Building up End-to-end Mask Optimization Framework with Self-training

Bentian Jiang, Xiaopeng Zhang, Lixin Liu, Evangeline F.Y. Young

CSE Dept., The Chinese University of Hong Kong

March 23, 2021

香港中文大學
The Chinese University of Hong Kong

Outline

Overview

Algorithm

Empirical Evaluations

Conclusion

Outline

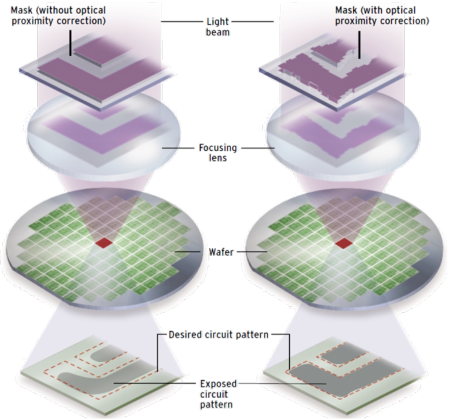
Mask Optimization

Lithography

- Use light to transfer patterns from photomask to light sensitive photoresist on wafer
- Mismatch between lithography system and device feature sizes

Optical proximity correction (OPC)

- OPC compensates the printing errors by modifying the masks
- Guided by lithography simulation models



OPC illustration (sources from F. Schellenberg [4]).

Inverse Lithography Technique (ILT)

A pixel-wise numerical OPC approach for mask optimization

- Describe forward lithography under nominal process as a function $f(\cdot, \mathbf{R}_{\mathsf{nom}})$

$$\mathbf{Z} = f(\mathbf{M};\ \mathbf{R}_{\mathsf{nom}}),\quad \mathbf{Z} : \text{printed shape},\ \mathbf{M} : \text{mask}.$$

- Primary objective: minimize printing error in an pixel-wise manner

$$Loss_{\text{ilt}} = \sum_{x=1}^{N}\sum_{y=1}^{N}(\mathbf{Z}(x,y) - \mathbf{Z}_{\mathsf{t}}(x,y))^2,\quad \mathbf{Z}_{\mathsf{t}} : \text{desired shape}.$$

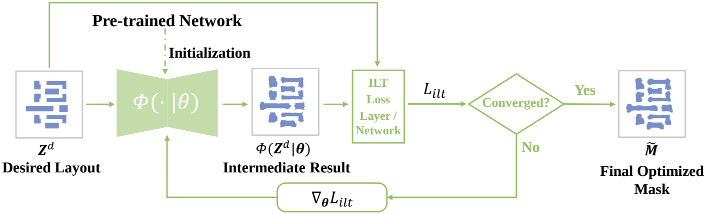
Properties

- Numerical: modify mask iteratively by gradient descent.
- Good QoR but extremely computational intensive.

Learning-based ILT (DNN-ILT)

On-neural-network ILT Correction (DNN-ILT)

- End-to-end solution, purely learning-based approaches.
- Supervised network pre-training + unsupervised mask optimization.
- Domain-specific model, faster convergence, better QoR.



DNN-ILT, a typical end-to-end on-neural-network ILT correction framework [3].

Challenges & Motivations

Challenges for learning-based OPC in public research

- Lack of realistic designs in advanced tech nodes.
- Lack of accurate lithography simulation recipe.
- Labelling costs with conventional ILT is drastically expensive.

Our Goals

- Build end-to-end learning-based OPC (ILT) tool
- Handle training data shortage
- Reduce labeling costs
- Modular design for easy upgrade

Outline

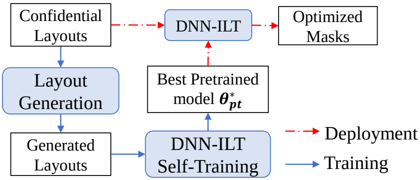
Overview of Framework

Problem Abstraction

- Construct a DNN-ILT platform
- Not involve confidential layout in training
- Deploy on confidential layouts

Two-stages Framework

- Layout generation stage
- DNN-ILT self-training stage



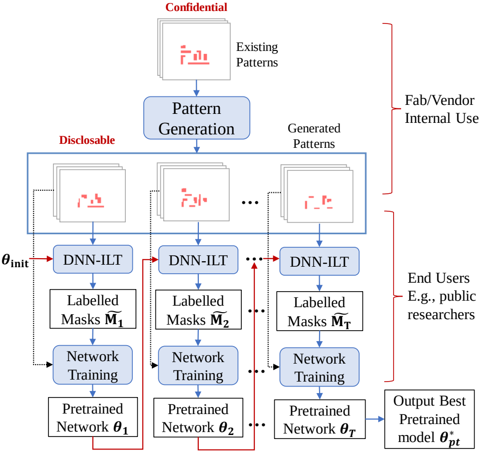
Overview of Framework - Cont'

Layout Generation Stage

- Fast: learning-based approach
- Realistic: style preserved, DRC free
- Secure: hard to revert

DNN-ILT Self-training Stage

- Fast labelling with DNN-ILT
- Self-contained, no input label required
- Domain-specific training



Overview of the proposed self-training pipeline, θ^*_{pt} is capable of conducting DNN-ILT for confidential layouts.

Layout Generation Stage - Base

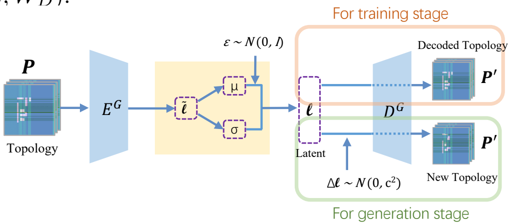
Training stage objectives

$$\min \quad \underbrace{D_{KL}(\mathcal{N}(\mu,\sigma^2)\|\mathcal{N}(0,I))}_{\text{KL Divergence}} + \underbrace{\lambda\|\mathbf{P}-\mathbf{P}'\|_F^2}_{\text{Self reconstruction}}$$

$$\begin{aligned} \text{s.t.} \quad & \mu = f_R^{\mu}(\tilde{l}, W_R^{\mu}), \quad \sigma = f_R^{\sigma}(\tilde{l}, W_R^{\sigma}) \\ & \tilde{l} = E^G(\mathbf{P}, W_E), \\ & l = \mu + \sigma \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \\ & \mathbf{P}' = D^G(l, W_D). \end{aligned}$$

Generation stage objectives

$$\begin{aligned} & \mathbf{P}' = D^G(l + \triangle l, W_D), \\ & \triangle l_i \sim \mathcal{N}(0, c^2), \quad i = 1, ..., K, \end{aligned}$$

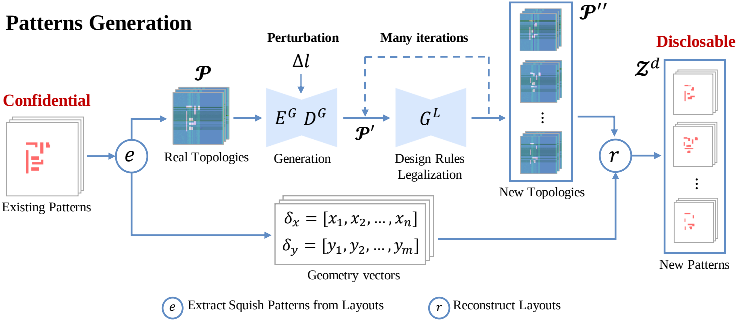


Layout topology generation based on variational convolutional auto-encoder (VCAE) architecture [6].

Layout Generation Stage - Cont'

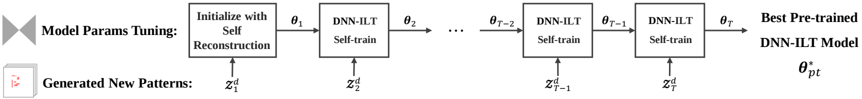
Full view of pattern generation

- VCAE & style detector [6] to preserve original layout style
- Legalization to ensure DRC free
- Nonlinearities in decoder and legalizer, hard to revert original layouts



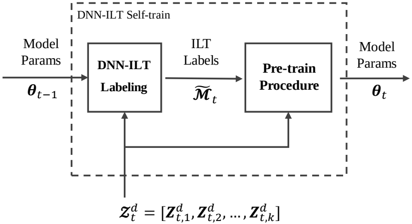
General Pipeline for Self-Training

- Generated patterns are spilt into T batches, $\mathcal{Z}^{\mathrm{d}} = \bigcup_{i=1}^{T} \mathcal{Z}_i^{\mathrm{d}}$
- The i^{th} batch $\mathcal{Z}_i^{\mathrm{d}} \to$ self-training of model θ_i



Basic unit of self-training

- Input: old model θ_{t-1}
  Output: new model θ_t
- DNN-ILT to label $\mathcal{Z}_t^{\mathrm{d}} \to \tilde{\mathcal{M}}_t$
- $(\mathcal{Z}_t^{\mathrm{d}}, \tilde{\mathcal{M}}_t)$ to pre-train θ_t



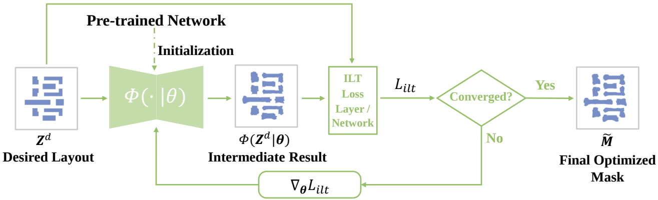
DNN-ILT - Cont'

- Label prediction $\Phi(\mathbf{Z}_t^{\mathsf{d}}|\theta_{t-1}) \rightarrow$ gradient descent $(\nabla_{\theta_{t-1}} L_{\text{dnn\_ilt}})$
- Optimize printability (ILT) + process variation (PVBand)

$$\begin{aligned} L_{\text{dnn\_ilt}} =& \alpha \cdot L_{\text{ilt}} + \beta \cdot L_{\text{pvband}} \\ =& \alpha \cdot ||f(\Phi(\mathbf{Z}_t^{\mathsf{d}}|\theta_{t-1}) \mid \mathbf{R}_{\mathsf{nom}}) - \mathbf{Z}_t^{\mathsf{d}}||_2^2 + \\ & \beta \cdot ||f(\Phi(\mathbf{Z}_t^{\mathsf{d}}|\theta_{t-1}) \mid \mathbf{R}_{\mathsf{min}}) - f(\Phi(\mathbf{Z}_t^{\mathsf{d}}|\theta_{t-1}) \mid \mathbf{R}_{\mathsf{max}})||_2^2, \end{aligned}$$

$$\nabla_\theta L_{\text{dnn\_ilt}} = \frac{\partial L_{\mathsf{dnn\_ilt}}}{\partial \theta} = \frac{\partial L_{\mathsf{dnn\_ilt}}}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \bar{\mathbf{M}}} \frac{\partial \bar{\mathbf{M}}}{\partial \Phi(\mathbf{Z}_{\mathsf{t}};\theta)} \frac{\partial \Phi(\mathbf{Z}_{\mathsf{t}};\theta)}{\partial \theta},$$

details in [3].



Pre-train Next Network

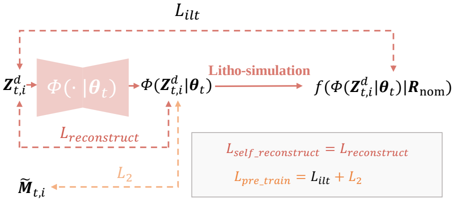
DNN-ILT $\rightarrow$ Initial solution sensitive, better θ_{pretrain} is indispensable

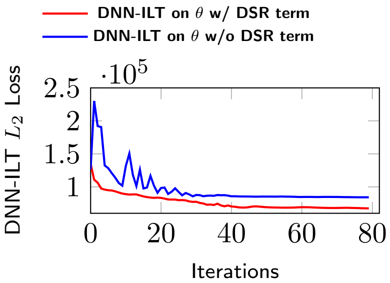
No prior knowledge to pre-train θ_1, let it reconstruct the input itself

$$L_{\text{self\_reconstruct}} = ||\Phi(\mathbf{Z}_1^{\text{d}}|\theta_1) - \mathbf{Z}_1^{\text{d}}||_2^2.$$

Special training recipe with domain specific regularization (DSR) term

$$L_{\text{pre\_train}} = \overbrace{||\Phi(\mathbf{Z}_t^{\text{d}}|\theta_t) - \tilde{\mathcal{M}}_t^{\text{d}}||_2^2}^{\text{supervised term}} + \overbrace{\eta||f(\Phi(\mathbf{Z}_t^{\text{d}}|\theta_t) \mid \mathbf{R}_{\text{nom}}) - \mathbf{Z}_t^{\text{d}}||_2^2}^{\text{domain-specific regularization (DSR) term}}, \quad t > 1$$





What's New?

	Conventional Self-training	Ours
Framework	Semi-supervised learning	Semi-supervised learning
Label Generation	Predict pseudo labels	Optimize exact labels (DNN-ILT)
Use Pre-label Data?	Yes	No, self-contained
Learn from Where?	Learn from pre-labelled data	Learn from scratch via DNN-ILT

Outline

Overview

Algorithm

Empirical Evaluations

Conclusion

Evaluation Settings

Dataset

- Original/confidential dataset $\mathcal{Z}^{\mathsf{u}}$ - 4000 instances [5], $2\text{x}2\mu m^2$, 32nm M1
- Generated dataset $\mathcal{Z}^{\mathsf{d}}$ - 4 new topologies for each $\mathcal{Z}^{\mathsf{u}}$ sample
- Test dataset - ICCAD 2013 mask optimization benchmark suite [1]

Baselines

	Conventional ILT [2]	Neural-ILT [3]	Ours
Platform	CPU	GPU	GPU
Training Dataset	N/A	Original dataset	Generated dataset
Labelling Tool	N/A	Conventional ILT [2]	Neural-ILT [3] (DNN-ILT)
Construction Method	N/A	Supervised learning	Self-training

Performance Comparisons with Baselines

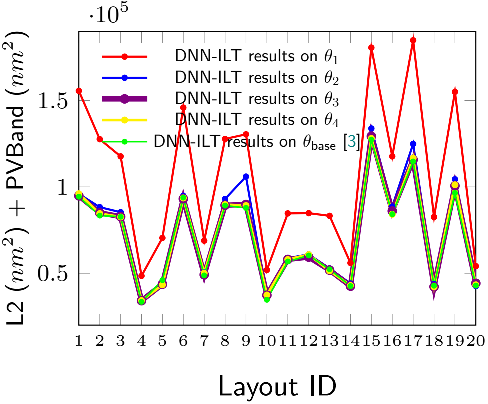
Benchmarks	Conventional ILT [2]			Neural-ILT [3]			Ours		
ID	Runtime (s)	L_2 (nm^2)	PVB (nm^2)	Runtime (s)	L_2 (nm^2)	PVB (nm^2)	Runtime (s)	L_2 (nm^2)	PVB (nm^2)
1	1280	49893	65534	13.57	50795	63695	14.92	52583	62753
2	381	50369	48230	14.37	36969	60232	13.98	41471	56559
3	1123	81007	108608	9.72	94447	85358	15.04	82360	111507
4	1271	20044	28285	10.40	17420	32287	21.42	17597	32607
5	1120	44656	58835	10.04	42337	65536	21.48	39405	64828
6	391	57375	48739	11.11	39601	59247	11.50	41535	56210
7	406	37221	43490	9.67	25424	50109	17.26	25884	50956
8	388	19782	22846	11.81	15588	25826	21.60	16562	26016
9	1138	55399	66331	9.68	52304	68650	12.25	53319	67376
10	387	24381	18097	11.46	10153	22443	21.59	12199	21790
Average	788.5	44012.7	50899.5	11.18	38504	53338	17.10	38292	55060
Ratio	1.00	1.00	1.00	0.014	0.875	1.048	0.022	0.870	1.082
Flow TAT*	N/A			$>$ 10 days			$<$ **3 days**		

- ICCAD 2013 benchmarks, comparable printability score (L2 + PVB)
- Comparing to conventional ILT, 50x ILT runtime speedup
- Comparing to Neural-ILT, 3x flow construction cost reduction

*Flow TAT: total time of flow construction: data labelling + model training

Performance in Original Dataset

- 20 unseen layouts from original (assume confidential) dataset
- $\theta_1 \sim \theta_4$ trained on generated dataset
- θ_{base} (Neural-ILT [3]) trained on original dataset
- $\theta_1 \rightarrow \theta_4$ converge to the performance of θ_{base} rapidly



Outline

Conclusion

Unified self-training framework to construct end-to-end learning-based mask optimization tool

Highlights

- Layout generation - massive, fast, realistic, secure
- DNN-ILT labelling - extremely fast labelling overhead
- Self training - self-contained, no pre-labelled data required

Modular Design

- Re-configure litho-simulation recipe
- Upgrade backbone model
- Customize objective functions for mask optimization

Q&A

Thanks and Questions?

Banerjee, S., Li, Z., and Nassif, S. R.
ICCAD-2013 CAD contest in mask optimization and benchmark suite.
In *Proc. ICCAD* (2013), pp. 271–274.

Gao, J.-R., Xu, X., Yu, B., and Pan, D. Z.
MOSAIC: Mask optimizing solution with process window aware inverse correction.
In *Proc. DAC* (2014), pp. 52:1–52:6.

Jiang, B., Liu, L., Ma, Y., Zhang, H., Yu, B., and Young, E. F. Y.
Neural-ilt: Migrating ilt to neural networks for mask printability and complexity co-optimization.
In *Proceedings of the 39th International Conference on Computer-Aided Design* (2020).

Schellenberg, F.
A little light magic [optical lithography].
IEEE Spectrum 40, 9 (2003), 34–39.

Yang, H., Li, S., Deng, Z., Ma, Y., Yu, B., and Young, E. F. Y.
GAN-OPC: Mask optimization with lithography-guided generative adversarial nets.
IEEE TCAD (2020).

Zhang, X., Shiely, J., and Young, E. F. Y.
Layout pattern generation and legalization with generative learning models.
In *Proceedings of the 39th International Conference on Computer-Aided Design* (2020).