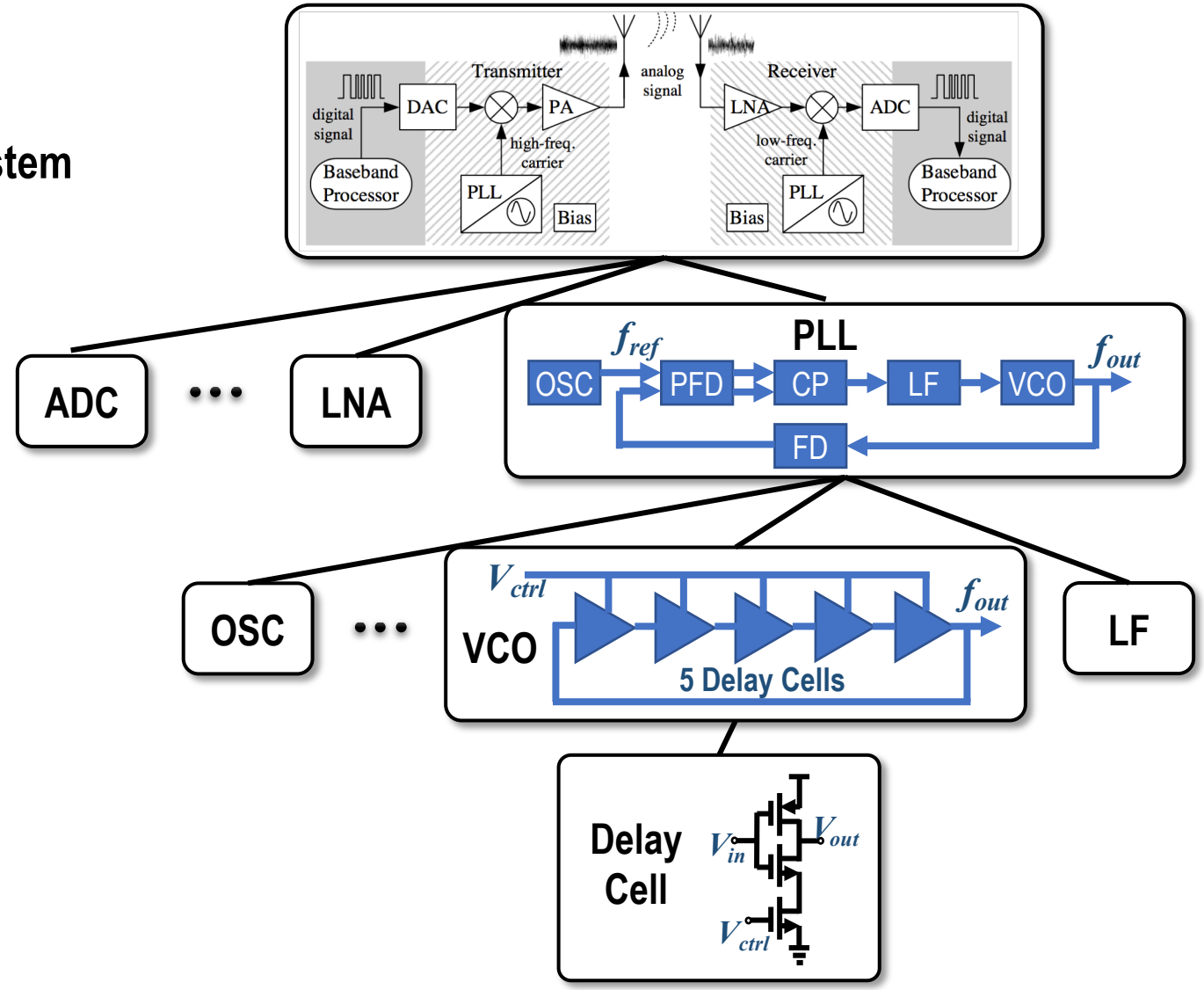
(constraint mapping)

Level 3

(constraint mapping)

Level 4
(Transistor Level)

(transistor sizing)



Hierarchical Optimization for Analog System

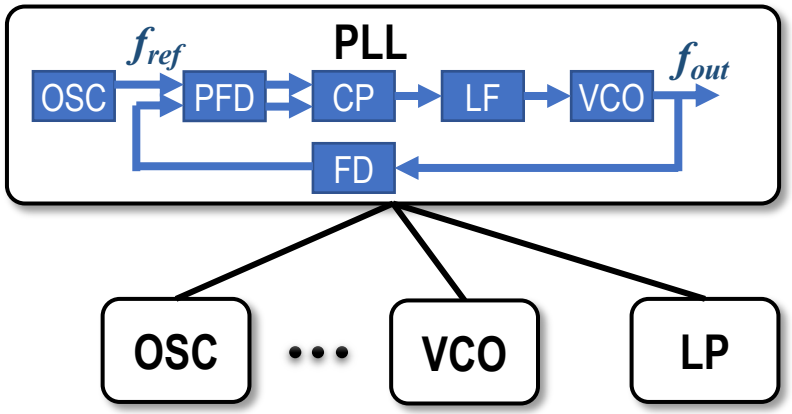
Example: PLL Design

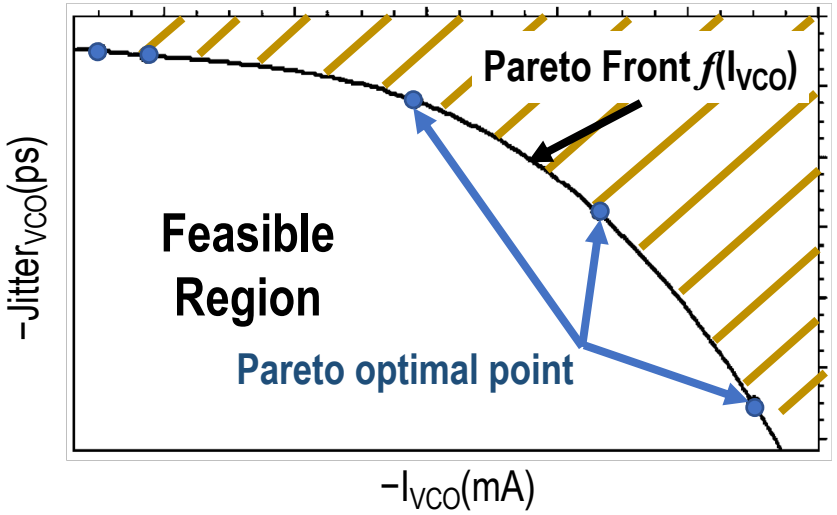
$$\min \ \text{Power}_{PLL}(\mathbf{x})$$
$$\text{s.t.} \ \ \text{PhaseNoise}_{PLL}(\mathbf{x}) \le \text{Spec}_{PN}$$

… …

Design Parameters $\mathbf{x}$: I_{LF}, I_{OSC}, I_{VCO}, Jitter_{VCO}, …

Behavioral Models : $\text{Power}_{PLL}(\mathbf{x})$, $\text{PhaseNoise}_{PLL}(\mathbf{x})$

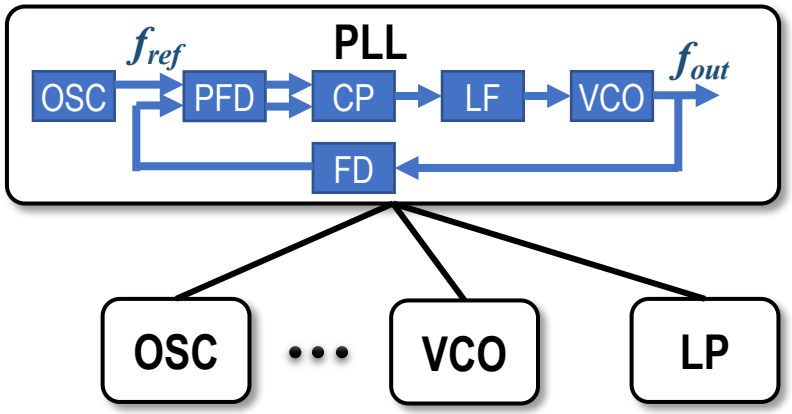




Hierarchical Optimization for Analog System

Example: PLL Design

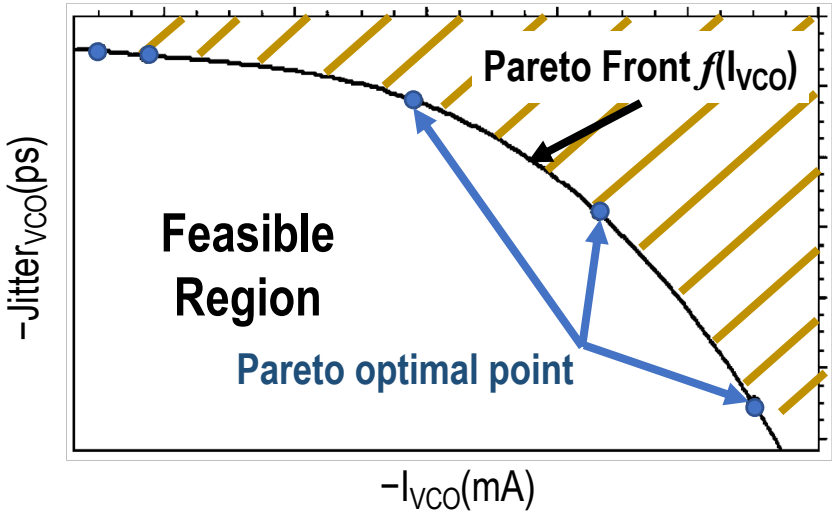
$$\min \ \text{Power}_{PLL}(\mathbf{x})$$
$$\text{s.t.} \ \ \text{PhaseNoise}_{PLL}(\mathbf{x}) \le \text{Spec}_{PN}$$
$$-\text{Jitter}_{VCO} \le f(\text{I}_{VCO})$$
$$\cdots \ \cdots$$



Design Parameters $\mathbf{x}$: I_{LP}, I_{OSC}, I_{VCO}, Jitter_{VCO}, ⋯

Behavioral Models : $\text{Power}_{PLL}(\mathbf{x})$, $\text{PhaseNoise}_{PLL}(\mathbf{x})$

Pareto Front (PF) : $f(\text{I}_{VCO})$



Pareto Front (PF) Modeling

- **Goal: Obtain trade-off models among different PoIs**
- **Many methods exist in the literature --- solve multi-objective optimization**
  - Simulation-based
    - Normal Boundary Intersection
    - Successive Pareto Optimization
    - WATSON, MOEA/D, NSGA-II
  - GP model-based
    - GPMOOG, MOBO
    - Square exponential covariance function:

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \theta_1 \cdot \exp\left(-\theta_2 \cdot \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$$

$\mathbf{x}_i$, $\mathbf{x}_j$: design variables

θ_1, θ_2: Hyper-parameters

- **Obtain PF model based on Bayesian Neural Network with few sims**

Pareto Front Modeling

- **Problem definition**
  - $\mathbf{f} = \mathbf{f}(\mathbf{x})$ dominant $\mathbf{f}^* = \mathbf{f}(\mathbf{x}^*)$

$$\mathbf{f} \neq \mathbf{f}^* \quad \text{and} \quad f_i \leq f_i^* \quad i = 1, 2, \cdots, m$$

$\mathbf{x} \in \Omega \subseteq \Re^d$: design variables

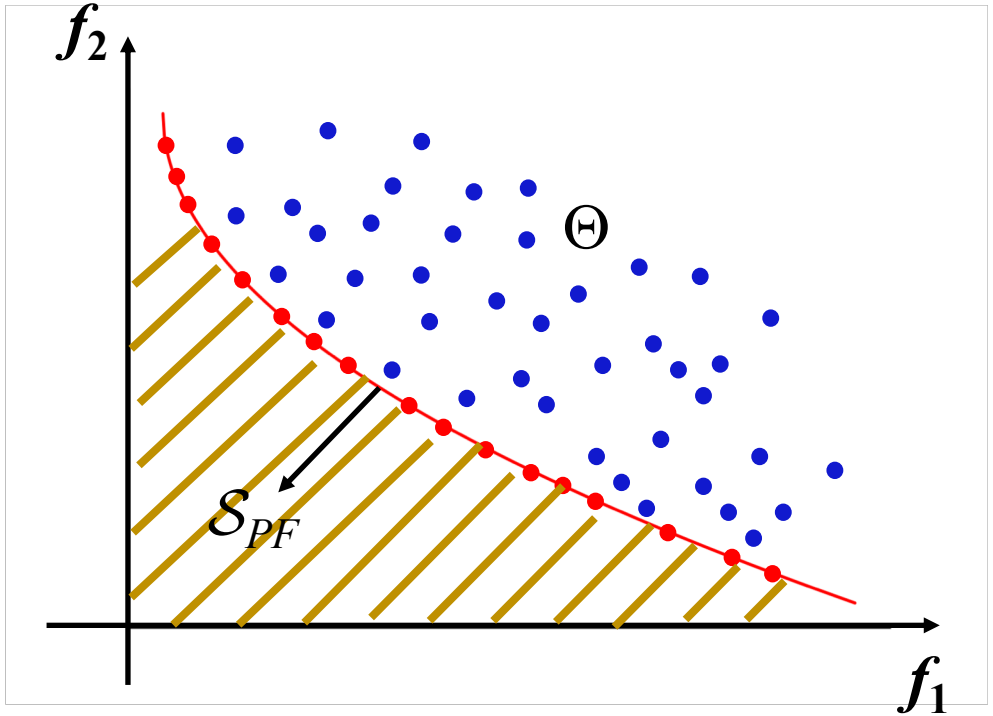
$\mathbf{f}(\mathbf{x}) \in \Theta \subseteq \Re^m$: Pols of analog circuit

  - No $\mathbf{f}$ in Θ dominant $\mathbf{f}^*$, $\mathbf{f}^*$ is Pareto optimal point, $\mathbf{x}^*$ is Pareto optimal design
  - $\mathcal{S}_{PS}$ and $\mathcal{S}_{PF}$ can be obtained by

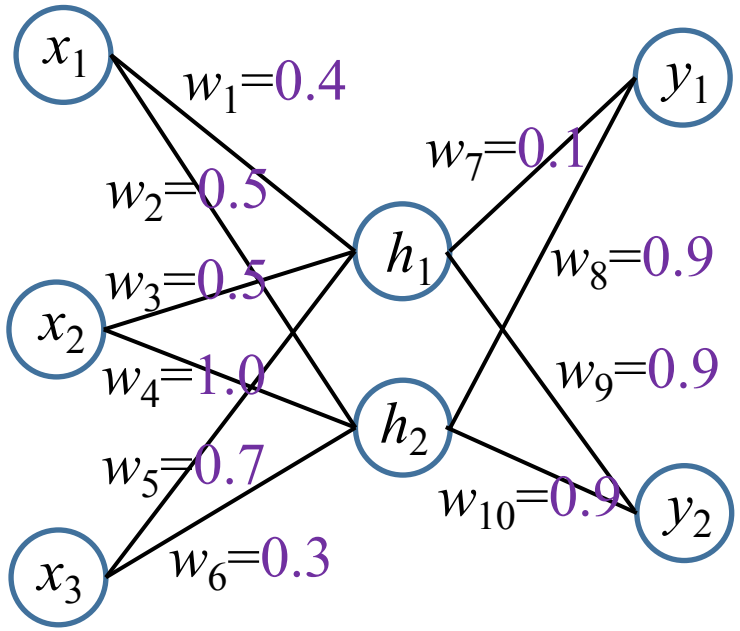
$$\min_{\mathbf{x} \in \Omega} \quad \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_m(\mathbf{x}) \right]^T$$

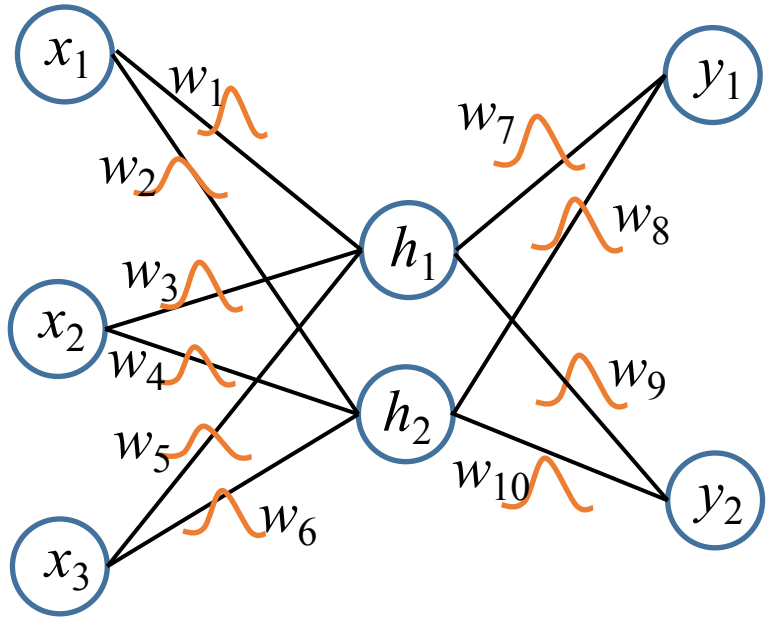
$\mathcal{S}_{PS}$: all Pareto optimal design

$\mathcal{S}_{PF}$: all Pareto optimal point



Bayesian Neural Network





- **Conventional neural network**
  - Parameters are deterministic
  - Deterministic prediction

- **Bayesian neural network (BNN)**
  - Parameters satisfy PDFs
  - Probabilistic prediction

Bayesian Neural Network

- **Prior** $p(\mathbf{w}|\alpha) = N(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$
- **Likelihood**

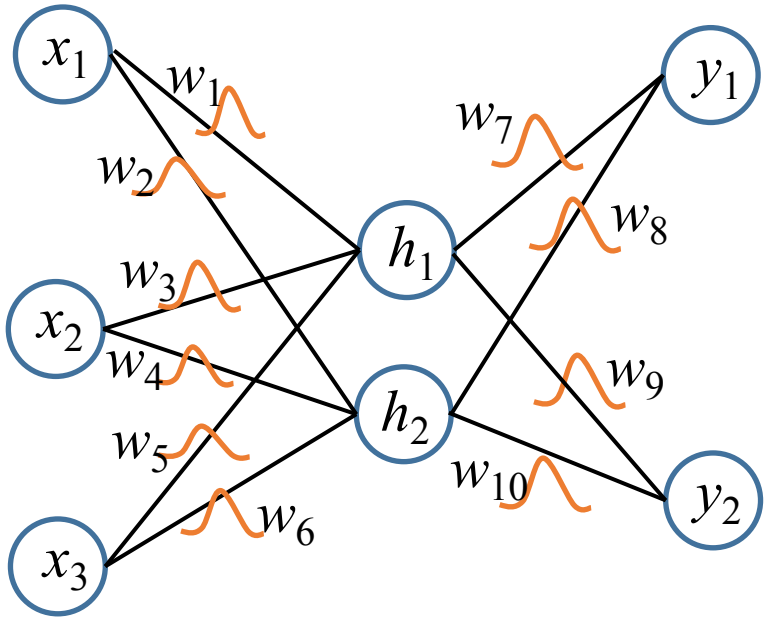
$$\mathbf{f}(\mathbf{x}) = \mathbf{y}(\mathbf{x}, \mathbf{w}) + \boldsymbol{\varepsilon}, \text{ where } \boldsymbol{\varepsilon} \sim N(\boldsymbol{\varepsilon}|\mathbf{0}, \beta^{-1}\mathbf{I})$$

$$\Rightarrow \boxed{p(\mathbf{f}|\mathbf{x}, \mathbf{w}, \beta) = N(\mathbf{f}|\mathbf{y}(\mathbf{x}, \mathbf{w}), \beta^{-1}\mathbf{I})}$$

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{f}_i)|i = 1, 2, \cdots, N\}$$

$$p(\mathcal{D}|\mathbf{w}, \beta) = \prod_{i=1}^{N} \boxed{p(\mathbf{f}_i|\mathbf{x}_i, \mathbf{w}, \beta)}$$

$$= \prod_{i=1}^{N} N(\mathbf{f}_i|\mathbf{y}(\mathbf{x}_i, \mathbf{w}), \beta^{-1}\mathbf{I})$$



$\mathbf{w} = [w_1\ w_2 \cdots w_W]^T$: BNN weights

$\mathbf{y}(\mathbf{x}, \mathbf{w}) \subseteq \Re^m$: BNN PoIs prediction

$\boldsymbol{\varepsilon} \subseteq \Re^m$: approximation error

$\mathcal{D}$: training data set

α, β: precision(inverse of variance)

Bayesian Neural Network

- **Prior** $p(\mathbf{w}|\alpha) = N(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$
- **Likelihood**

$$p(\mathcal{D}|\mathbf{w}, \beta) = \prod_{i=1}^{N} N(\mathbf{f}_i|\mathbf{y}(\mathbf{x}_i, \mathbf{w}), \beta^{-1}\mathbf{I})$$

- **Posterior**

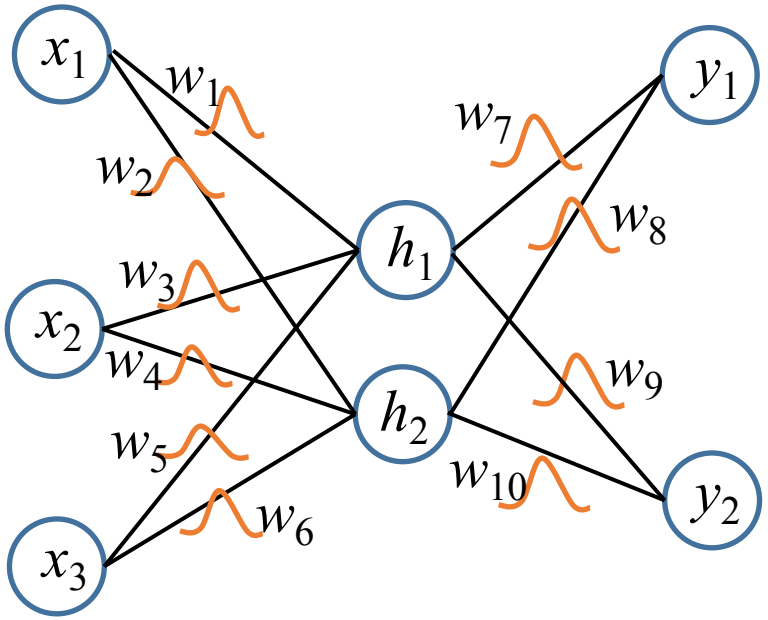
Bayes Theorem: Posterior $\propto$ Prior $\times$ Likelihood, i.e.,

$$p(\mathbf{w}|\mathcal{D}, \alpha, \beta) \propto p(\mathbf{w}|\alpha) \cdot p(\mathcal{D}|\mathbf{w}, \beta)$$

- **Predictive function**

$$p(\mathbf{f}|\mathbf{x}, \mathcal{D}, \alpha, \beta) = \int p(\mathbf{f}|\mathbf{x}, \mathbf{w}, \beta) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta) d\mathbf{w}$$

$$= \int N(\mathbf{f}|\mathbf{y}(\mathbf{x}, \mathbf{w}), \beta^{-1}\mathbf{I}) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta) d\mathbf{w}$$



Bayesian Neural Network

- **Prior** $p(\mathbf{w}|\alpha) = N(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$
- **Likelihood**

$$p(\mathcal{D}|\mathbf{w}, \beta) = \prod_{i=1}^{N} N(\mathbf{f}_i | \mathbf{y}(\mathbf{x}_i, \mathbf{w}), \beta^{-1}\mathbf{I})$$

- **Posterior**

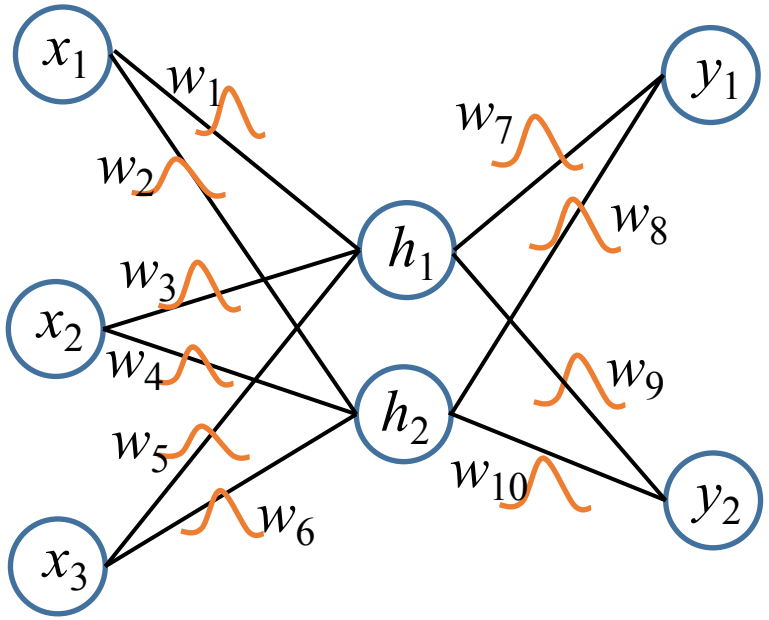
Bayes Theorem: Posterior $\propto$ Prior $\times$ Likelihood, i.e.,

$$p(\mathbf{w}|\mathcal{D}, \alpha, \beta) \propto p(\mathbf{w}|\alpha) \cdot p(\mathcal{D}|\mathbf{w}, \beta)$$

- **Predictive function**

$$p(\mathbf{f}|\mathbf{x}, \mathcal{D}, \alpha, \beta) = \int p(\mathbf{f}|\mathbf{x}, \mathbf{w}, \beta) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta) d\mathbf{w}$$

$$= \int N(\mathbf{f}|\mathbf{y}(\mathbf{x}, \mathbf{w}), \beta^{-1}\mathbf{I}) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta) d\mathbf{w}$$



Bayesian Neural Network

- **Prior** $p(\mathbf{w}|\alpha) = N(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$
- **Likelihood**

$$p(\mathcal{D}|\mathbf{w}, \beta) = \prod_{i=1}^{N} N(\mathbf{f}_i | \mathbf{y}(\mathbf{x}_i, \mathbf{w}), \beta^{-1}\mathbf{I})$$

- **Posterior**

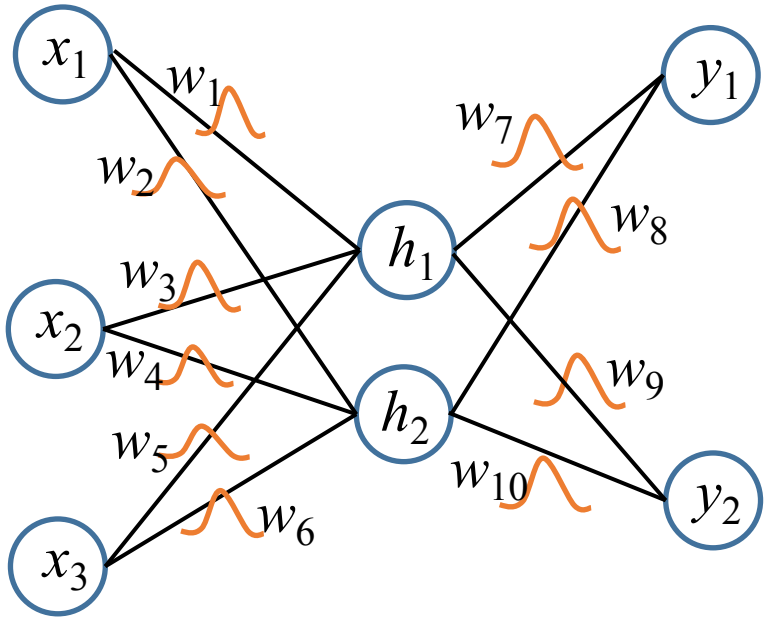
Bayes Theorem: Posterior $\propto$ Prior $\times$ Likelihood, i.e.,

$$p(\mathbf{w}|\mathcal{D}, \alpha, \beta) \propto p(\mathbf{w}|\alpha)\, p(\mathcal{D}|\mathbf{w}, \beta)$$

- **Predictive function**

$$p(\mathbf{f}|\mathbf{x}, \mathcal{D}, \alpha, \beta) = \int p(\mathbf{f}|\mathbf{x}, \mathbf{w}, \beta) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta)\, d\mathbf{w}$$

$$= \int N(\mathbf{f}|\mathbf{y}(\mathbf{x}, \mathbf{w}), \beta^{-1}\mathbf{I}) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta)\, d\mathbf{w}$$



Bayesian Neural Network

- **Approximate Posterior**

$$\max_{\boldsymbol{\varphi}} \quad L(\boldsymbol{\varphi}) = \mathrm{E}_{q(\mathbf{w}|\boldsymbol{\varphi})}\left[\log p(\mathbf{w}, \mathcal{D}, \alpha, \beta) - \log q(\mathbf{w}|\boldsymbol{\varphi})\right]$$

$$s.t. \quad \mathrm{supp}\left[q(\mathbf{w}|\boldsymbol{\varphi})\right] \subseteq \mathrm{supp}\left[p(\mathbf{w}|\mathcal{D}, \alpha, \beta)\right]$$

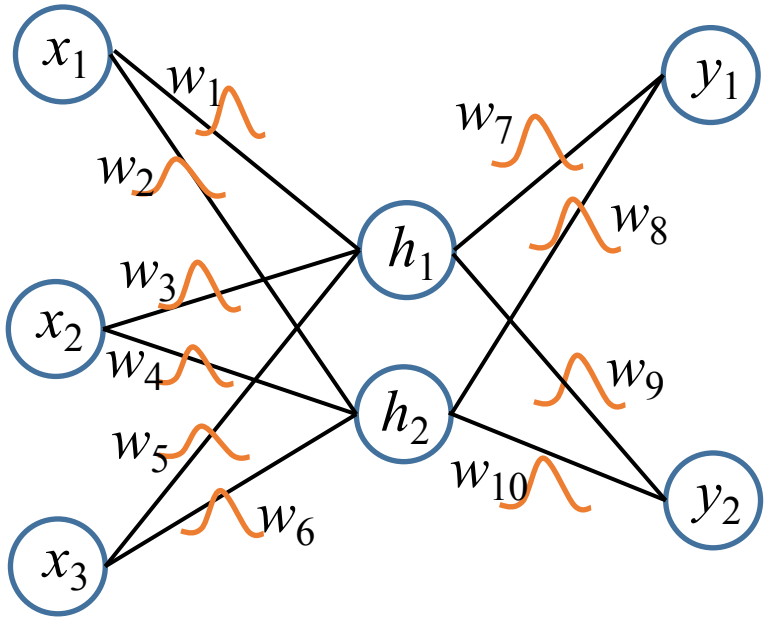
  - Automatic differential variational inference (ADVI)

$$\Rightarrow \quad q(\mathbf{w}|\boldsymbol{\varphi}^*)$$

- **Predictive function**

$$p(\mathbf{f}|\mathbf{x}, \mathcal{D}, \alpha, \beta) = \int p(\mathbf{f}|\mathbf{x}, \mathbf{w}, \beta) \cdot p(\mathbf{w}|\mathcal{D}, \alpha, \beta) d\mathbf{w}$$

$$\approx \int N(\mathbf{f}|\mathbf{y}(\mathbf{x}, \mathbf{w}), \beta^{-1}\mathbf{I}) \cdot q(\mathbf{w}|\boldsymbol{\varphi}^*) d\mathbf{w}$$

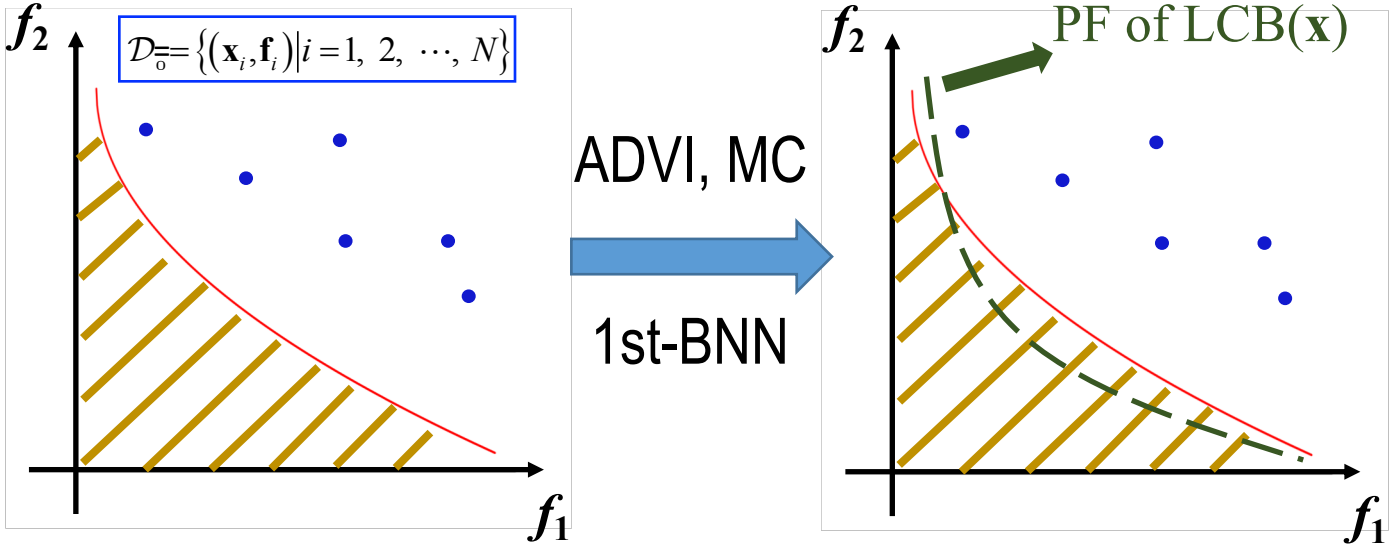


$q(\mathbf{w}|\boldsymbol{\varphi})$: PDF parametrized by $\boldsymbol{\varphi}$

$\mathrm{supp}[q(\mathbf{w}|\boldsymbol{\varphi})]$: domain of $q(\mathbf{w}|\boldsymbol{\varphi})$

$\mathrm{supp}[p(\mathbf{w}|\mathcal{D}, \alpha, \beta)]$: domain of $p(\mathbf{w}|\mathcal{D}, \alpha, \beta)$

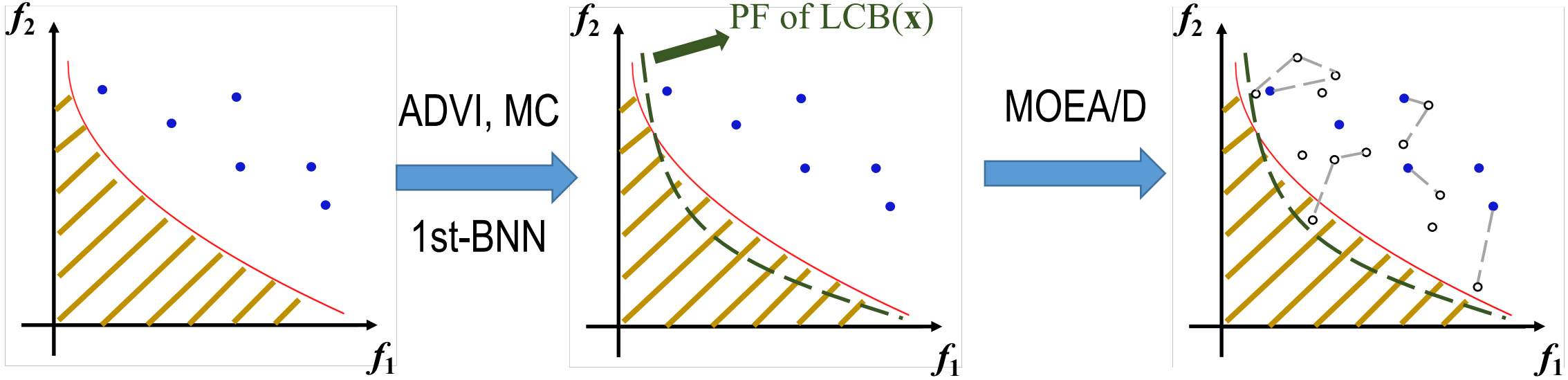
Pareto Front Modeling based on BNN



$$\mathrm{LCB}(\mathbf{x}) = \mathrm{mean}\left[\mathbf{f}(\mathbf{x})\right] - K \cdot \mathrm{var}\left[\mathbf{f}(\mathbf{x})\right]$$

K: balance sampling process

Pareto Front Modeling based on BNN

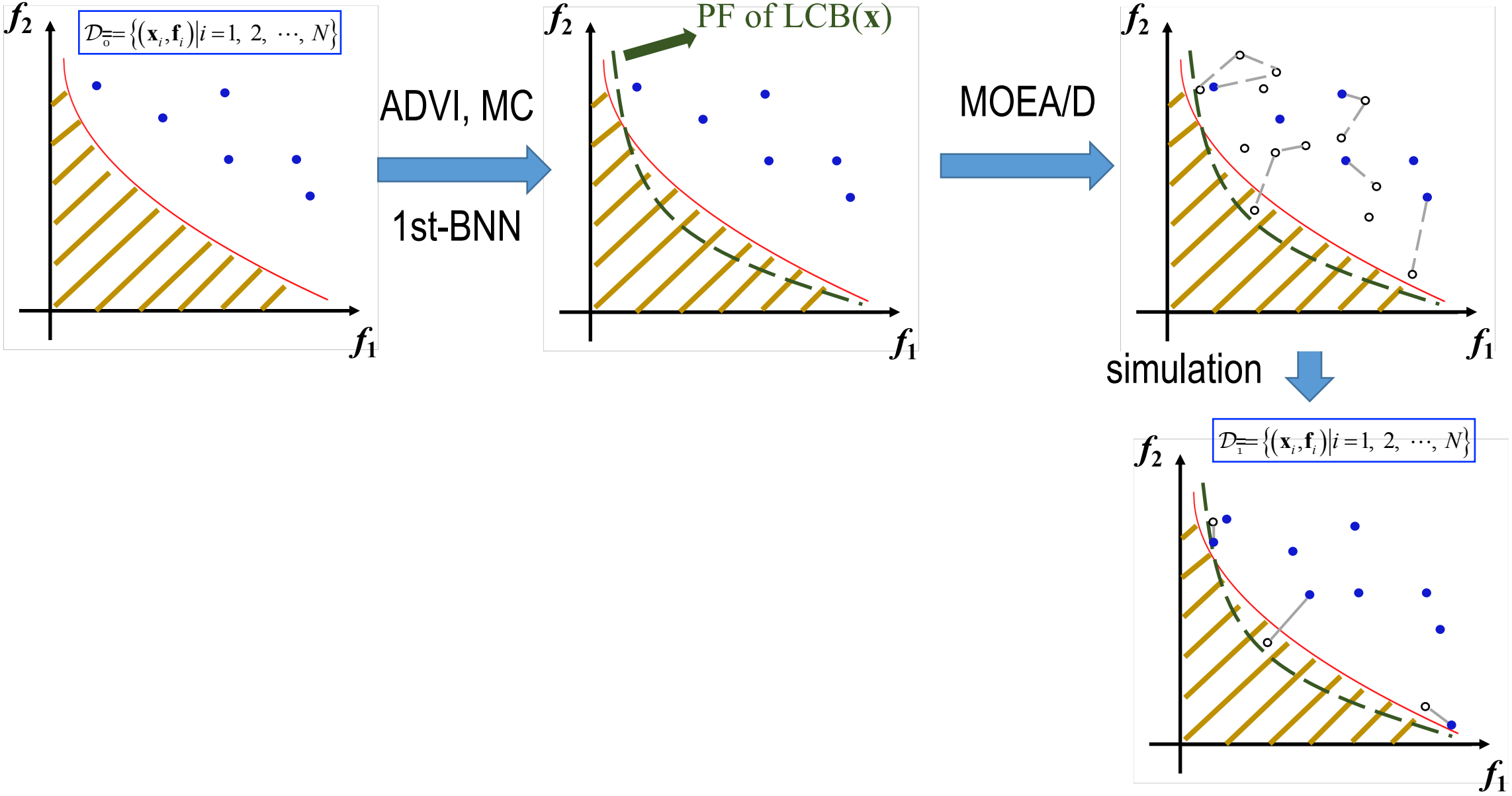


$$\mathrm{LCB}(\mathbf{x}) = \mathrm{mean}\left[\mathbf{f}(\mathbf{x})\right] - K \cdot \mathrm{var}\left[\mathbf{f}(\mathbf{x})\right]$$

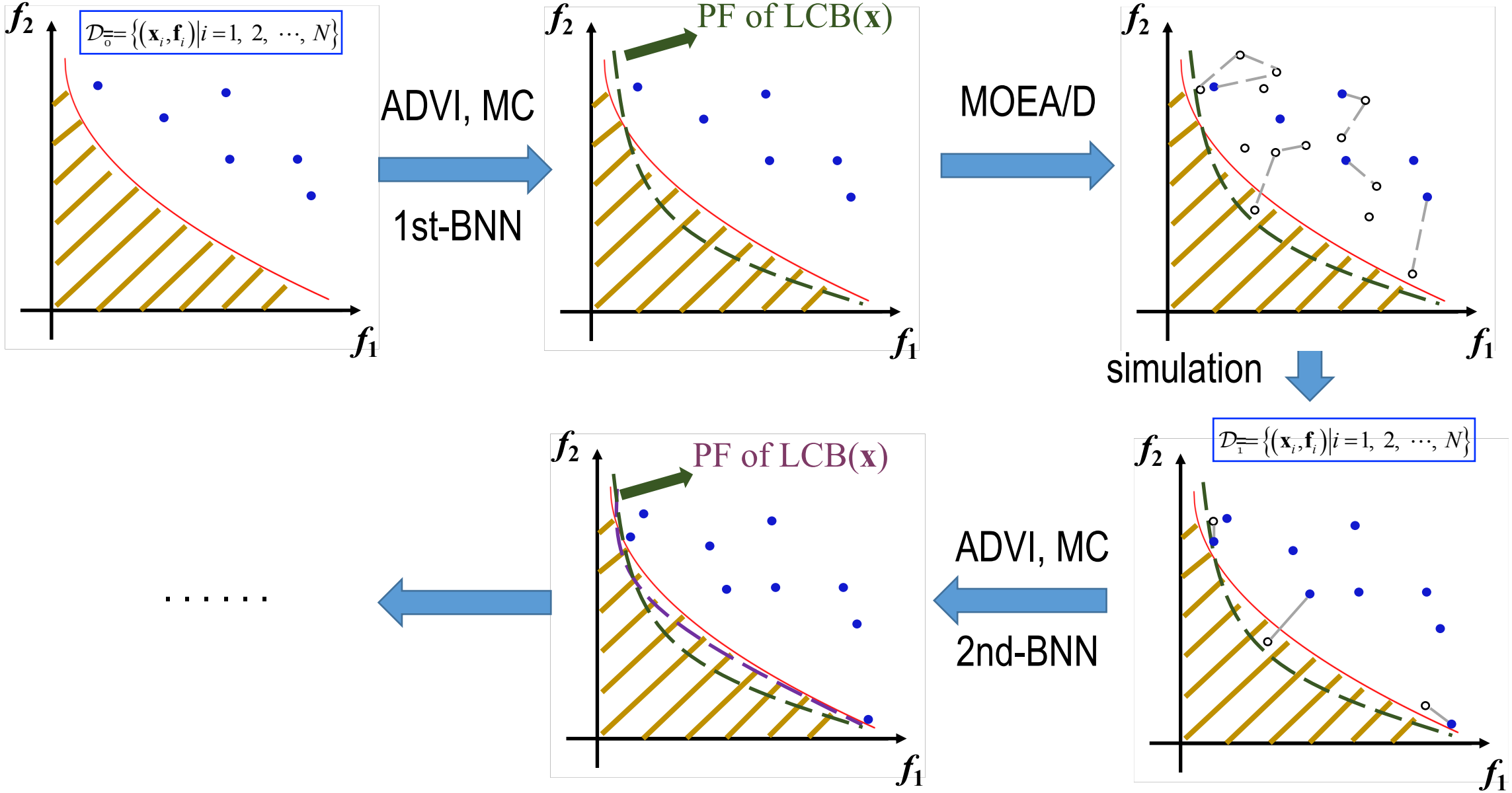
K: balance sampling process

ρ: sampling ratio, #black/(#black + #blue)

Pareto Front Modeling based on BNN



Pareto Front Modeling based on BNN

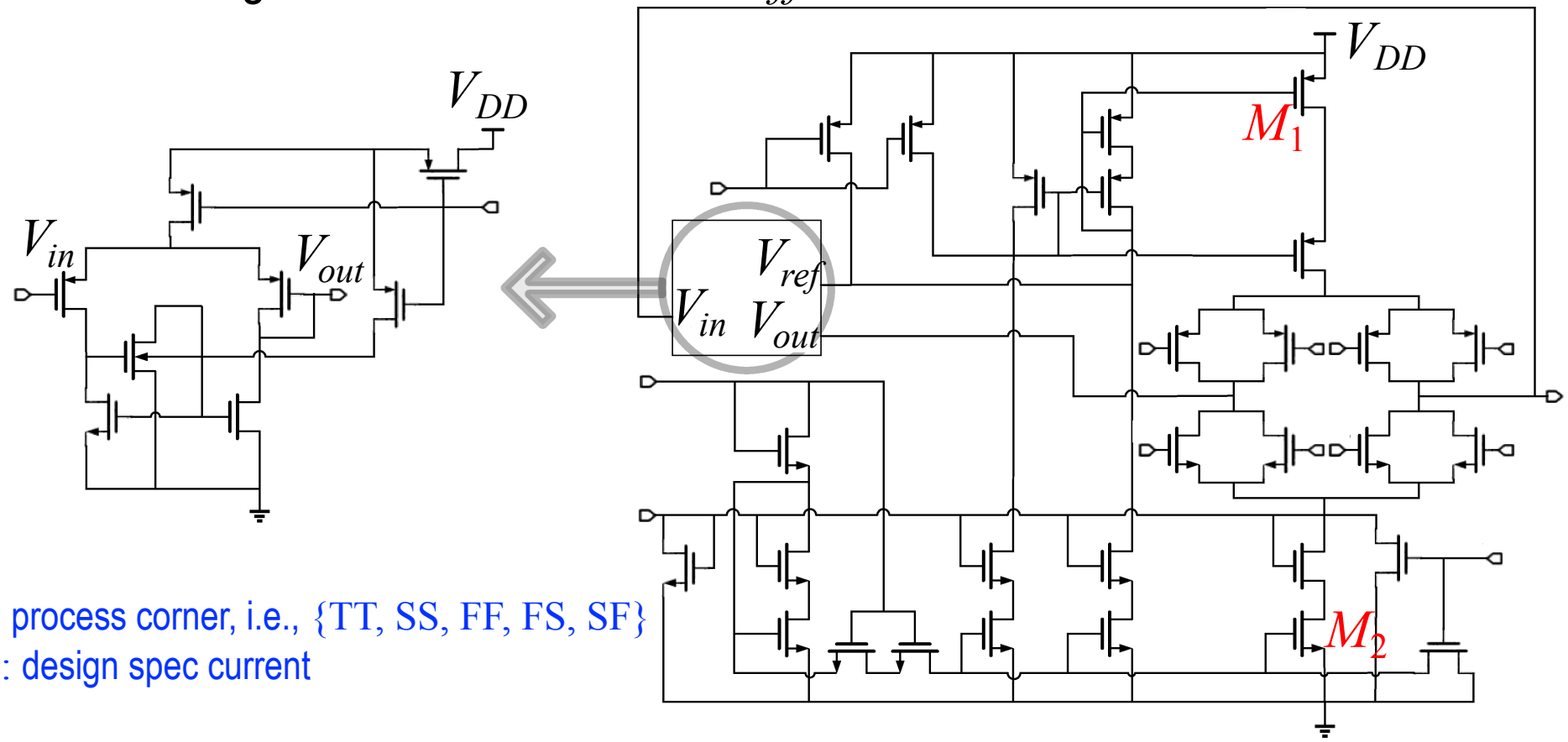


Example1: Charge Pump

- 40nm CMOS Technology
- Performance metrics: $(devi, diff)$
- 16 design variables
- Good design: small *devi* and small *diff*

$$diff = \max_{\Lambda}\left(I_{M_1,max} - I_{M_1,avg}\right) + \max_{\Lambda}\left(I_{M_1,avg} - I_{M_1,min}\right) + \max_{\Lambda}\left(I_{M_2,max} - I_{M_2,avg}\right) + \max_{\Lambda}\left(I_{M_2,avg} - I_{M_2,min}\right)$$

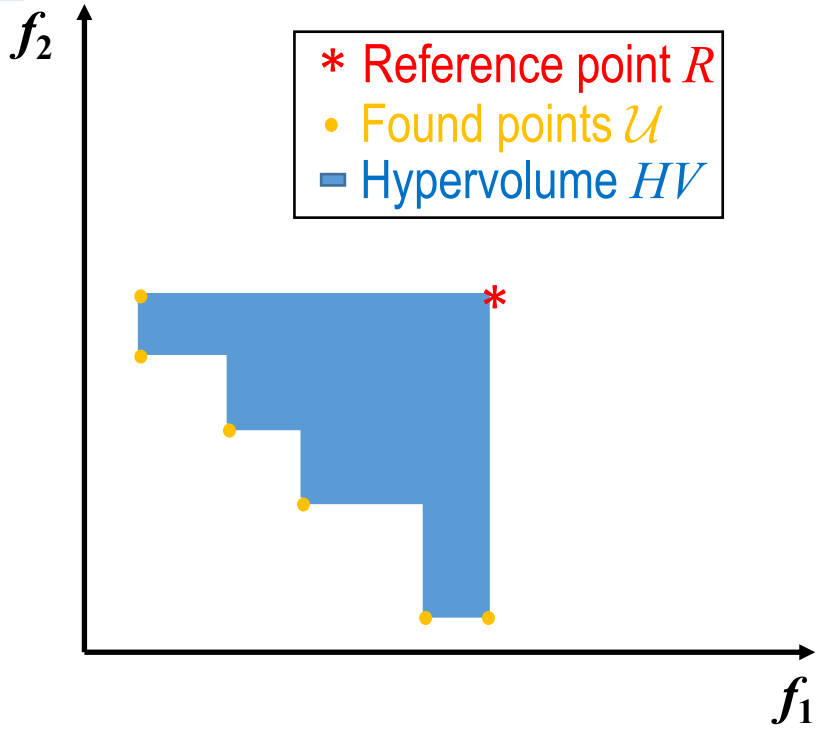
$$devi = \max_{\Lambda}\left|I_{M_1,avg} - I_{sta}\right| + \max_{\Lambda}\left|I_{M_2,avg} - I_{sta}\right|$$



Λ: process corner, i.e., {TT, SS, FF, FS, SF}

I_{sta}: design spec current

Example1: Charge Pump



- HV measures the quality of PF
- HV maximizes when $\mathcal{U}$ is exact PF

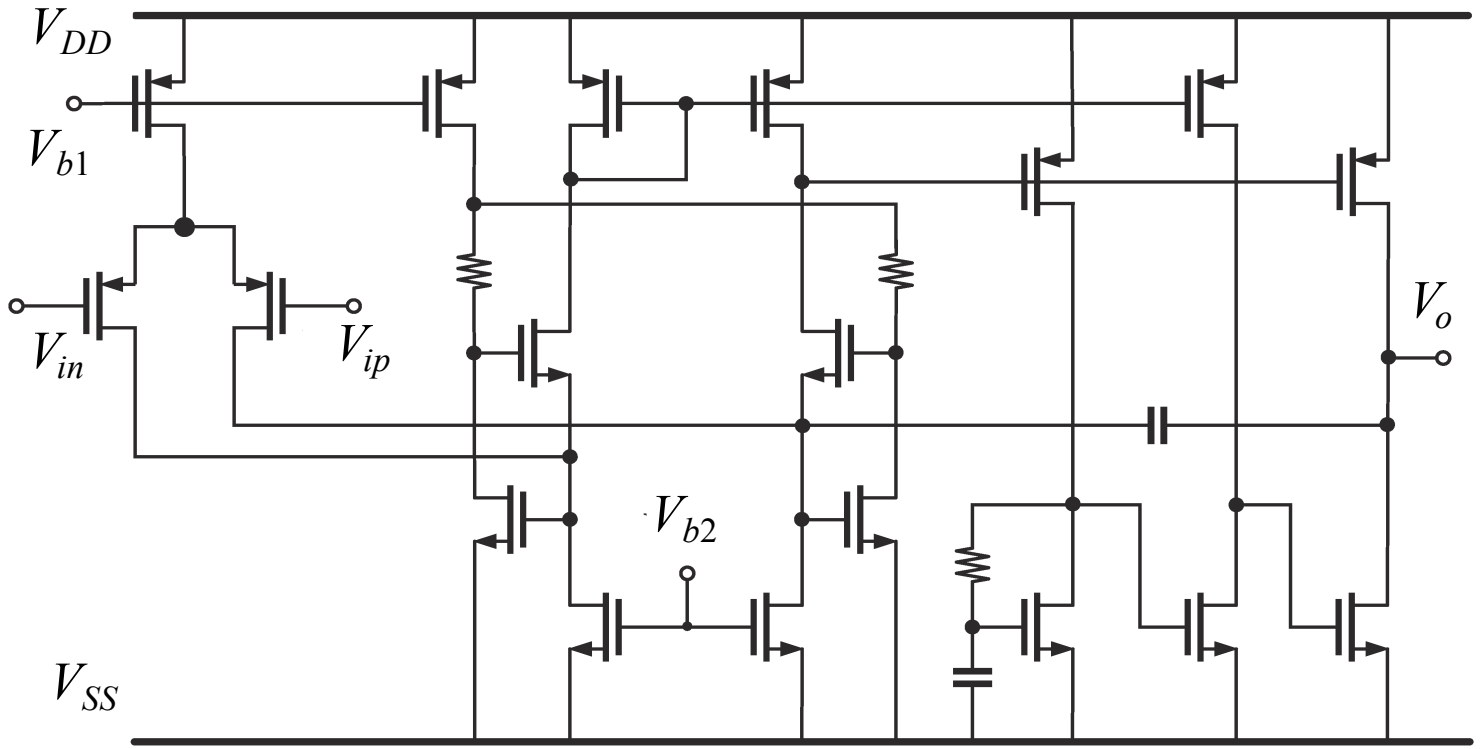
- Limit the number of allowed transistor-level simulations
- BNN: two hidden layers with 25 and 10 hidden units.
- BNNBO achieves small "Min *devi*", "Min *diff*" and large HV

Method	HV	Min *devi*	Min *diff*
BNNBO-150	**15046.55**	**0.19**	**137.32**
MOEA/D-150	14076.10	0.33	179.58
MOBO-150	14263.70	0.27	174.84
MOEA/D-300	14248.47	0.33	174.56
MOBO-300	14285.07	0.26	174.35

Example2: Low-Power Amplifier

- 40nm CMOS Technology
- Performance metrics $Gain_w = \min_{\Lambda} Gain,\ UGF_w = \min_{\Lambda} UGF,\ PM_w = \min_{\Lambda} PM$
- 13 design variables

Λ: process corner, i.e., {TT, SS, FF, FS, SF}



Example2: Low-Power Amplifier

Method	HV	Max $Gain_w$	Max UGF_w	Max PM_w
BNNBO-200	**10725.00**	101.78	**1.23**	**93.74**
MOEA/D-200	6036.89	68.87	0.98	72.96
MOBO-200	8001.19	84.78	1.20	74.39
MOEA/D-400	9456.72	**103.77**	1.16	74.46
MOBO-400	9938.76	99.84	1.22	77.18

- The discrepancy in Max $Gain_w$ is small enough to be neglected
- BNNBO-200 performs better than MOEA/D-400 and MOBO-400.

Conclusions

- **Bayesian Neural Network (BNN)**
  - Trained by ADVI method
  - Probabilistic prediction obtained by MC method
- **Bayesian Optimization**
  - Solve the optimum of LCB function by modified MOEA/D
  - LCB asymptotically approaches true PF
- **Future work**
  - BNN structure for different circuits


THANKS