UCLA

International Symposium on Physical Design

sigda

LVFGen: Efficient Liberty Variation Format (LVF) Generation Using Variational Analysis and Active Learning

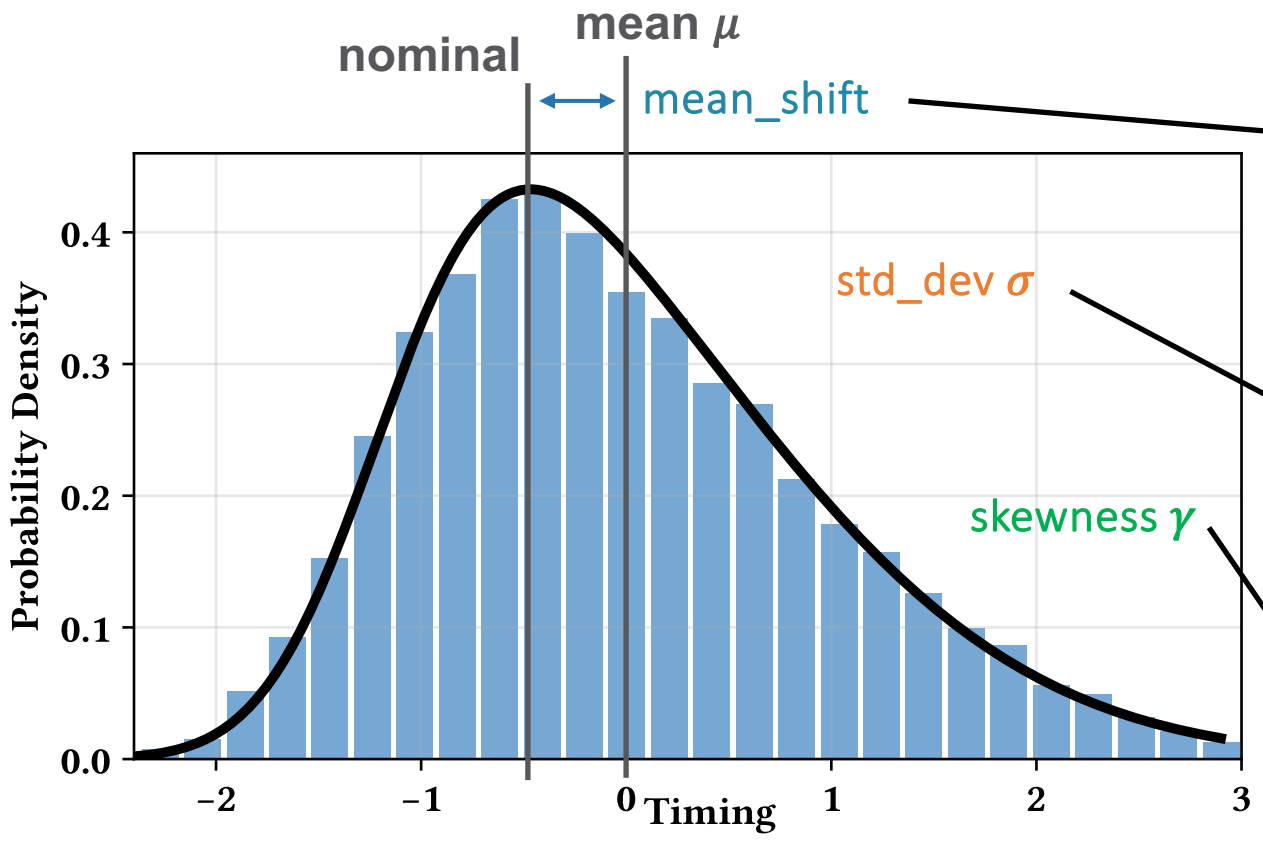
Junzhuo Zhou[*], Ting-Jung Lin[*], Haoxuan Xia, Li Huang,

Wei Xing[†], and Lei He[†]

University of California, Los Angeles

Statistical Characterization is the Foundation for SSTA

LVF uses 3 moments to model PDF of a cell performance

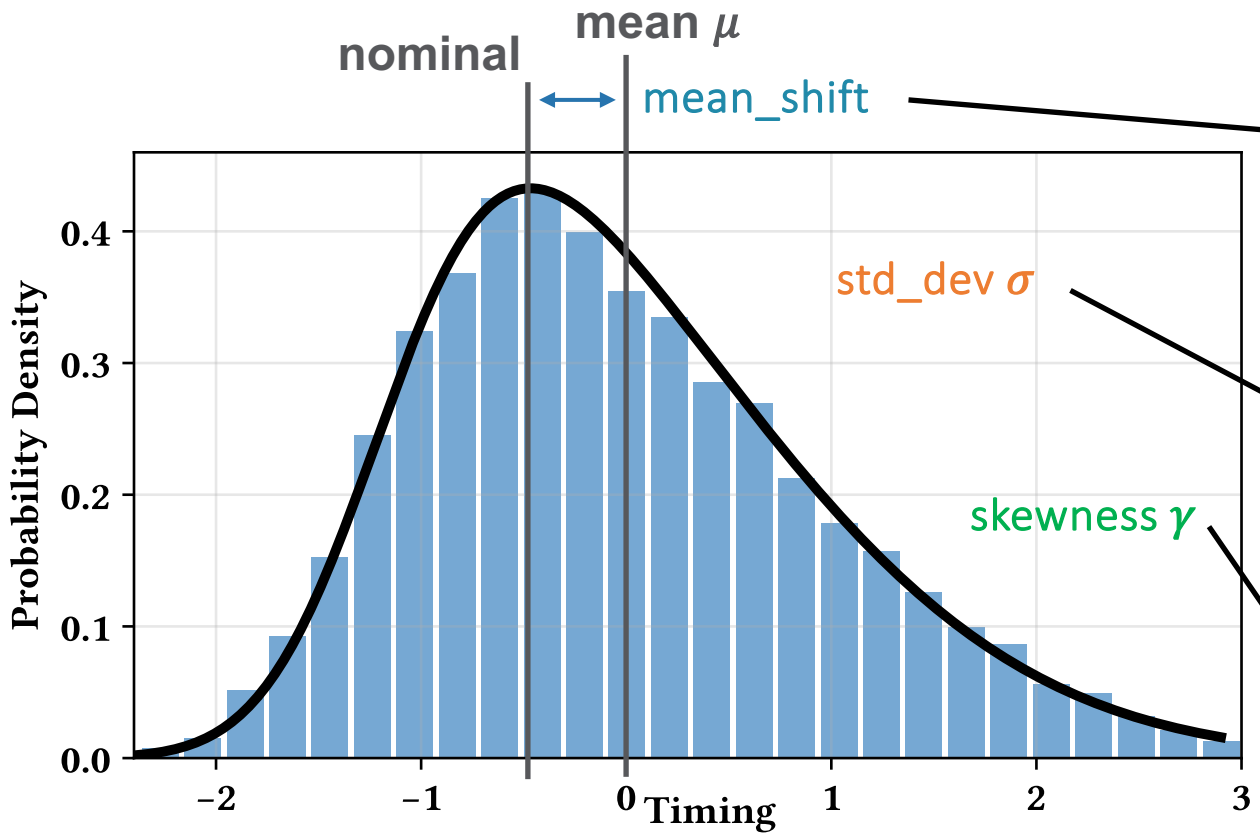



```
ocv_mean_shift_xx() {
index_1( "0.1, 0.5, …" );
index_2( "0.1, 0.5, …" );
values ( \
"0.00, 0.12, …", \
"0.14, 0.16, …", \
…

ocv_std_dev_xx() { …
values ( \
"1.0, 1.2, …", \
"1.1, 1.4, …", \
…

ocv_skewness_xx() { …
values ( \
"0.7, 0.9, …", \
"0.8, 0.9, …", \
```

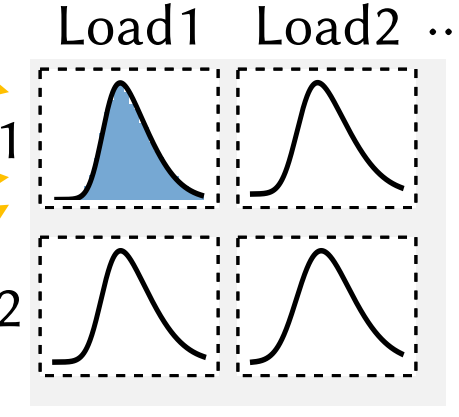
LVF is necessary for 22nm and below

Statistical Characterization is the Foundation for SSTA

LVF uses 3 moments to model PDF of a cell performance




```
ocv_mean_shift_xx() {
index_1( "0.1, 0.5, …" );
index_2( "0.1, 0.5, …" );
values ( \
"0.00, 0.12, …", \
"0.14, 0.16, …", \
…

ocv_std_dev_xx() { …
values ( \
"1.0, 1.2, …", \
"1.1, 1.4, …", \
…

ocv_skewness_xx() { …
values ( \
"0.7, 0.9, …", \
"0.8, 0.9, …", \
```

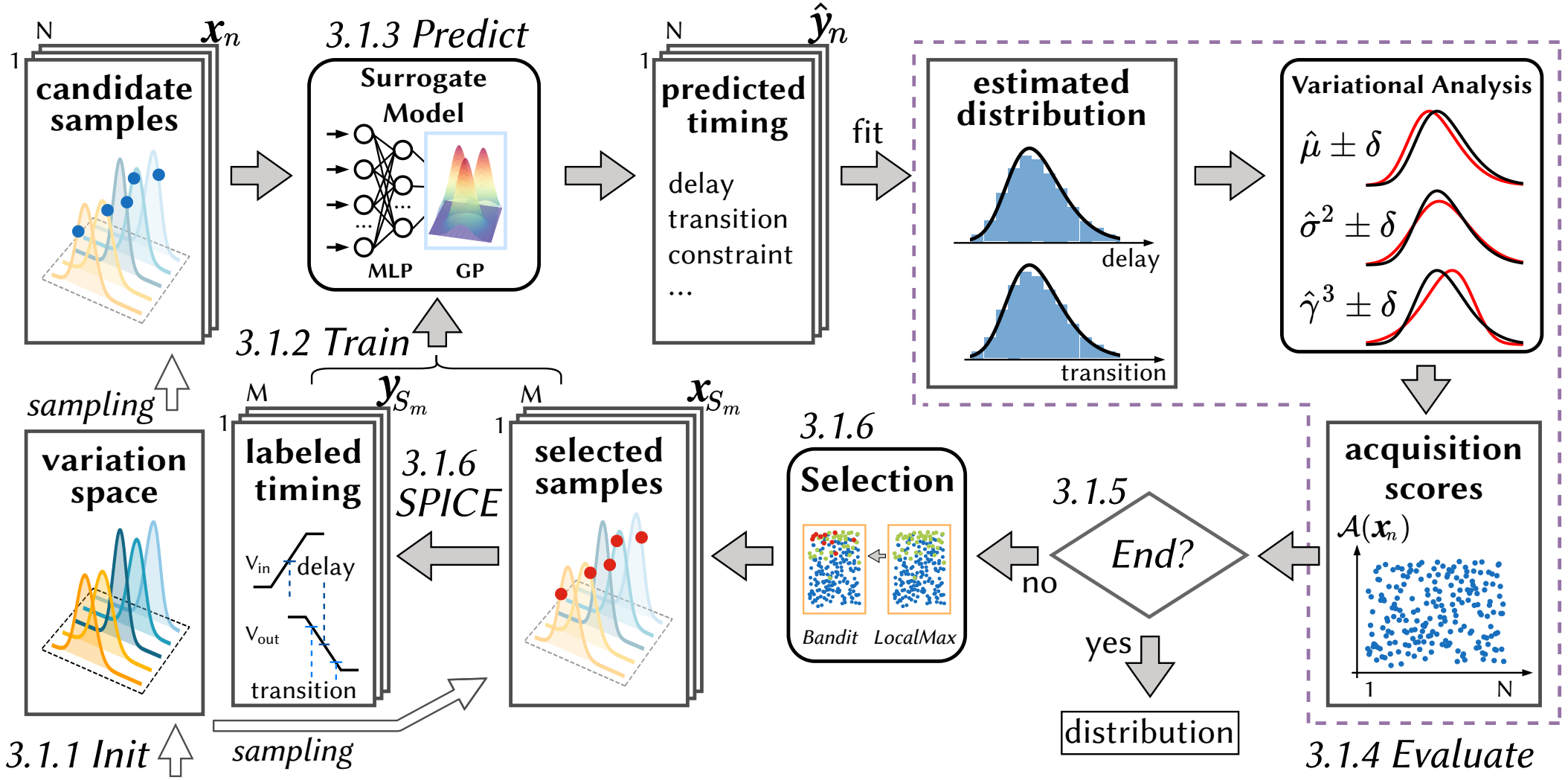


LVF is necessary for 22nm and below

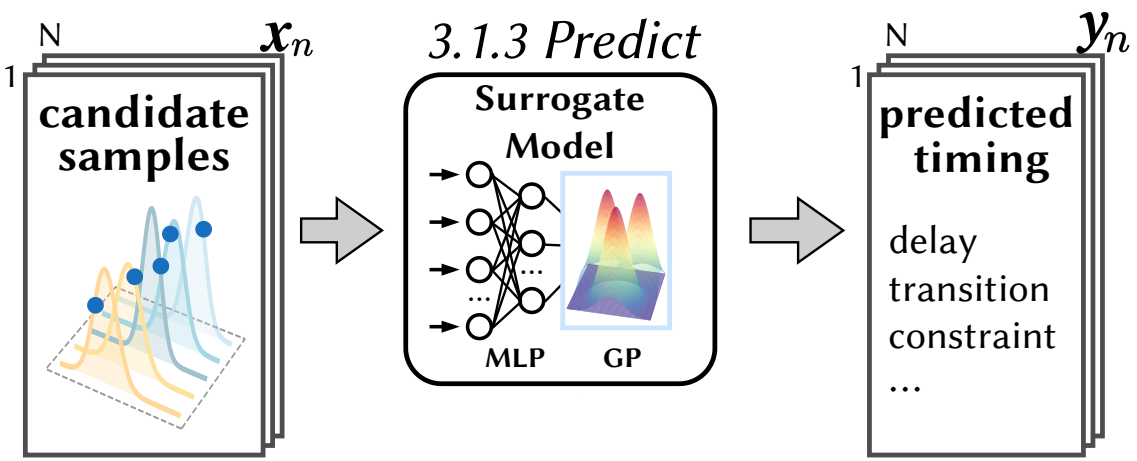
LVFGen

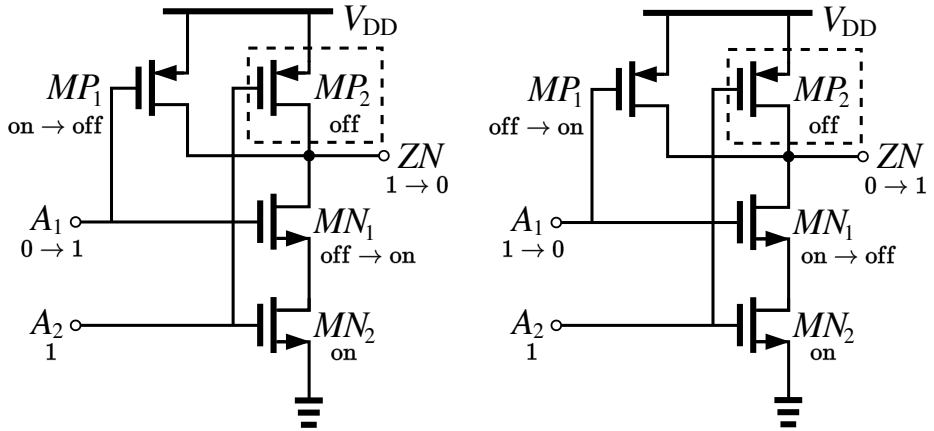
- Traditionally, using Monte Carlo simulation to estimate LVF's 3 moments
- Could be improved, as the process variations impact timing in a predictable way
- Small samples to train a surrogate model for timing distribution
- To select the most effective samples, according to proposed acquisition
- The first indepth study

LVFGen: Active Learning Framework



Timing Prediction by Surrogate Model





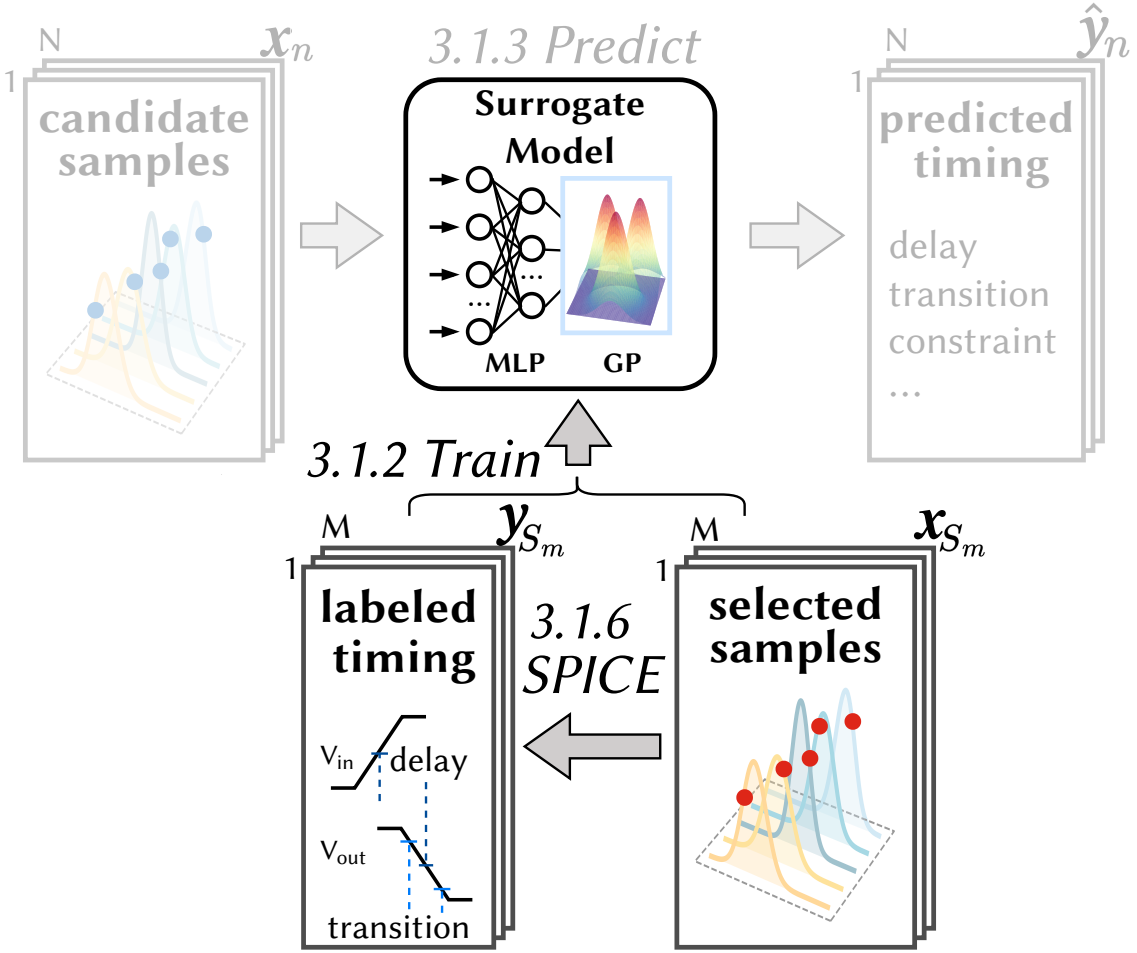
(a) A_1 **rising,** ZN **falling** **(b)** A_1 **falling,** ZN **rising**

Figure 2: Dimension pruning for NAND2

- DeepGP as surrogate model, regression task
  $\hat{y}_n = g_2(g_1(x_n)) \sim \mathcal{N}(\mu(x_n), v(x_n))$
  - g_1: Multi-layer Perceptron, MLP
  - g_2: Gaussian Process, GP

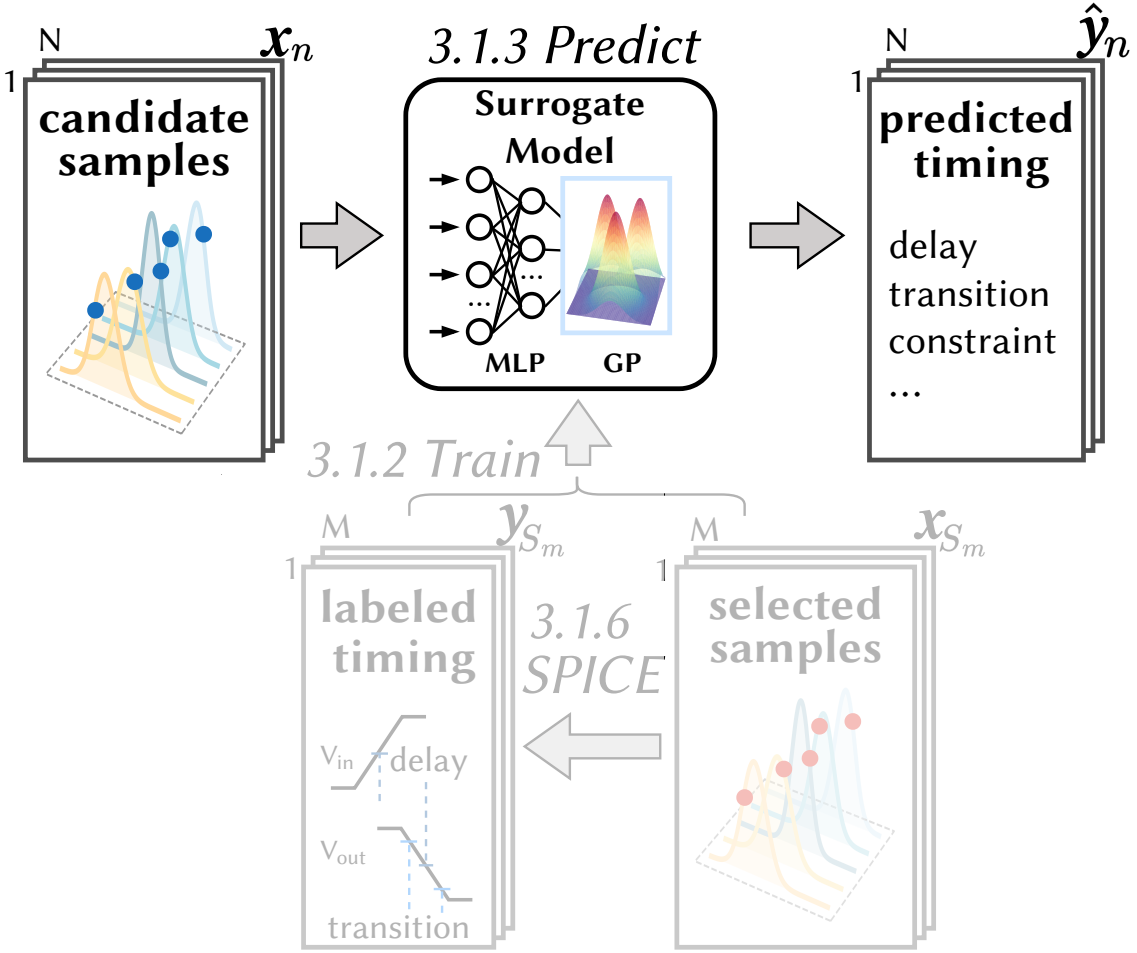
- **Dimension Pruning**
  - GP can predict uncertainty
  - But GP preforms bad in high dimensionality
  - Firstly fit & prune dimensions by MPL
  - Then pass the low-dimension data to GP

Timing Prediction by Surrogate Model



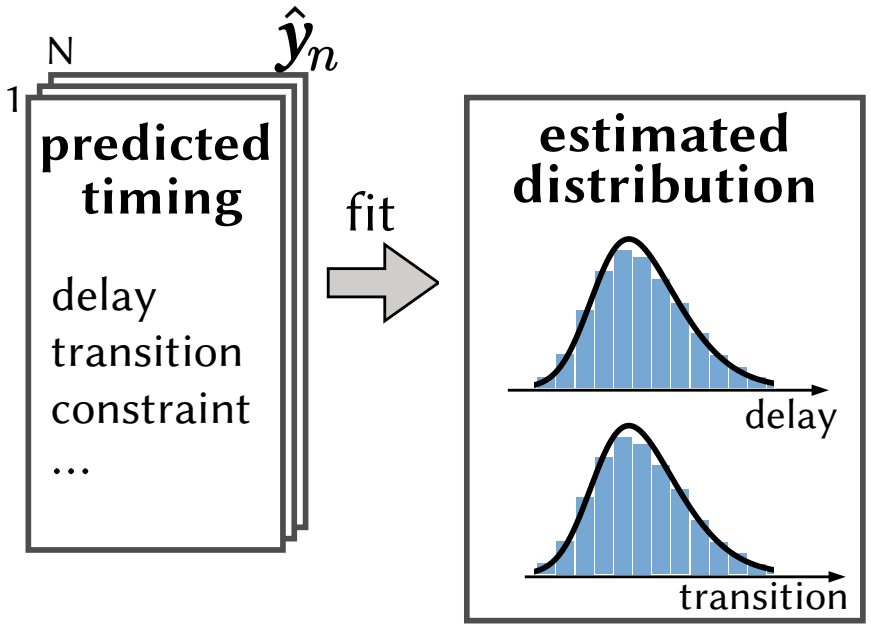
- DeepGP as surrogate model, regression task
  $\hat{y}_n = g_2(g_1(x_n)) \sim \mathcal{N}(\mu(x_n), v(x_n))$
  - g_1: Multi-layer Perceptron, MLP
  - g_2: Gaussian Process, GP
- Training it with observed sample **X** & timing **Y**

Timing Prediction by Surrogate Model



- DeepGP as surrogate model, regression task
  $\hat{y}_n = g_2(g_1(x_n)) \sim \mathcal{N}(\mu(x_n), v(x_n))$
  - g_1: Multi-layer Perceptron, MLP
  - g_2: Gaussian Process, GP
- Training it with observed sample **X** & timing **Y**
- GP predict all candidate **X** set
  - Get both **Y**'s expectation and **uncertainty**
  - **Y**'s expectation to estimate timing dist.
  - **Y**'s **uncertainty** to evaluate acquisition

Estimate Timing Distribution from Prediction



$$\hat{\mu} = \frac{\sum_{n=1}^{N} \hat{y}_n}{N}$$

$$\hat{\sigma}^2 = \frac{\sum_{n=1}^{N} (\hat{y}_n - \hat{\mu})^2}{N}$$

$$\hat{\gamma}^3 = \frac{\sum_{n=1}^{N} (\hat{y}_n - \hat{\mu})^3}{N}$$

- Given $\hat{y}_n \sim \mathcal{N}(\mu(x_n), v(x_n))$
- Easily have

$$E[\hat{\mu}] = \frac{\sum_{n=1}^{N} v(x_n)}{N}$$

$$Var[\hat{\mu}] = \frac{\sum_{n=1}^{N} v(x_n)}{N}$$

- Substitution

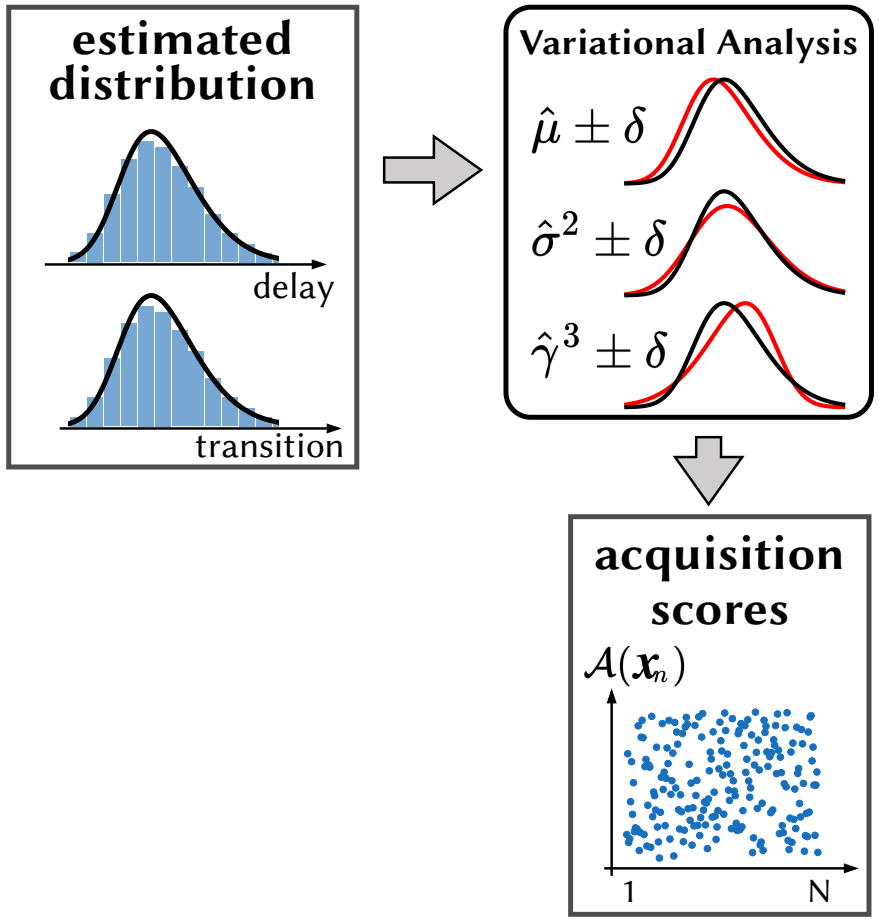
$$(\hat{y}_n - \hat{\mu}) \sim \mathcal{N}(\mu'_n, v'_n)$$
$$\mu'_n = \mu(x_n) - E[\hat{\mu}]$$
$$v'_n = v(x_n) + Var[\hat{\mu}]$$

- Finally

$$E[\hat{\sigma}^2] = \frac{\sum_{n=1}^{N} (\mu'^2_n + v'_n)}{N}$$

$$E[\hat{\gamma}^3] = \frac{\sum_{n=1}^{N} (\mu'^3_n + 3\mu'_n v'_n)}{N}$$

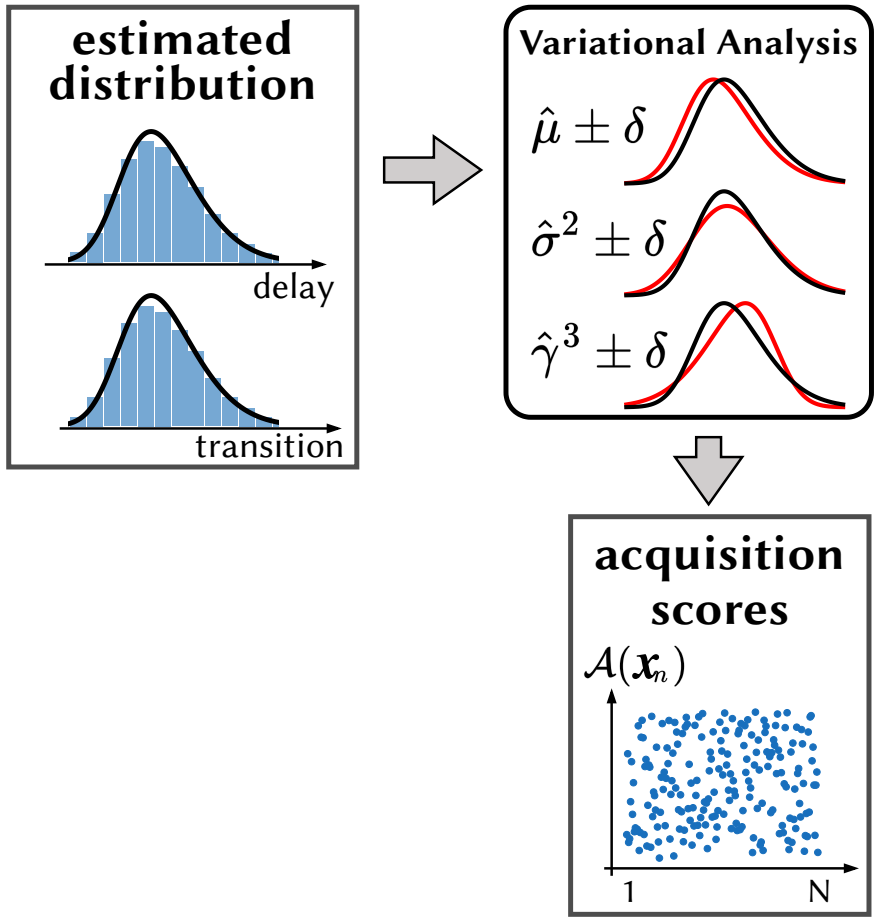
Evaluate Acquisition Score



- Goal: minimize the distance from estimation to real

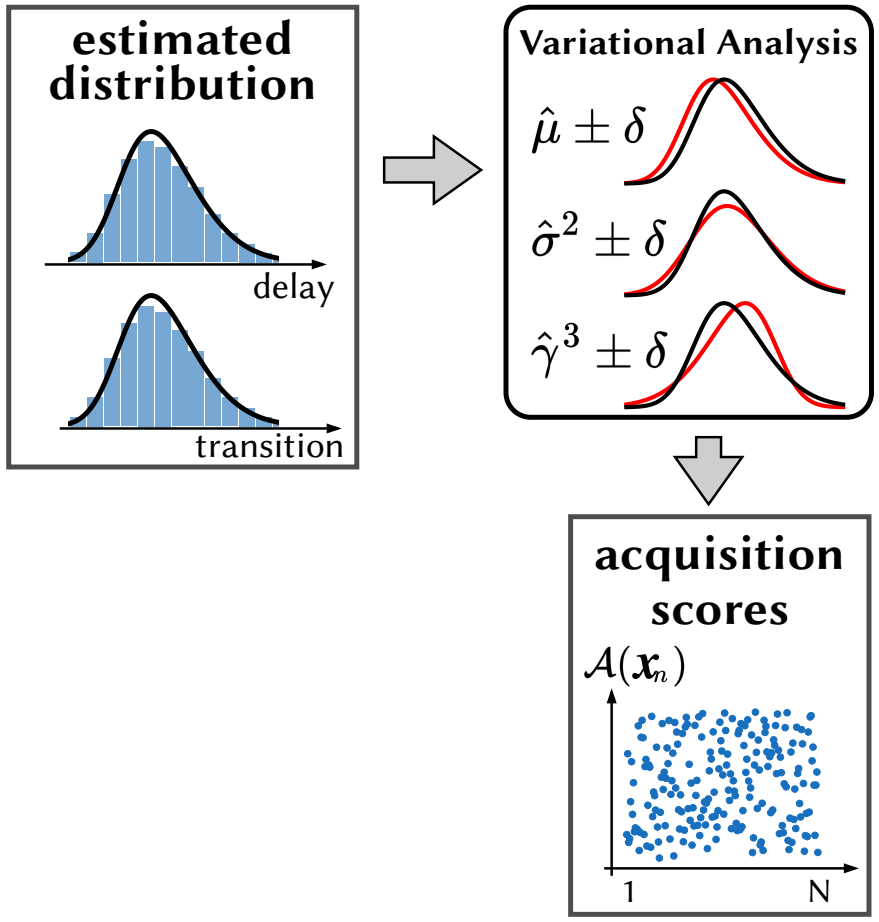
minimize $\mathcal{L} = D[LVF(\hat{\mu}, \hat{\sigma}^2, \hat{\gamma}^3) \mid LVF(\mu, \sigma^2, \gamma^3)]$

Evaluate Acquisition Score



- Goal: minimize the distance from estimation to real
- Acquisition Score $\mathcal{A}(x_n)$: **sample's contribution for estimated distribution's uncertainty**

Evaluate Acquisition Score



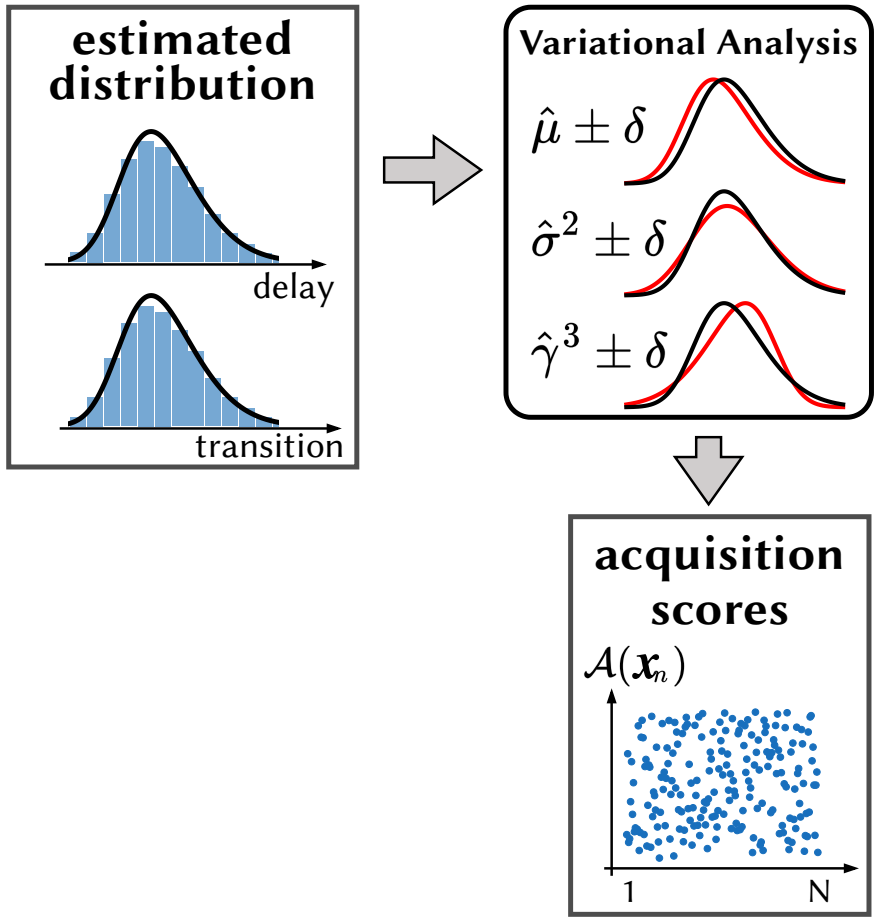
- Goal: minimize the distance from estimation to real
- Acquisition Score $\mathcal{A}(x_n)$: **sample's contribution for estimated distribution's uncertainty**
- Propagate from timing uncertainty to $\mathcal{A}(x_n)$
  - Timing uncertainty → Each moment uncertainty

$$Var[\hat{\mu}] = \frac{\sum_{n=1}^{N} v(x_n)}{N}$$

$$Var[\hat{\sigma}^2] = \frac{\sum_{n=1}^{N} (4\mu_n'^2 v_n' + 2v_n'^2)}{N}$$

$$Var[\hat{\gamma}^3] = \frac{\sum_{n=1}^{N} (9\mu_n'^4 v_n' + 36\mu_n'^2 v_n'^2 + 15v_n'^3)}{N}$$

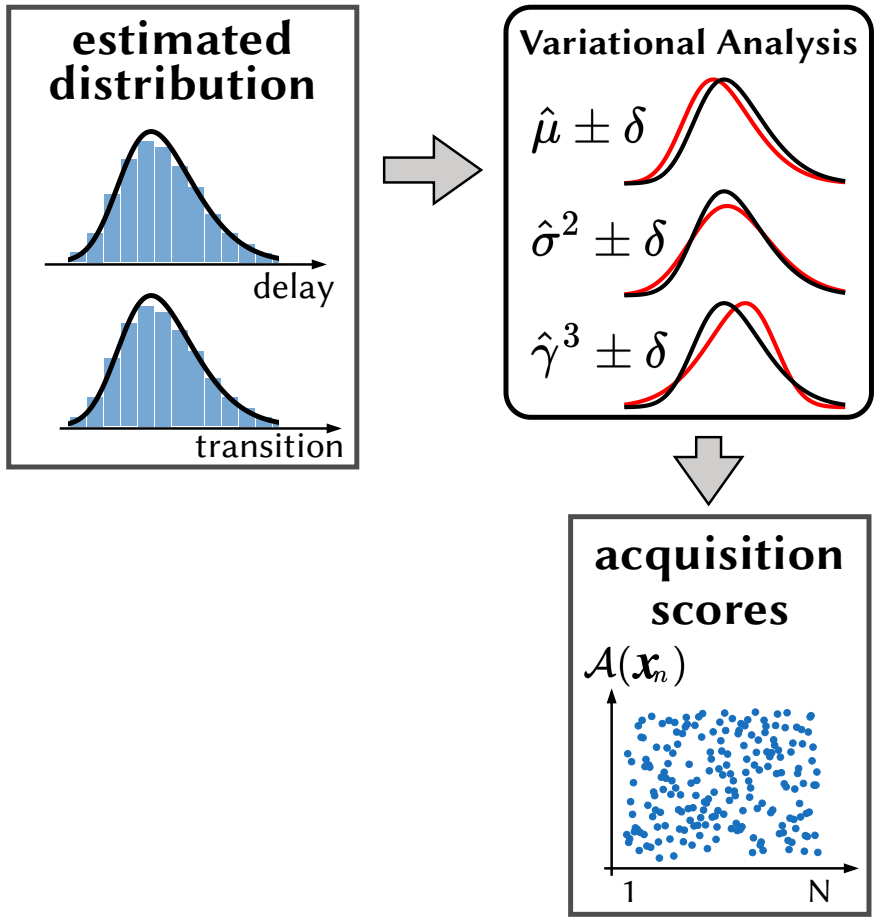
Evaluate Acquisition Score



- Goal: minimize the distance from estimation to real
- Acquisition Score $\mathcal{A}(x_n)$: **sample's contribution for estimated distribution's uncertainty**
- Propagate from timing uncertainty to $\mathcal{A}(x_n)$
  - Timing uncertainty → Each moment uncertainty
  - Moment uncertainty → Distribution uncertainty
    Add perturbation (variational analysis)

$$\nabla_{\hat{\sigma}^2}\mathcal{L} = \lim_{\delta\to 0}\frac{1}{\delta}\left(D[LVF(\hat{\mu},\hat{\sigma}^2+\delta,\hat{\gamma}^3)\mid LVF(\mu,\sigma^2,\gamma^3)]\right.$$
$$\left.-D[LVF(\hat{\mu},\hat{\sigma}^2,\hat{\gamma}^3)\mid LVF(\mu,\sigma^2,\gamma^3)]\right)$$
$$\approx \lim_{\delta\to 0}\frac{1}{\delta}D[LVF(\hat{\mu},\hat{\sigma}^2+\delta,\hat{\gamma}^3)\mid LVF(\hat{\mu},\hat{\sigma}^2,\hat{\gamma}^3)]$$

Evaluate Acquisition Score

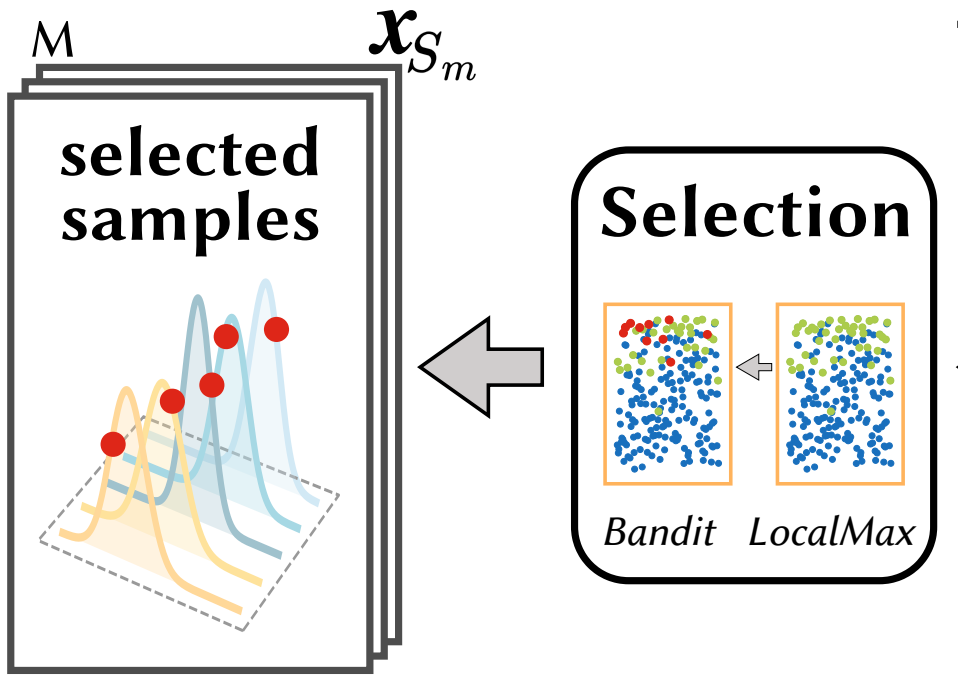


- Goal: minimize the distance from estimation to real
- Acquisition Score $\mathcal{A}(x_n)$: **sample's contribution for estimated distribution's uncertainty**
- Propagate from timing uncertainty to $\mathcal{A}(x_n)$
  - Timing uncertainty → Each moment uncertainty
  - Moment uncertainty → Distribution uncertainty

$$\mathcal{A}(x_n) = \left(\nabla_{\hat{\mu}}\mathcal{L}\right)^2 \cdot \nabla_{v(x_n)} Var[\hat{\mu}] \cdot v(x_n) + \left(\nabla_{\hat{\sigma}^2}\mathcal{L}\right)^2 \cdot \nabla_{v(x_n)} Var[\hat{\sigma}^2] \cdot v(x_n) + \left(\nabla_{\hat{\gamma}^3}\mathcal{L}\right)^2 \cdot \nabla_{v(x_n)} Var[\hat{\gamma}^3] \cdot v(x_n)$$

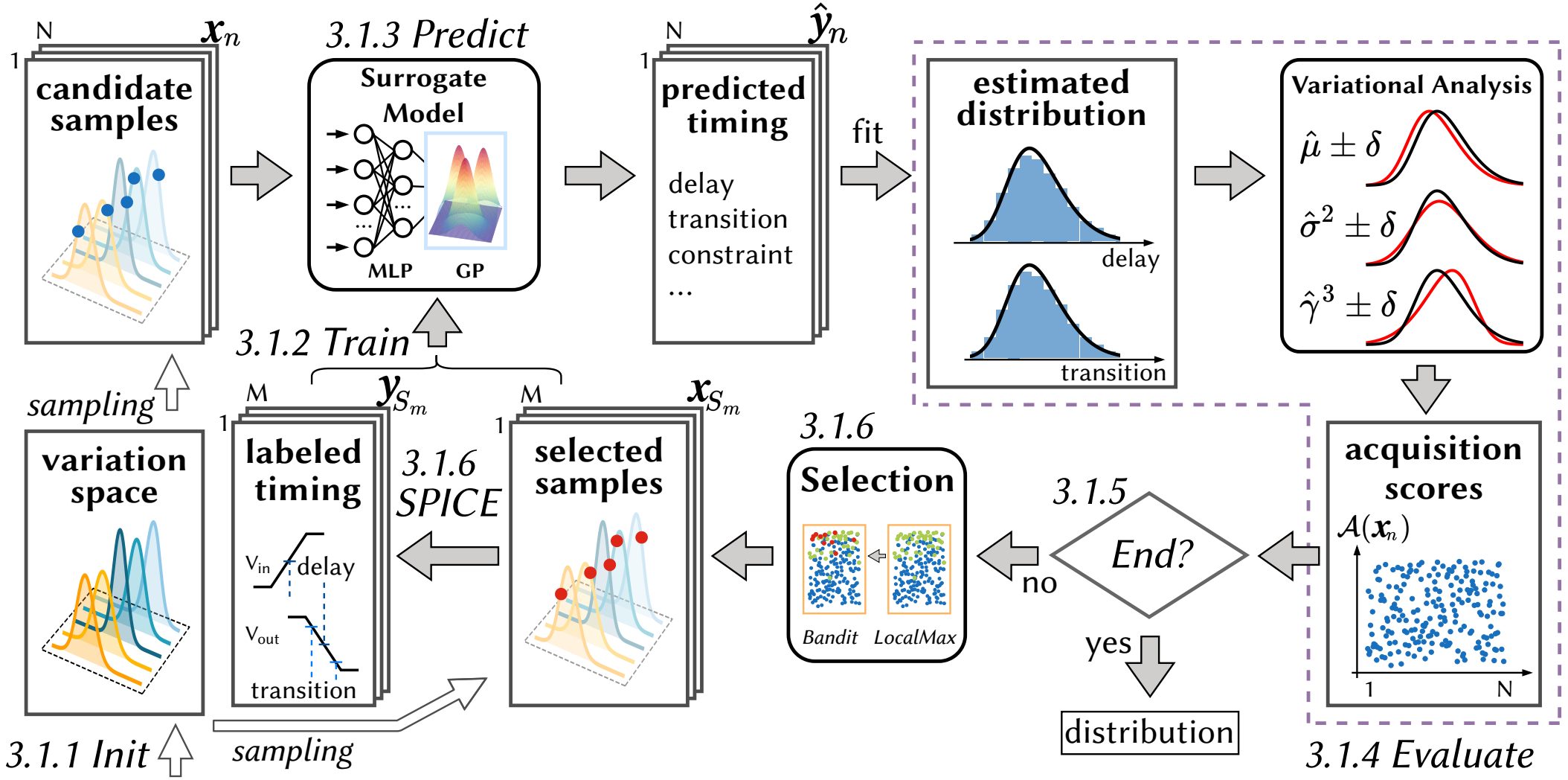
- **Multi-timing co-optimization**
  Weighting each timing's (delay/transition/setup/...) acquisition score together

Batch Selection



- Select samples with batch size
- To avoid all selections come from same maximal area
  - Local Max, in multi-dimension space
  - Bandit, the possibility to be selected ↔ its value
- To next iteration …

Recap LVFGen



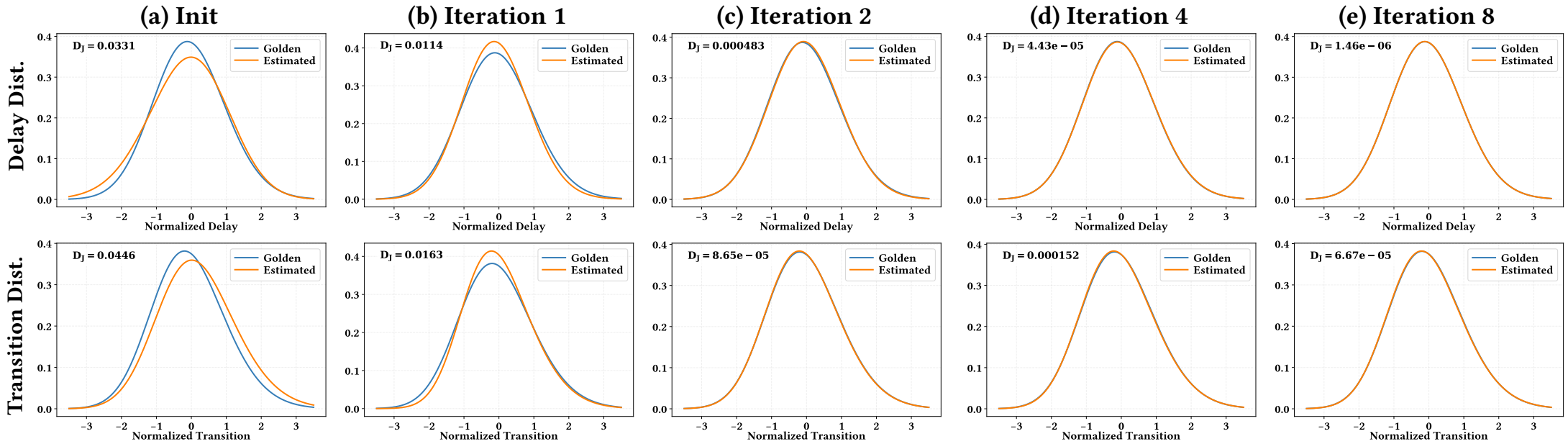
Experiment

- Setup
  - TSMC22nm, TTGlobal, 0.8V, 25 $^{\circ}C$
  - Golden: the result of random MC with 100k samples
  - Jensen–Shannon divergence to measure the distributions' distance

- Methods to compare
  - Random
  - Sobol's QMC
  - LVFGen

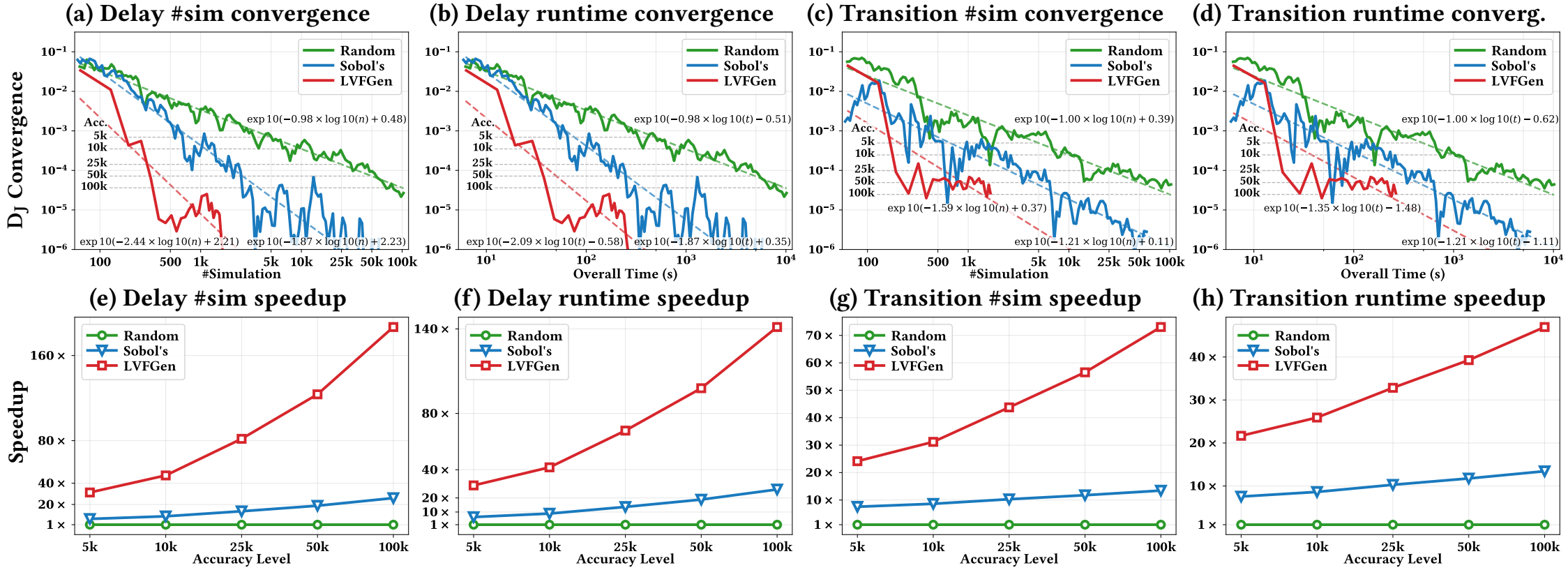
[TCAD'10: Why Quasi-Monte Carlo is Better Than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis]

Comparison of PDF for OR2 Cell



- 48-dimensional OR2X2 cell, 32 samples for each iteration (batch size)
- Delay/Transition time co-optimization
- The estimation distribution rapidly converge to golden
- The J-divergence also reduces rapidly

Comparison of Convergence for OR2 Cell



- Use J-divergence to quantificationally compare LVFGen with Random and Sobol's
- Better convergence & speedup

Speedup for Cell Library

Table 1: Speedup Comparison Using Standard Cell Library.

Cell Type*	#Transistor	Origin dim.	Pruned dim.	Delay Runtime Speedup† (×)								Transition Runtime Speedup† (×)							
				5k Accuracy		10k Accuracy		50k Accuracy		100k Accuracy		5k Accuracy		10k Accuracy		50k Accuracy		100k Accuracy	
				Sobol's	Ours	Sobol's	Ours	Sobol's	Ours	Sobol's	Ours	Sobol's	Ours	Sobol's	Ours	Sobol's	Ours	Sobol's	Ours
OR2X1	6	36	12	9.23	**34.99**	10.95	**46.00**	15.91	**85.02**	18.54	**110.71**	4.36	**10.62**	4.60	**11.51**	5.30	**12.96**	5.68	**13.30**
OR2X2	8	48	12	8.45	**26.80**	10.41	**42.11**	15.89	**118.03**	18.67	**182.37**	4.99	**10.29**	6.25	**13.57**	10.15	**25.28**	12.40	**32.24**
NOR2X1	4	24	10	8.99	**56.35**	9.95	**74.81**	12.48	**142.98**	13.75	**187.76**	3.31	**5.45**	3.66	**5.08**	**4.48**	3.76	**4.86**	3.13
NOR2X2	8	48	10	3.56	**9.30**	3.72	**12.66**	4.07	**25.63**	4.21	**34.99**	**2.15**	1.37	**2.34**	1.42	**2.85**	1.51	**3.10**	1.54
AND2X1	6	36	10	26.02	**63.02**	33.54	**90.60**	60.51	**207.56**	78.24	**294.38**	10.88	**11.30**	**11.88**	11.23	**14.51**	10.10	**15.68**	9.34
AND2X2	8	48	10	17.59	**48.12**	22.22	**73.23**	38.08	**196.07**	47.80	**298.94**	7.13	**16.84**	7.97	**22.15**	10.04	**39.51**	10.97	**51.11**
NAND2X1	4	24	12	82.08	**247.67**	114.61	**365.67**	254.07	**876.82**	360.19	**1262.33**	**19.07**	11.11	**25.65**	10.49	**49.33**	9.37	**64.52**	8.34
NAND2X2	8	48	12	8.86	**20.58**	10.20	**28.64**	14.11	**62.54**	16.22	**87.48**	**4.89**	1.92	**5.57**	2.00	**7.46**	2.16	**8.43**	2.23
AOI21X1	6	36	16	62.72	**156.12**	95.28	**258.62**	250.36	**825.74**	378.69	**1354.56**	20.16	**34.11**	28.00	**48.22**	59.59	**102.51**	82.32	**138.16**
OAI21X1	6	36	16	37.83	**106.35**	48.08	**157.71**	83.02	**396.32**	104.60	**584.31**	**11.79**	3.99	**13.78**	3.19	**19.79**	1.70	**23.05**	1.26
Low-dim. Avg.				26.53	**76.93**	35.90	**115.01**	74.85	**293.67**	104.09	**439.78**	8.87	**10.70**	10.97	**12.89**	18.35	**20.89**	23.10	**26.06**
XNOR2X1	12	72	16	30.13	**34.35**	34.25	**53.61**	44.88	**150.67**	50.22	**234.26**	12.61	**26.34**	14.19	**40.22**	18.14	**106.02**	19.87	**160.88**
XNOR2X2	14	84	16	5.99	**18.60**	6.19	**27.24**	6.50	**65.76**	6.58	**95.91**	2.89	**14.65**	3.21	**21.32**	4.07	**50.70**	4.53	**73.46**
XOR2X1	12	72	16	8.47	**25.85**	9.98	**38.81**	14.16	**98.52**	16.40	**146.28**	12.34	**25.30**	13.55	**37.96**	17.09	**97.25**	18.34	**145.01**
XOR2X2	14	84	16	22.37	**31.83**	29.92	**50.72**	58.24	**150.20**	77.72	**239.00**	8.24	**22.39**	10.58	**34.32**	18.57	**91.10**	23.38	**137.97**
AOI21X2	12	72	16	11.06	**20.28**	13.57	**31.52**	21.56	**87.20**	26.17	**134.96**	**4.17**	3.97	**4.68**	4.37	**6.10**	4.93	**6.80**	5.02
OAI21X2	12	72	16	4.13	**20.39**	4.56	**30.40**	5.70	**76.58**	6.26	**113.99**	1.43	**5.29**	1.47	**6.68**	1.57	**11.28**	1.61	**14.02**
Middle-dim. Avg.				13.69	**25.22**	16.41	**38.72**	25.17	**104.82**	30.56	**160.73**	6.95	**16.32**	7.95	**24.15**	10.92	**60.21**	12.42	**89.39**
OR2X4	16	96	12	16.92	**19.25**	20.48	**29.95**	31.80	**82.82**	38.37	**128.05**	9.77	**10.84**	12.40	**15.66**	21.27	**36.51**	26.55	**52.39**
NOR2X4	16	96	10	8.38	**13.36**	9.75	**19.88**	13.87	**49.44**	16.13	**72.82**	4.16	**5.26**	4.79	**6.87**	6.51	**12.57**	7.38	**16.18**
AND2X4	16	96	10	16.66	**16.93**	22.55	**27.65**	44.73	**85.25**	59.73	**137.77**	**9.53**	8.43	**12.80**	12.54	24.54	**30.93**	32.11	**45.27**
NAND2X4	16	96	12	**2.50**	1.96	**2.51**	2.30	2.53	**3.21**	2.54	**3.65**	**1.50**	1.32	**1.44**	1.34	1.31	**1.36**	1.26	**1.36**
HA1X1	20	120	16	**15.28**	11.38	**19.48**	17.27	32.88	**45.37**	41.04	**68.49**	**9.10**	4.82	**10.36**	6.64	**14.10**	13.72	16.24	**18.60**
HA1X2	24	144	16	**10.12**	6.02	**11.53**	9.02	15.96	**22.82**	18.52	**33.95**	**3.63**	2.82	**4.20**	4.02	6.11	**9.07**	7.18	**12.85**
XNOR2X4	26	156	16	**5.14**	2.07	**5.85**	2.86	**7.77**	6.03	**8.73**	**8.29**	4.14	1.41	**4.80**	1.87	**6.73**	3.59	**7.77**	4.73
XOR2X4	26	156	16	**7.48**	1.89	**8.62**	2.57	**11.13**	5.17	**12.10**	6.93	**4.57**	1.96	**4.98**	2.66	**6.07**	5.29	6.60	**7.05**
AOI21X4	24	144	16	**6.14**	3.60	**7.97**	5.38	**14.69**	13.69	19.18	**20.45**	**2.51**	2.10	**3.30**	3.00	6.24	**6.81**	8.20	**9.67**
OAI21X4	24	144	16	3.88	**4.50**	4.19	**6.53**	4.93	**15.51**	5.27	**22.51**	1.56	**1.83**	1.54	**2.43**	1.47	**4.66**	1.43	**6.13**
High-dim. Avg.				9.25	**8.10**	11.29	**12.34**	18.03	**32.93**	22.16	**50.29**	**5.05**	4.08	**6.06**	5.70	9.44	**12.45**	11.47	**17.42**
Overall				16.92	**38.52**	21.94	**57.91**	41.53	**149.81**	55.61	**225.58**	6.96	**9.45**	8.38	**12.72**	13.21	**26.72**	16.16	**37.35**
Speedup Compare to Sobol's				1×	**2.28×**	1×	**2.64×**	1×	**3.61×**	1×	**4.06×**	1×	**1.36×**	1×	**1.52×**	1×	**2.02×**	1×	**2.31×**

* One timing arc test for each cell type, containing 8×8 delay and transition distributions; † Runtime speedup compared to Random MC.

- Validation on 26 cells
- 8×8 slew-load pairs
- Evaluate the speedup on accuracy level from 5k to 100k
- LVFGen achieves up to **4×** and **2.3×** speedup, compare to Sobol's

Application to Timing Analysis

Table 2: Accuracy Comparison for ISCAS'89 circuits, Golden Uses 100k Accuracy for Cells.

Benchmark	#Gates	Mean for Critical Delay (ns)			Std. Dev. for Critical Delay (ns)		
		MC 100k	**Sobol's** 5k	**LVFGen** 0.7k	**MC** 100k	**Sobol's** 5k	**LVFGen** 0.7k
s27	15	13.031190(0‰)	13.031190 (0‰)	13.031192 (1.53E-04‰)	0.000956(0‰)	0.000955 (1.05‰)	0.000956 (0‰)
s298	106	0.086129(0‰)	0.086129 (0‰)	0.086129 (0‰)	0.007545(0‰)	0.007545 (0‰)	0.007545 (0‰)
s641	129	13.197377(0‰)	13.197379 (1.52E-04‰)	13.197384 (5.30E-04‰)	0.003137(0‰)	0.003134 (9.56E-01‰)	0.003139 (6.38E-01‰)
s1196	519	13.194728(0‰)	13.194734 (4.55E-04‰)	13.194733 (3.79E-04‰)	0.002978(0‰)	0.002976 (6.72E-01‰)	0.002979 (3.36E-01‰)
s15850	589	13.025476(0‰)	13.025476 (0‰)	13.025476 (0‰)	0.001611(0‰)	0.001611 (0‰)	0.001613 (1.24‰)
s9234_1	1045	13.053585(0‰)	13.053590 (3.83E-04‰)	13.053593 (6.13E-04‰)	0.001939(0‰)	0.001938 (5.16E-01‰)	0.001940 (5.16E-01‰)
s13207	1067	13.024756(0‰)	13.024757 (7.68E-05‰)	13.024757 (7.68E-05‰)	0.001007(0‰)	0.001006 (9.93E-01‰)	0.001010 (2.98‰)
s5378	1524	13.132874(0‰)	13.132888 (1.07E-03‰)	13.132887 (9.90E-04‰)	0.003658(0‰)	0.003656 (5.47E-01‰)	0.003664 (1.64‰)
MRAE*		-	1× , 2.67E-4‰	1.29× , 3.43E-4‰	-	1× , 0.59‰	1.55× , 0.92‰

* **Mean Relative Absolute Error.**

- PrimeTime SSTA on ISCAS'89 circuits
- 3 libraries with equivalent 100k MC's accuracy
- Library generation runtime
  - 5k Sobol's: 290 hr.
  - 0.7k LVFGen: 80 hr. (3.5× speedup)
- There is almost no accuracy loss

Conclusions and Future Work

- LVFGen: Efficient Cells Library Generation
  - 3.5× faster, 290 hr. to 80 hr.
  - Almost no accuracy loss

- Future Work
  - Setup/Hold specified algorithm
  - Cross PVT corners, reduce overhead
  - Extend to LVF^2

[DAC'24: A Statistical Timing Model based on Gaussian Mixture for Yield Estimation and Speed Binning]

Thank You

junzhuo22@ucla.edu