



# Bayesian Inference on Introduced General Region: An Efficient Parametric Yield Estimation Method for Integrated Circuits

Zhengqi Gao<sup>1</sup>, Zihao Chen<sup>1</sup>, Jun Tao<sup>1</sup>, Yangfeng Su<sup>2</sup>, Dian Zhou<sup>3</sup>, Xuan Zeng<sup>1</sup>

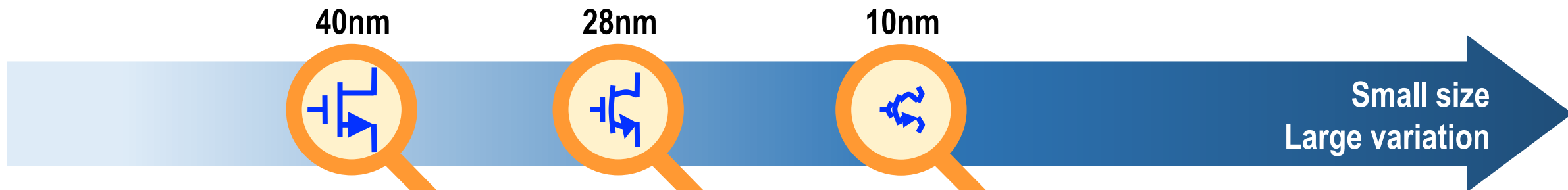
<sup>1</sup> ASIC& System State Key Lab, School of Microelectronics, Fudan University, Shanghai, China

<sup>2</sup> School of Mathematical Sciences, Fudan University, Shanghai, China

<sup>3</sup> University of Texas at Dallas, Dallas, USA

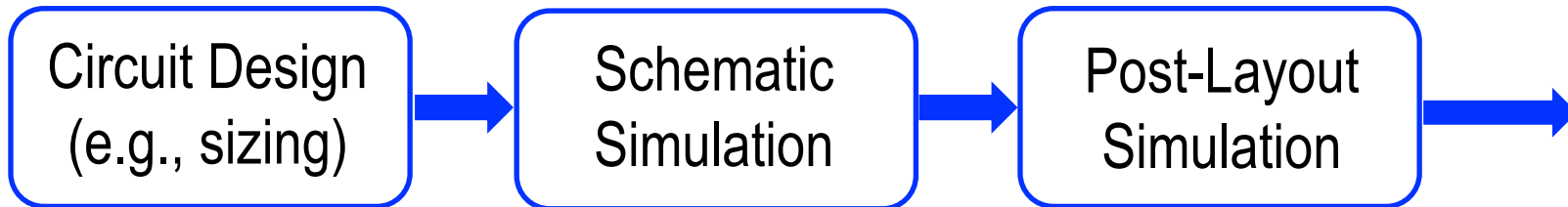
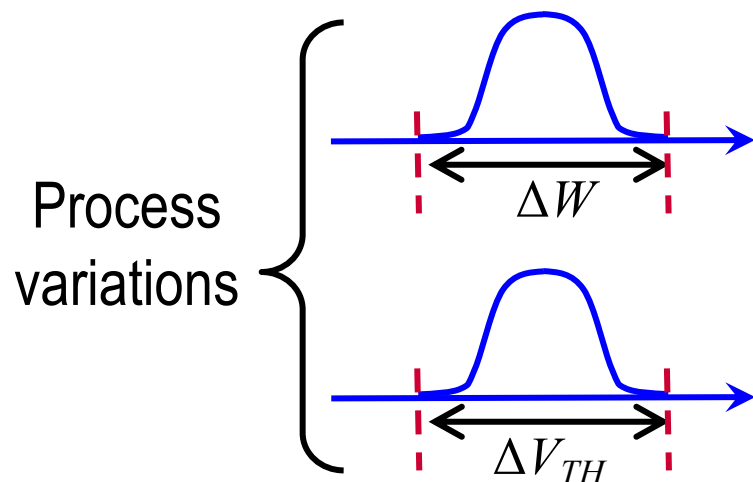


# IC Technology Scaling

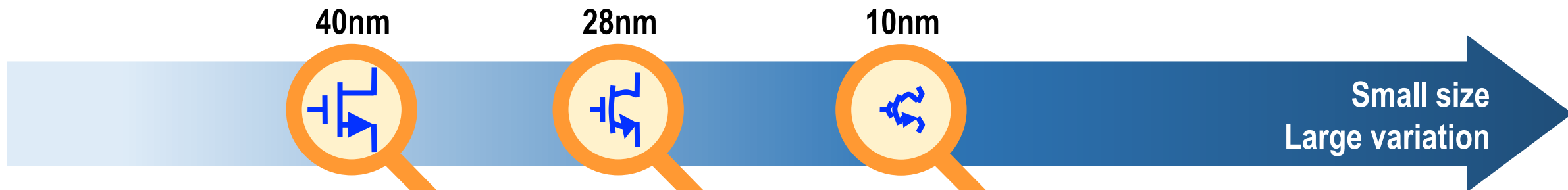


$$W = \bar{W} + \Delta W$$

$$V_{TH} = \bar{V}_{TH} + \Delta V_{TH}$$

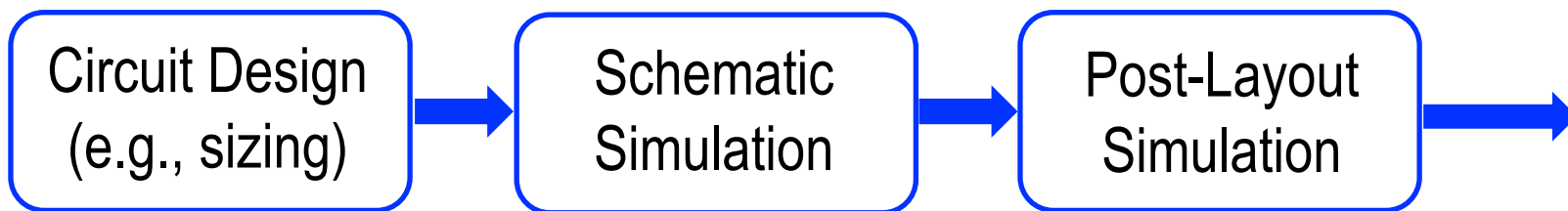
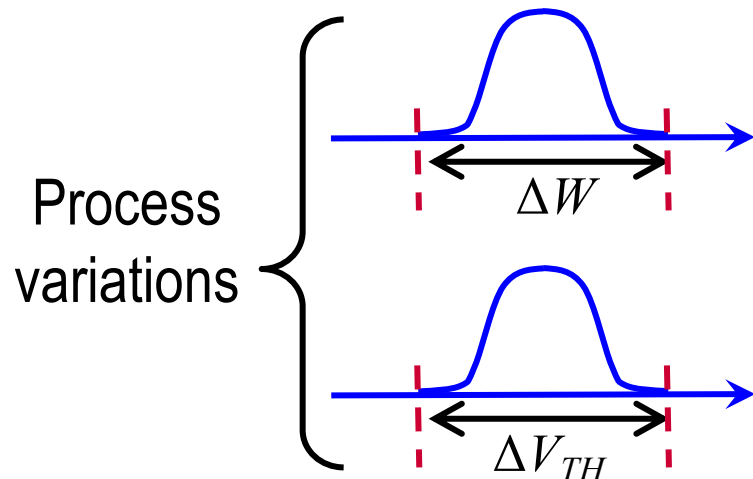


# IC Technology Scaling



$$W = \bar{W} + \Delta W$$

$$V_{TH} = \bar{V}_{TH} + \Delta V_{TH}$$



Post-layout simulation is time consuming ~ several Mins

Schematic simulation is affordable ~ several Secs

# Background

- **Conventional approach**

- Brute force Monte-Carlo

$$\beta^{l,\text{MC}} = \frac{1}{N^l} \cdot \sum_{n=1}^{N^l} y_n^l, \text{ based on dataset } D^l = \{(\mathbf{x}_n^l, y_n^l); n = 1, 2, \dots, N^l\}$$

$\mathbf{x}_n^l$ : the n-th sampled design feature vector at the late stage

$y_n^l$ : the n-th simulated result (i.e., '1' for pass, '0' for fail)

$N^l$ : the number of samples at the late stage

- **Bayesian model fusion (BMF)**

- Encode the knowledge learnt at the early stage by a prior distribution  $p(\beta^{l,\text{BMF}} | a, b)$
- Use it to estimate the late-stage yield  $p(\beta^{l,\text{BMF}} | a, b, \{y_n^l\}) = p(\{y_n^l\} | \beta^{l,\text{BMF}}) \cdot p(\beta^{l,\text{BMF}} | a, b)$



# Background

- **Conventional approach**

- Brute force Monte-Carlo

$$\beta^{l,MC} = \frac{1}{N^l} \cdot \sum_{n=1}^{N^l} y_n^l, \text{ based on dataset } D^l = \{(\mathbf{x}_n^l, y_n^l); n = 1, 2, \dots, N^l\}$$

$\mathbf{x}_n^l$ : the n-th sampled design feature vector at the late stage

$y_n^l$ : the n-th simulated result (i.e., '1' for pass, '0' for fail)

$N^l$ : the number of samples at the late stage

- **Bayesian model fusion (BMF)**

- Encode the knowledge learnt at the early-stage by a prior distribution  $p(\beta^{l,BMF} | a, b)$

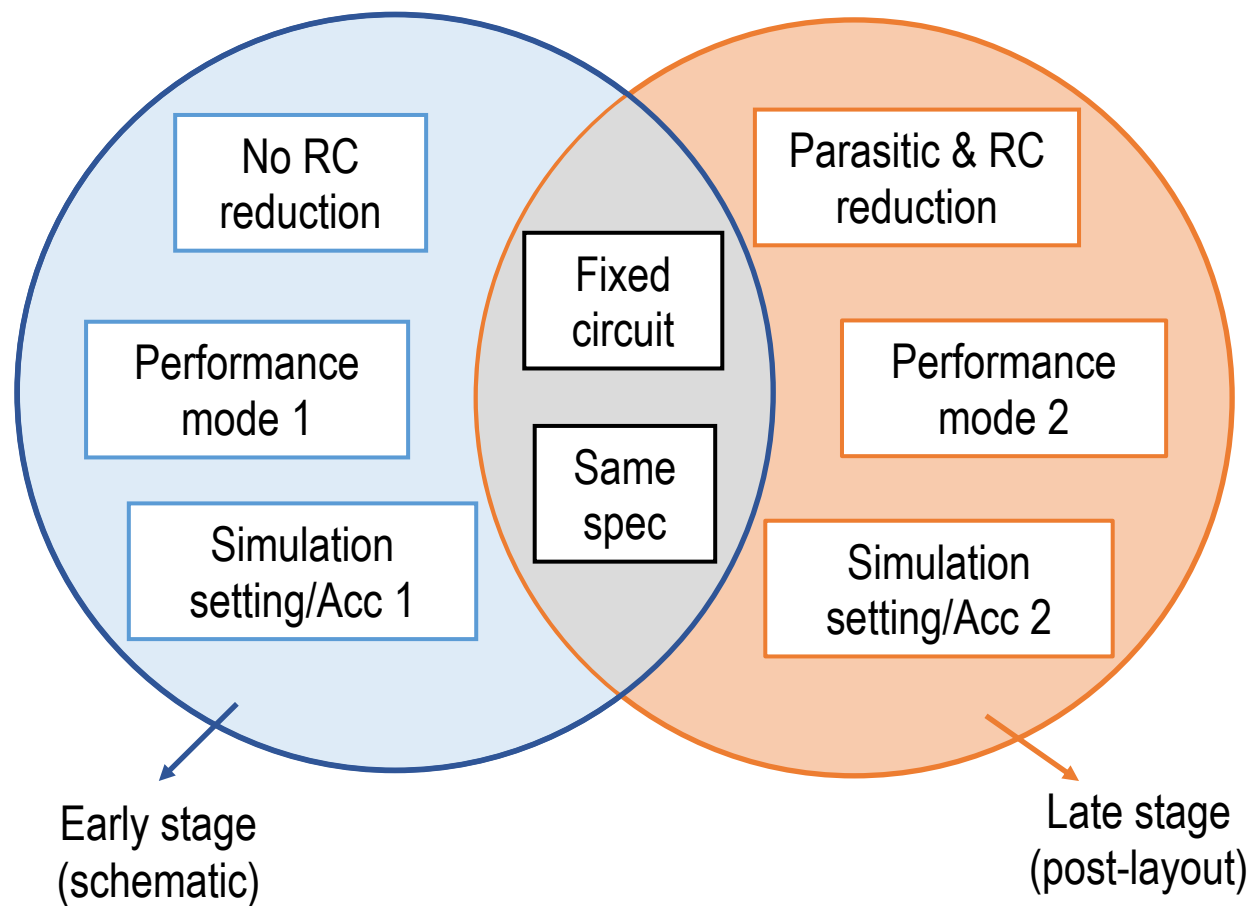
- Use it to estimate the late-stage yield  $p(\beta^{l,BMF} | a, b, \{y_n^l\}) = p(\{y_n^l\} | \beta^{l,BMF}) \cdot p(\beta^{l,BMF} | a, b)$



(1) Neglect the info of  $\mathbf{x}$ ! But the mapping  $\mathbf{x}^l \rightarrow y^l$  and  $\mathbf{x}^e \rightarrow y^e$  should be similar

(2) Early- and late- stages used asymmetrically

# Yield Estimation



common features: fixed circuit topology, etc.

distinct features: RC parasitic, etc.

# Yield Estimation

- Introduce classifiers

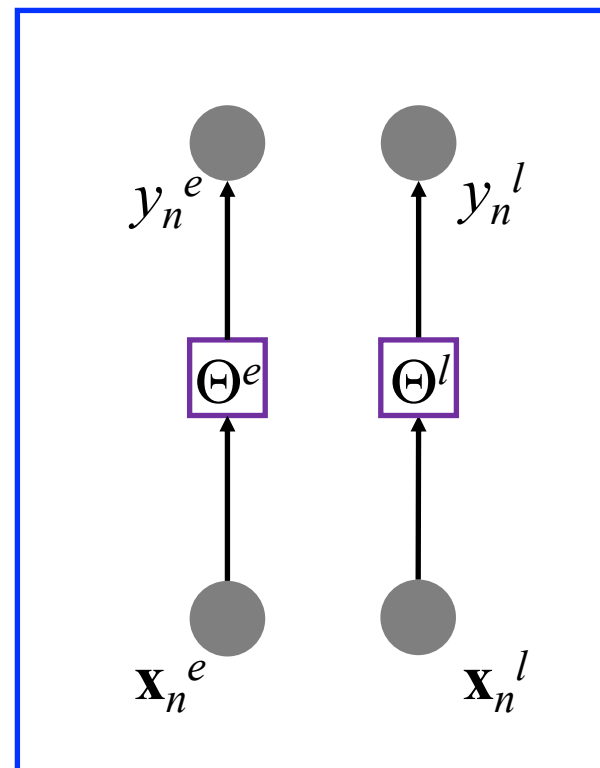
$$p(y^e = b | \mathbf{x}^e, \Theta^e) = \frac{\exp\left[(\lambda_b^e)^T \cdot \mathbf{f}(\mathbf{x}^e)\right]}{\exp\left[(\lambda_0^e)^T \cdot \mathbf{f}(\mathbf{x}^e)\right] + \exp\left[(\lambda_1^e)^T \cdot \mathbf{f}(\mathbf{x}^e)\right]} = \frac{\exp\left[(\lambda_b^e)^T \cdot \mathbf{f}(\mathbf{x}^e)\right]}{C^e}$$

$$p(y^l = b | \mathbf{x}^l, \Theta^l) = \frac{\exp\left[(\lambda_b^l)^T \cdot \mathbf{f}(\mathbf{x}^l)\right]}{\exp\left[(\lambda_0^l)^T \cdot \mathbf{f}(\mathbf{x}^l)\right] + \exp\left[(\lambda_1^l)^T \cdot \mathbf{f}(\mathbf{x}^l)\right]} = \frac{\exp\left[(\lambda_b^l)^T \cdot \mathbf{f}(\mathbf{x}^l)\right]}{C^l}$$

$\Theta^e = \{\lambda_0^e, \lambda_1^e \in \mathcal{R}\}$ ,  $\Theta^l = \{\lambda_0^l, \lambda_1^l \in \mathcal{R}\}$ : classifier parameters

$b=0$  or  $1$ : class labels, '1' for pass, '0' for fail

$f(\bullet)$ : feature mapping

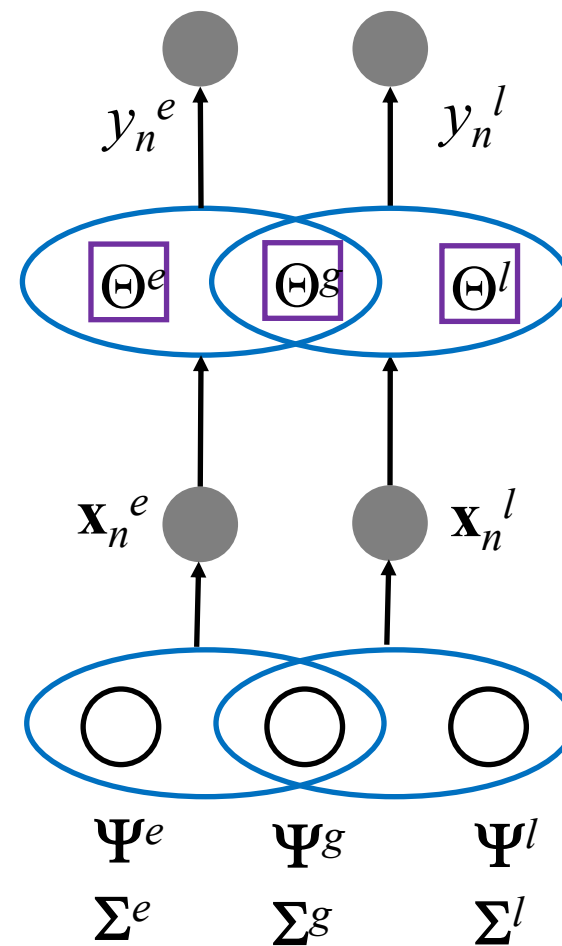




# Yield Estimation

- Introduce general region

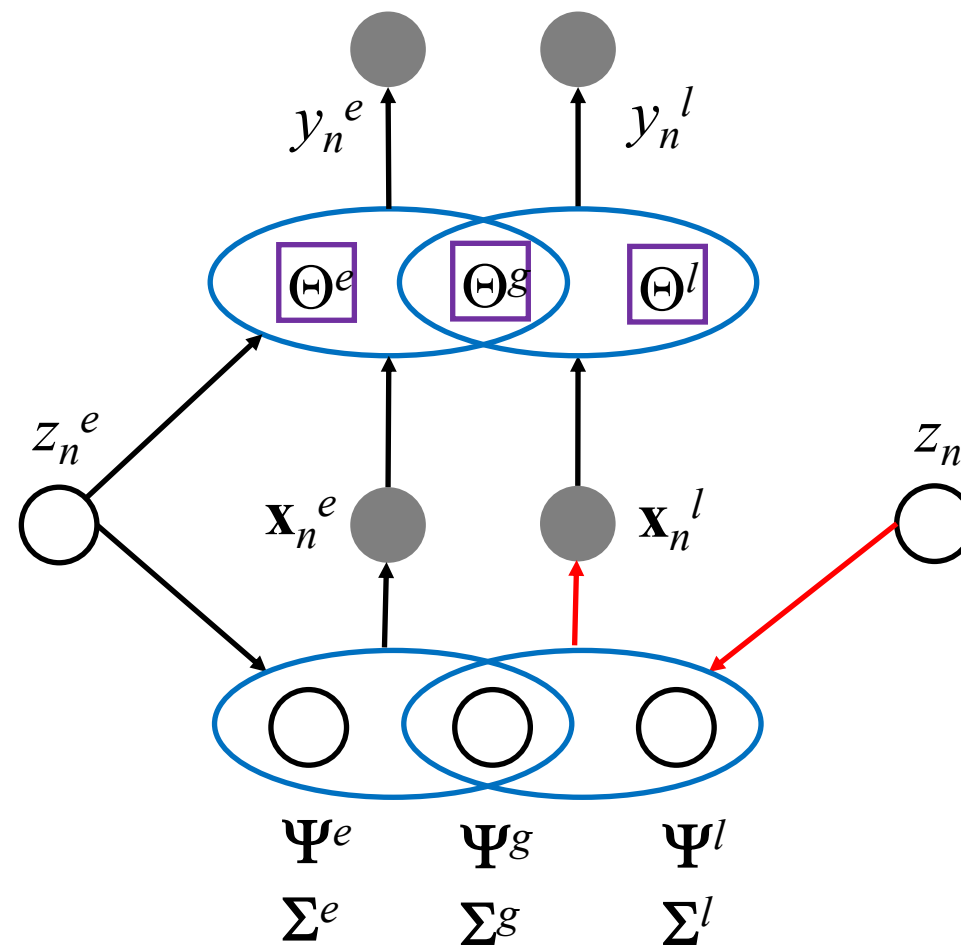
- $\mathbf{x}^l$  may generate from region related to  $\{\Psi^l, \Sigma^l\}$  or  $\{\Psi^g, \Sigma^g\}$ .
- How to parametrize the generation process of  $\mathbf{x}^l$ ?
- Multi-variate Gaussian distribution works.



# Yield Estimation

- Introduce general region
  - Control generating  $\mathbf{x}_n^l$

$$p(\mathbf{x}^l | z^l, \Psi^l, \Sigma^l, \Psi^g, \Sigma^g) = \begin{cases} N(\mathbf{x}^l | \Psi^g, \Sigma^g) & \text{if } z^l = 0 \\ N(\mathbf{x}^l | \Psi^l, \Sigma^l) & \text{if } z^l = 1 \end{cases}$$



# Yield Estimation

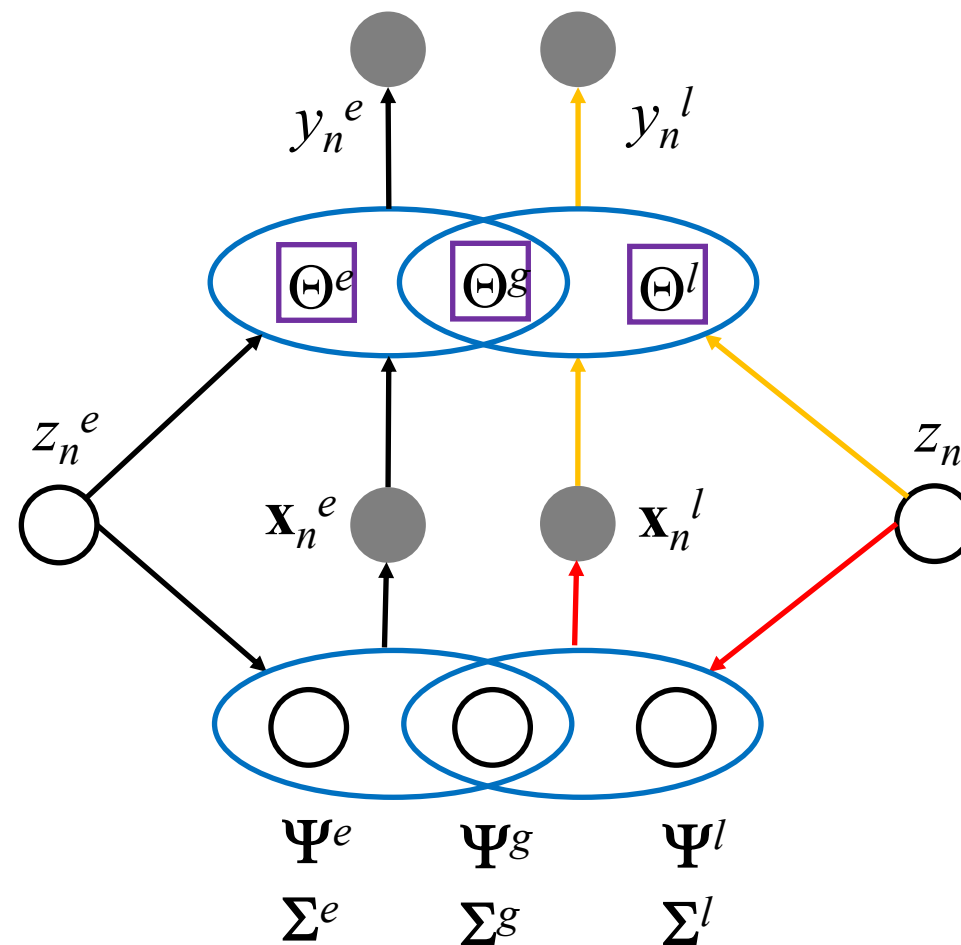
- Introduce general region

- Control generating  $\mathbf{x}_n^l$

$$p(\mathbf{x}^l | z^l, \Psi^l, \Sigma^l, \Psi^g, \Sigma^g) = \begin{cases} N(\mathbf{x}^l | \Psi^g, \Sigma^g) & \text{if } z^l = 0 \\ N(\mathbf{x}^l | \Psi^l, \Sigma^l) & \text{if } z^l = 1 \end{cases}$$

- Control generating  $\mathbf{y}_n^l$

$$p(y^l = b | \mathbf{x}^l, z^l, \Theta^l, \Theta^g) = \begin{cases} \exp\left[(\lambda_b^g)^T \cdot \mathbf{f}(\mathbf{x}^l)\right] / C^g & \text{if } z^l = 0 \\ \exp\left[(\lambda_b^l)^T \cdot \mathbf{f}(\mathbf{x}^l)\right] / C^l & \text{if } z^l = 1 \end{cases}$$





# Yield Estimation

- Introduce general region

- Control generating  $\mathbf{x}_n^l$

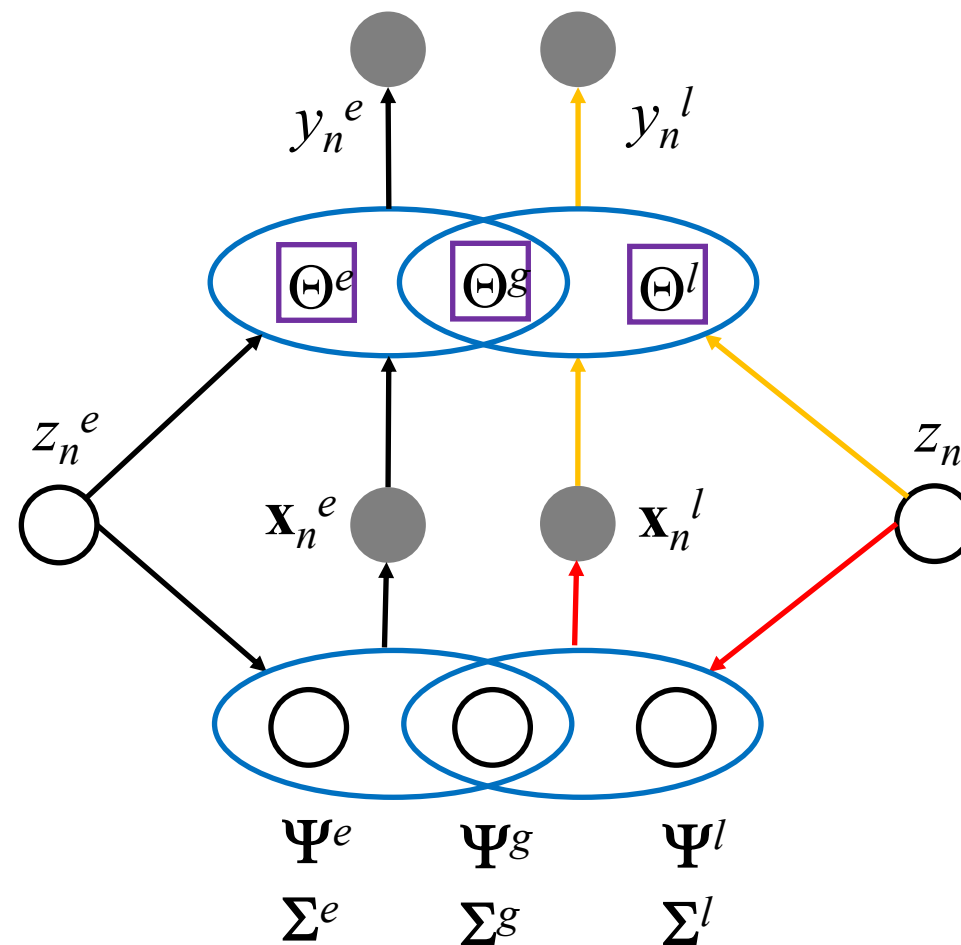
$$p(\mathbf{x}^l | z^l, \Psi^l, \Sigma^l, \Psi^g, \Sigma^g) = \begin{cases} N(\mathbf{x}^l | \Psi^g, \Sigma^g) & \text{if } z^l = 0 \\ N(\mathbf{x}^l | \Psi^l, \Sigma^l) & \text{if } z^l = 1 \end{cases}$$

- Control generating  $\mathbf{y}_n^l$

$$p(y^l = b | \mathbf{x}^l, z^l, \Theta^l, \Theta^g) = \begin{cases} \exp[(\lambda_b^g)^T \cdot \mathbf{f}(\mathbf{x}^l)] / C^g & \text{if } z^l = 0 \\ \exp[(\lambda_b^l)^T \cdot \mathbf{f}(\mathbf{x}^l)] / C^l & \text{if } z^l = 1 \end{cases}$$

$z_n^l$  controls using  $l$  or  $g$ !

But how to set  $z_n^l$ ?



# Yield Estimation

- Introduce general region

- Control generating  $\mathbf{x}_n^l$

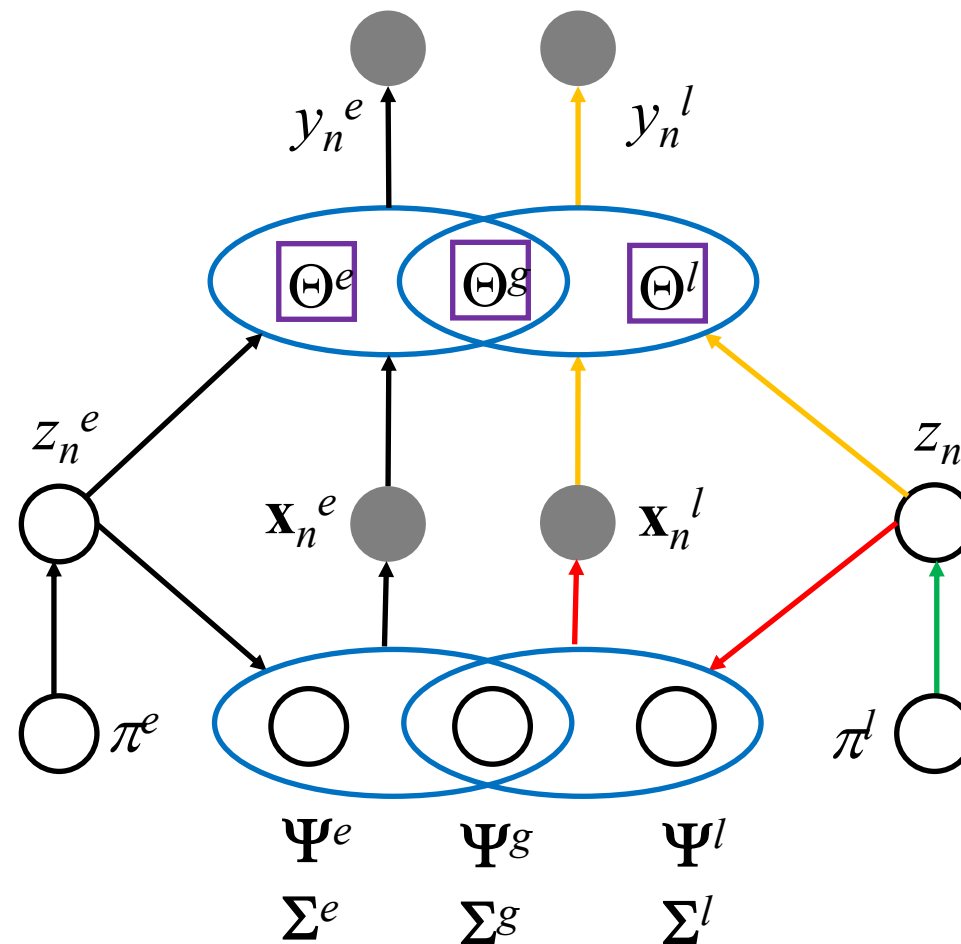
$$p(\mathbf{x}^l | z^l, \Psi^l, \Sigma^l, \Psi^g, \Sigma^g) = \begin{cases} N(\mathbf{x}^l | \Psi^g, \Sigma^g) & \text{if } z^l = 0 \\ N(\mathbf{x}^l | \Psi^l, \Sigma^l) & \text{if } z^l = 1 \end{cases}$$

- Control generating  $\mathbf{y}_n^l$

$$p(y^l = b | \mathbf{x}^l, z^l, \Theta^l, \Theta^g) = \begin{cases} \exp[(\lambda_b^g)^T \cdot \mathbf{f}(\mathbf{x}^l)] / C^g & \text{if } z^l = 0 \\ \exp[(\lambda_b^l)^T \cdot \mathbf{f}(\mathbf{x}^l)] / C^l & \text{if } z^l = 1 \end{cases}$$

- Control generating  $\mathbf{z}_n^l$

$$p(z^l | \pi^l) = (\pi^l)^{z^l} \cdot (1 - \pi^l)^{1-z^l}$$



# Yield Estimation

- Complete graphical model

- Parameters construct a high-dimensional space

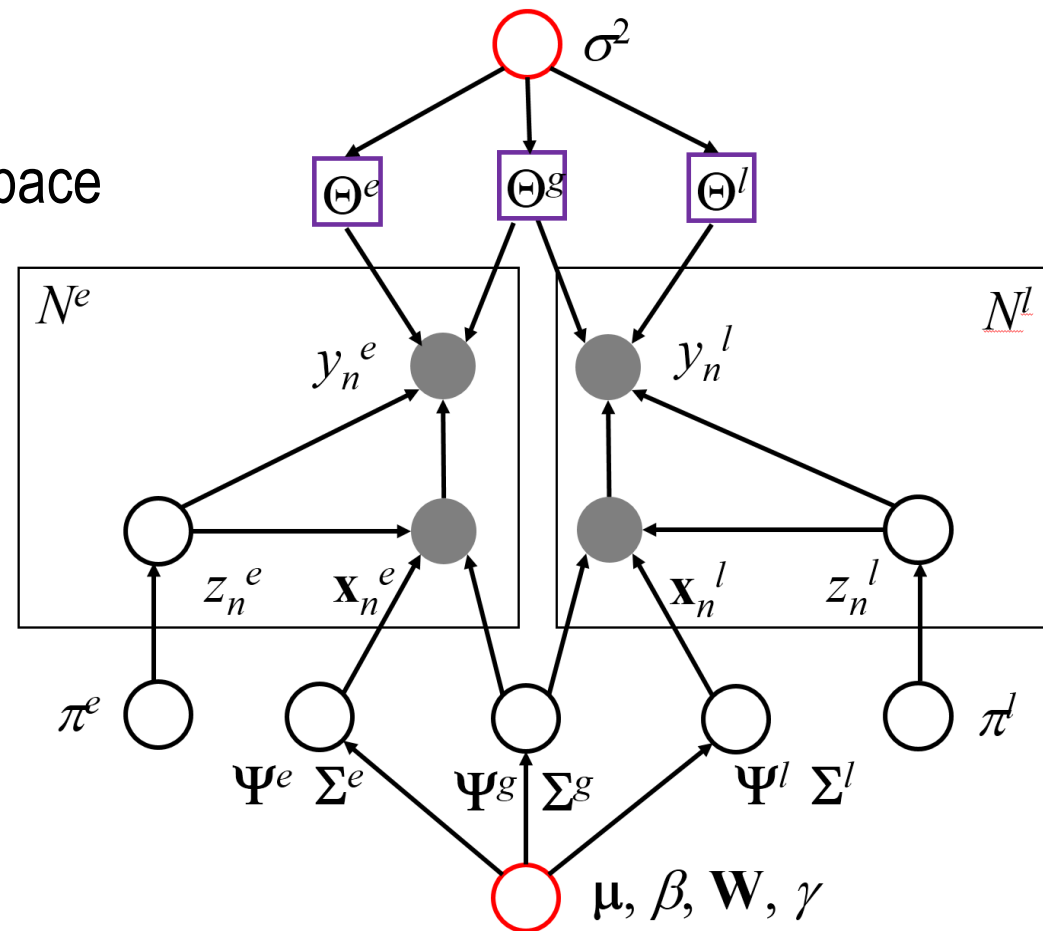
- Overfitting might occur

- Include regularization terms

➔ Prior distributions

- Define a loss function, solved by CEM

- Estimate yield by sampling method





# Example1: Charge Pump

- 40nm CMOS Technology
- Five performance metrics

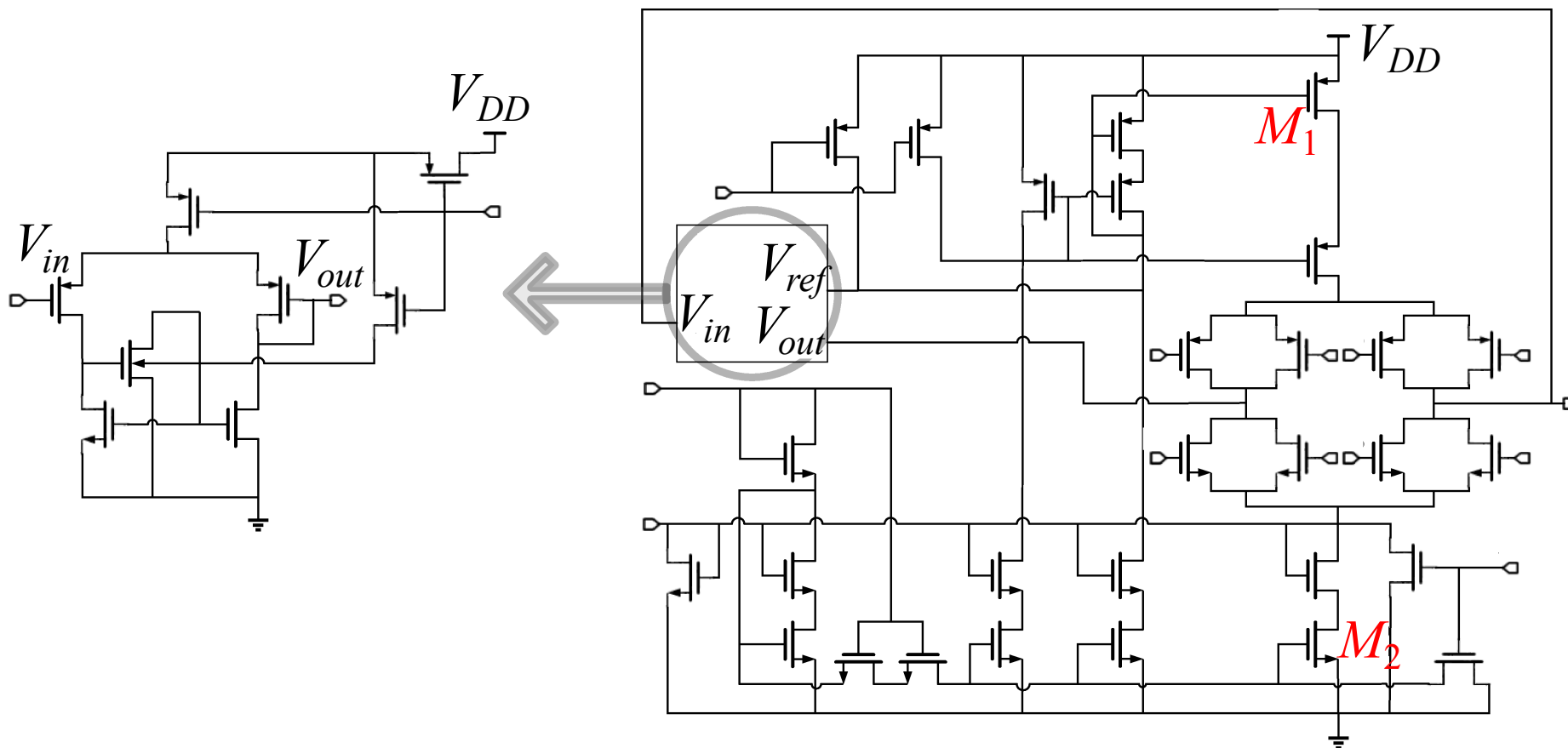
$\Lambda$ : process corner, i.e., {TT, SS, FF, FS, SF}

$I_{sta}$ : design spec current

$$diff_1 = \max_{\Lambda} (I_{M_1,max} - I_{M_1,avg}), \quad diff_2 = \max_{\Lambda} (I_{M_1,avg} - I_{M_1,min})$$

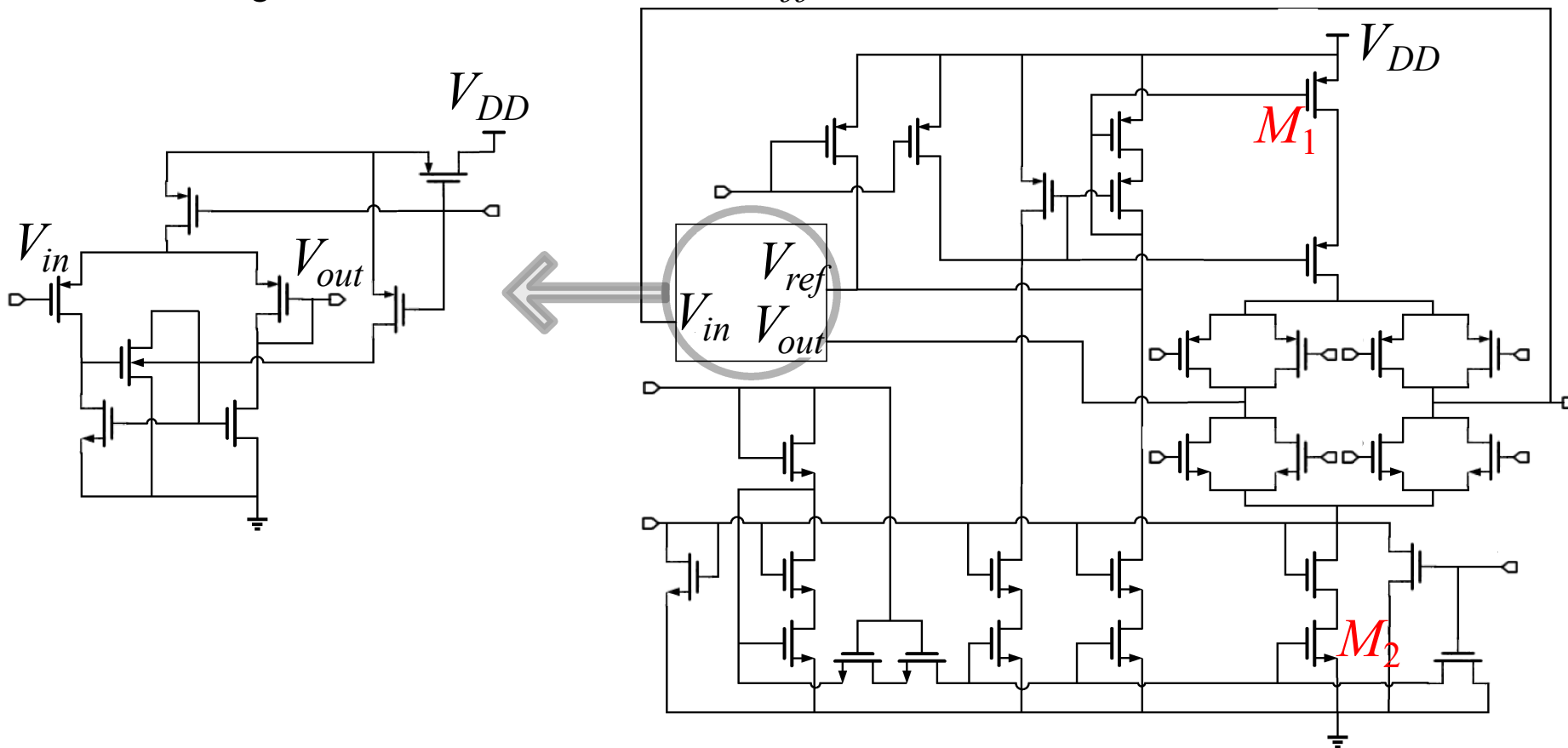
$$diff_3 = \max_{\Lambda} (I_{M_2,max} - I_{M_2,avg}), \quad diff_4 = \max_{\Lambda} (I_{M_2,avg} - I_{M_2,min})$$

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

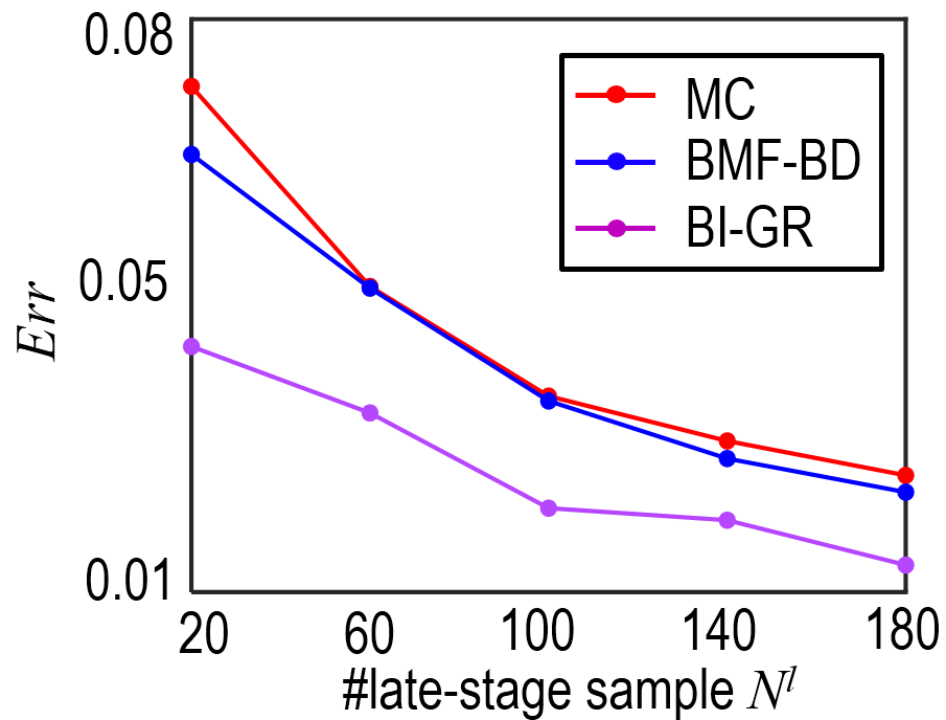


# Example1: Charge Pump

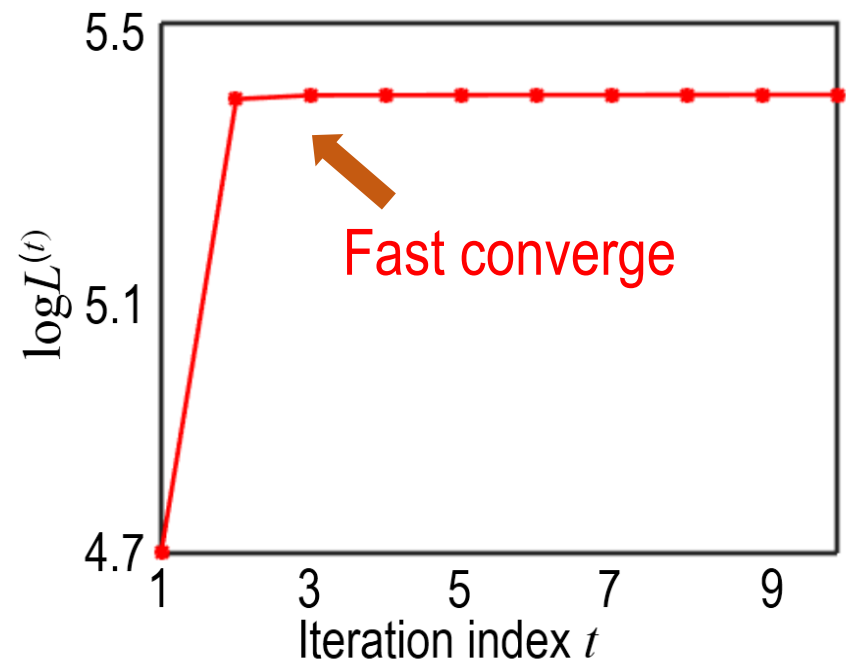
- 40nm CMOS Technology
- Five performance metrics
- 16 design variables
- Good design: small *devi* and small *diff*



# Example1: Charge Pump



- BMF-BD is slightly better than MC
- Proposed BI-GR performs the best

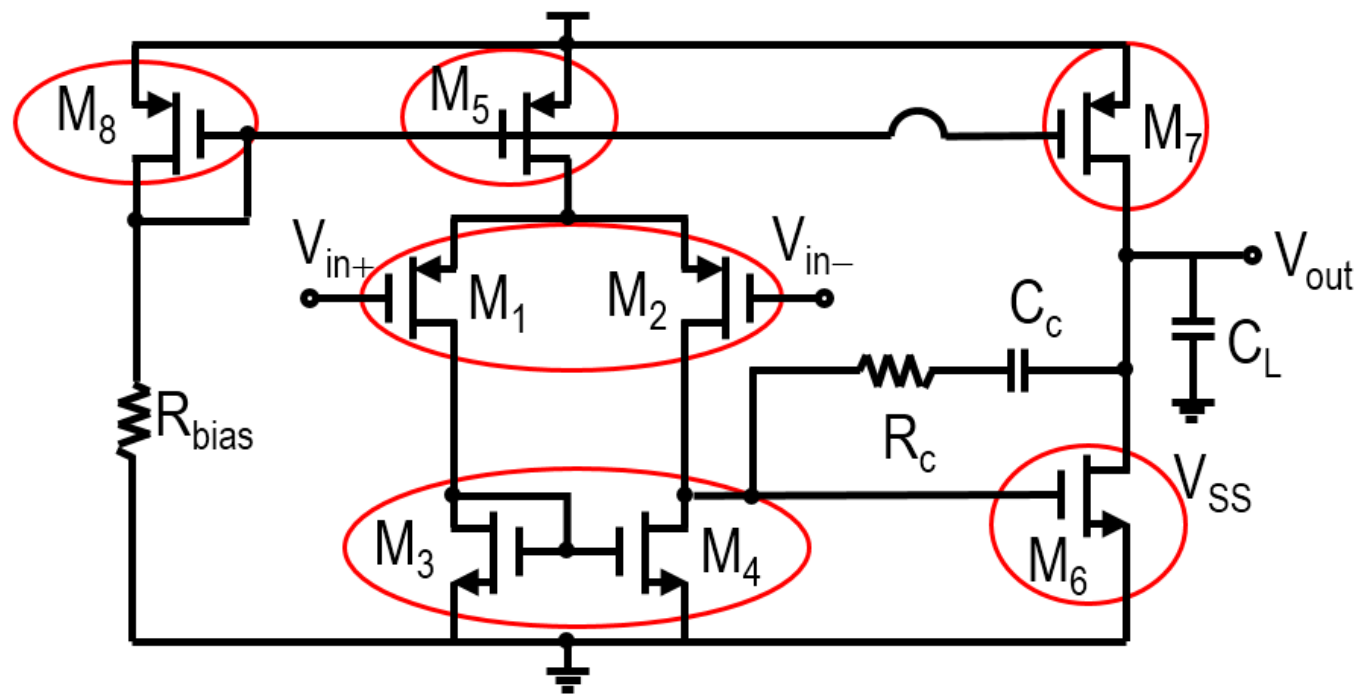


$N^l$	Absolute error Err		
	MC	BMF-BD	BI-GR
20	0.072	0.063	<b>0.039</b>
60	0.047	0.046	<b>0.031</b>
100	0.033	0.032	<b>0.019</b>
140	0.027	0.025	<b>0.017</b>
180	0.023	0.021	<b>0.012</b>



## Example2: Op-amp

- 40nm CMOS Technology
- Four performance metrics: power, unit bandwidth frequency, phase margin, and gain
- 6 design variables



## Example2: Op-amp

$N^l$	Absolute error Err		
	MC	BMF-BD	BI-GR
20	0.087	0.087	<b>0.085</b>
60	0.055	0.054	<b>0.043</b>
100	0.037	0.035	<b>0.028</b>
140	0.032	0.032	<b>0.027</b>
180	0.030	0.028	<b>0.022</b>

- The proposed BI-GR performs the best



# Conclusions

- **Late-stage yield estimation method**
  - Introduce classifiers for the mapping  $\mathbf{x}^l \rightarrow \mathbf{y}^l$
  - Encode the correlation symmetrically

