Outline

- Motivation
- Introduction to neural network model
- Knowledge-based NN model
  - Training with inductive bias
  - Building prior knowledge from multiple tasks
- End-to-end autonomous technology development

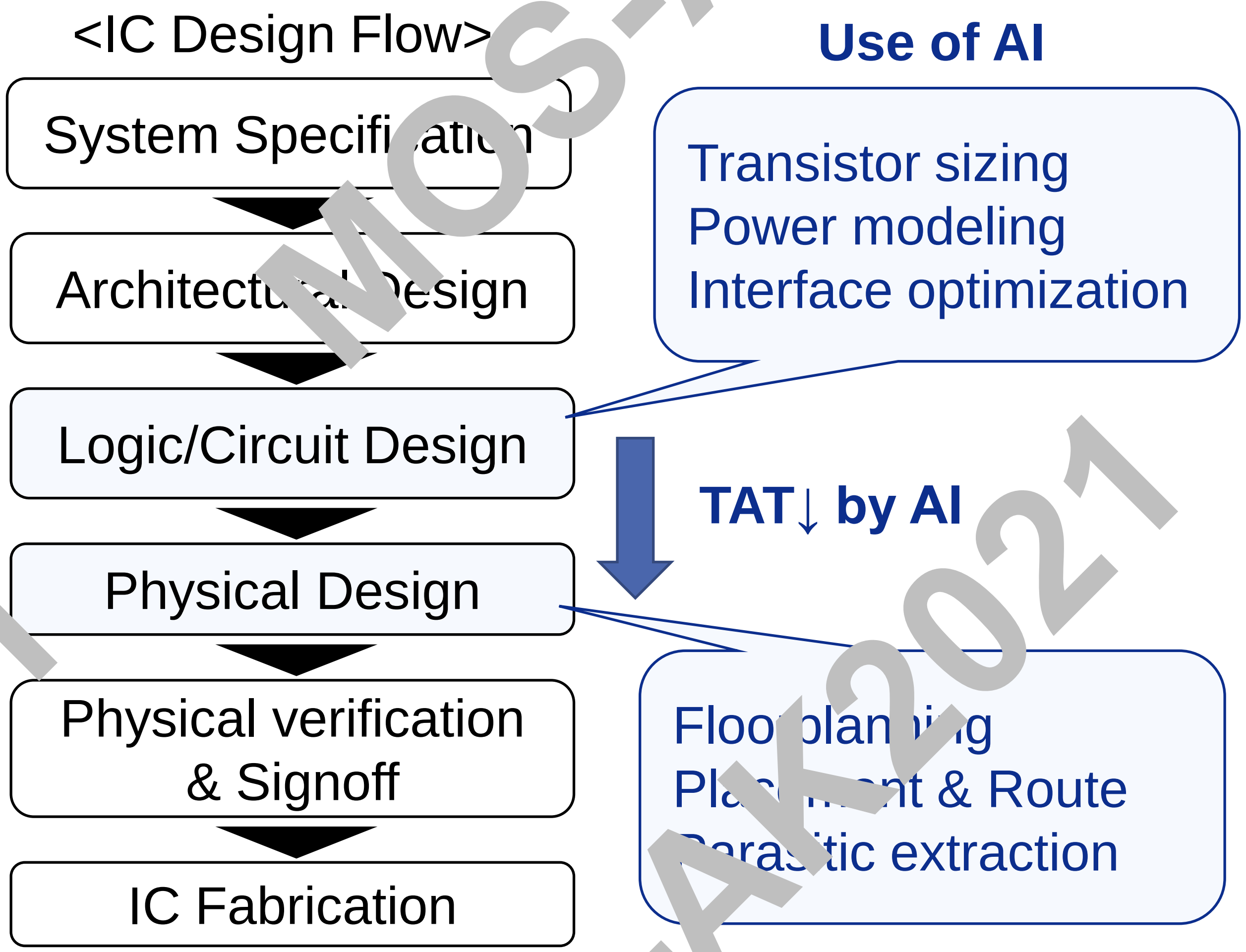
Goal:
1. Provide understanding of neural network device modeling
2. Share possible directions to make NN device modeling more intelligent

Outline

- Motivation
- Introduction to neural network model
- Knowledge-based NN model
  - Training with inductive bias
  - Building prior knowledge from multiple tasks
- End-to-end autonomous technology development

AI in IC Design

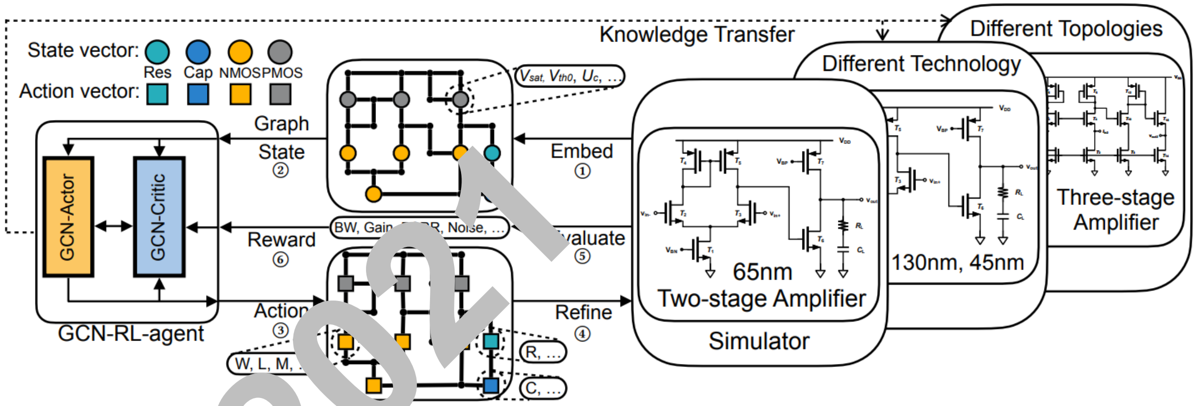
- Active research in hardware design optimization with AI



AI in IC Design - Examples

[Transistor Sizing]

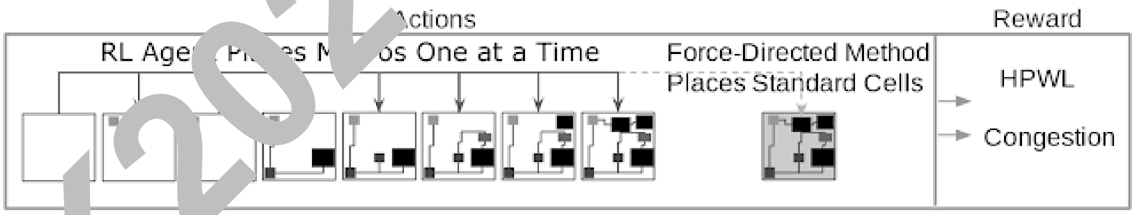
- Transferable transistor sizing with AI
  - Automated transistor sizing based on circuit performance results
  - Transfer knowledge of one technology/topology to another technology/topology



[H. Wang et al., "GCN-RL circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," DAC, 2020]

[Floorplanning]

- Next generation Google AI accelerator designed using AI
  - "In under six hours, our method automatically generates chip floorplans that are superior or comparable to those produced by humans in all key metrics, including power consumption, performance, and chip area."



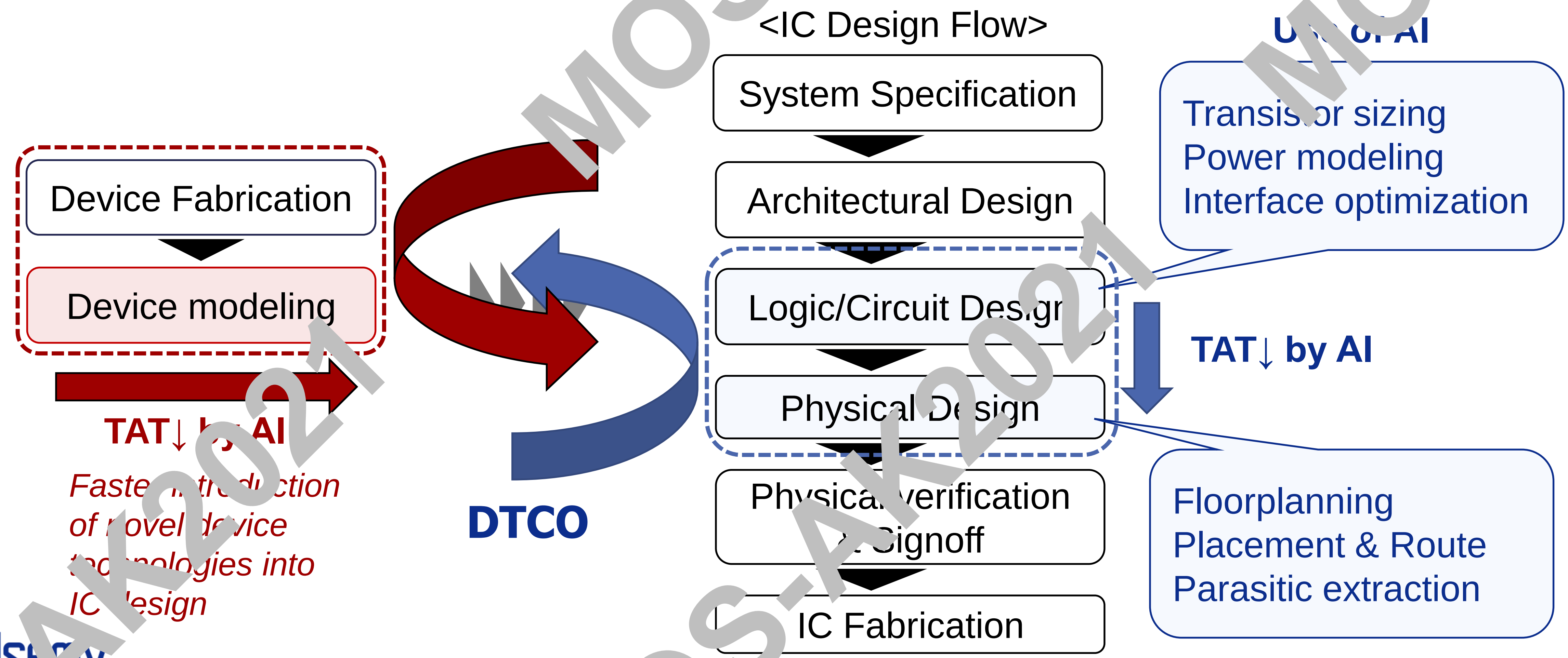
[https://ai.googleblog.com/2020/04/chip-design-with-deep-reinforcement.html]

[A. Mirhoseini et al., "A graph placement methodology for fast chip design," Nature, 2021]

AI in Device Modeling

- Limited research in device modeling using AI
- DTCO facilitated by AI in device modeling

    Design Technology Co-optimization

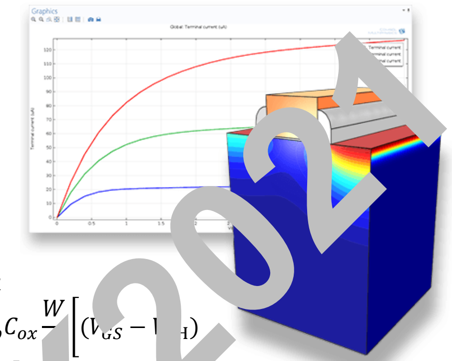


Conventional device modeling: Analytical model

- Physics-based analytical model: Long development time, limited accuracy

Device analysis & Analytical model development

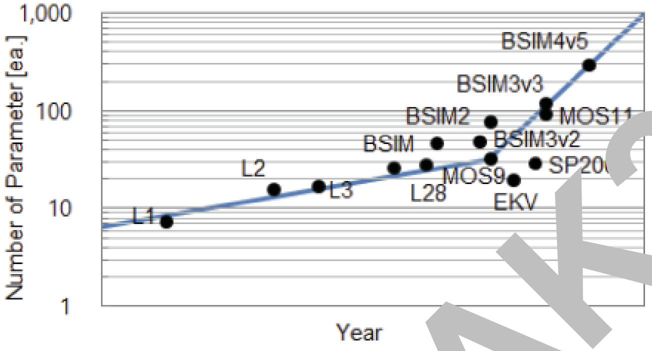
- Development time: years
- Include known physics + fitting parameters
- Need semiconductor experts



$$I_{D,tri} = \mu_p C_{ox} \frac{W}{L}\left[(V_{GS} - V_{TH}) - \frac{V_{DS}}{2}\right] V_{DS}\left[1 - \lambda(V_{DS} - V_{D,sat})\right]$$

Parameter extraction

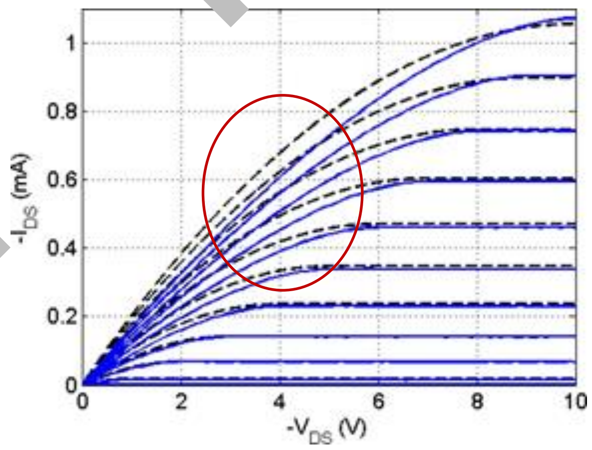
- Development time: months – reduced to hours by using AI [Primarus SDEP]
- Hundreds of parameters to fit



[Y. Kim *et al.*, "The efficient DTCO compact Modeling Solutions to Improve MHC and Reduce TAT," SISPAD 2018]

Analytical model

- Inaccuracy from unknown physics

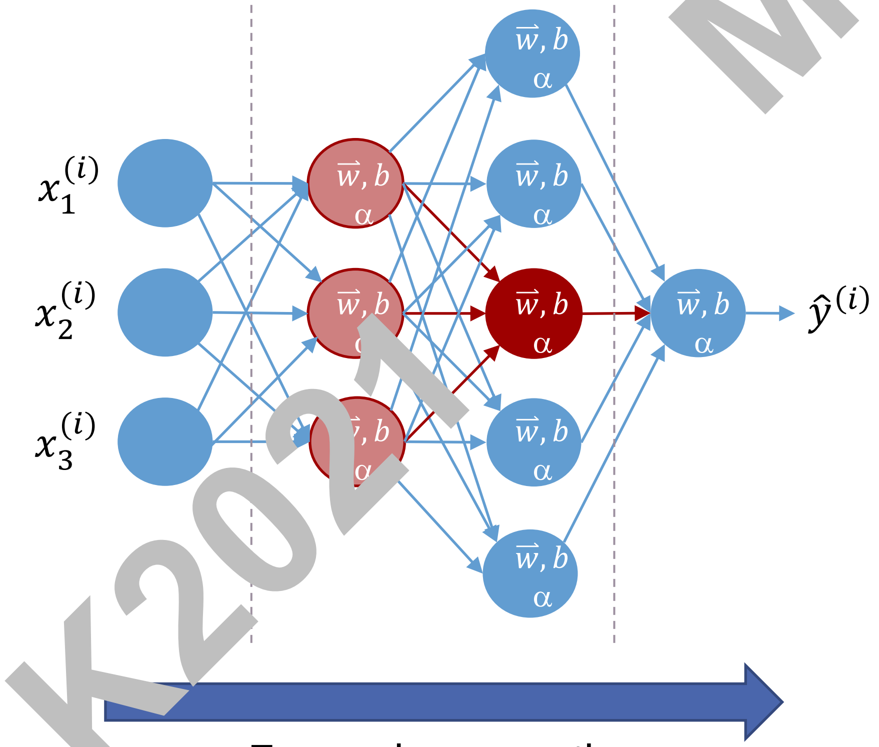


Outline

- Motivation
- Introduction to neural network model
- Knowledge-based NN model
  - Training with inductive bias
  - Building prior knowledge from multiple tasks
- End-to-end autonomous technology development

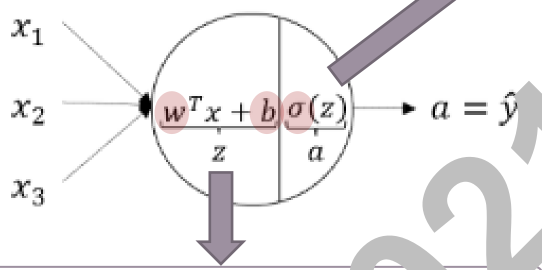
Neural network basics – Forward propagation

- Data set: $\left(\begin{pmatrix}x_1^{(1)}\\x_2^{(1)}\\x_3^{(1)}\end{pmatrix}, y^{(1)}\right), \left(\begin{pmatrix}x_1^{(2)}\\x_2^{(2)}\\x_3^{(2)}\end{pmatrix}, y^{(2)}\right), \ldots, \left(\begin{pmatrix}x_1^{(m)}\\x_2^{(m)}\\x_3^{(m)}\end{pmatrix}, y^{(m)}\right)$



Forward propagation

Each node:

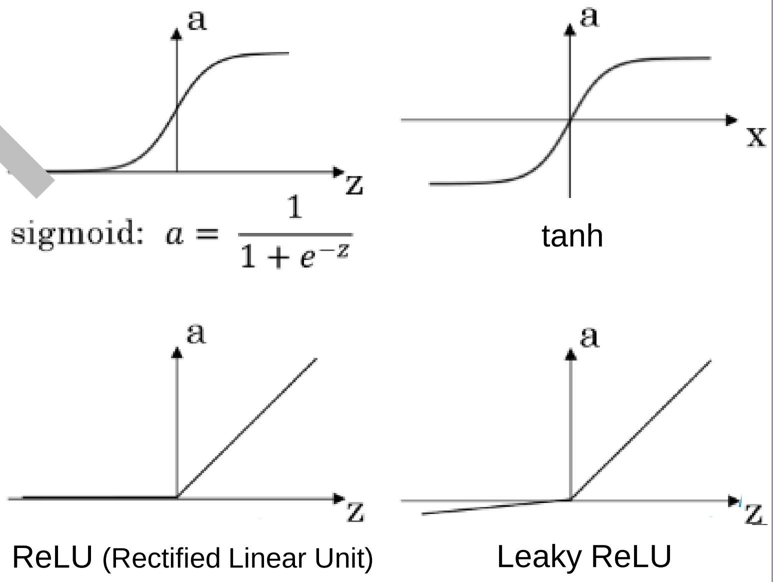


weight $\begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix}$, bias

Linear combination of inputs

activation function σ

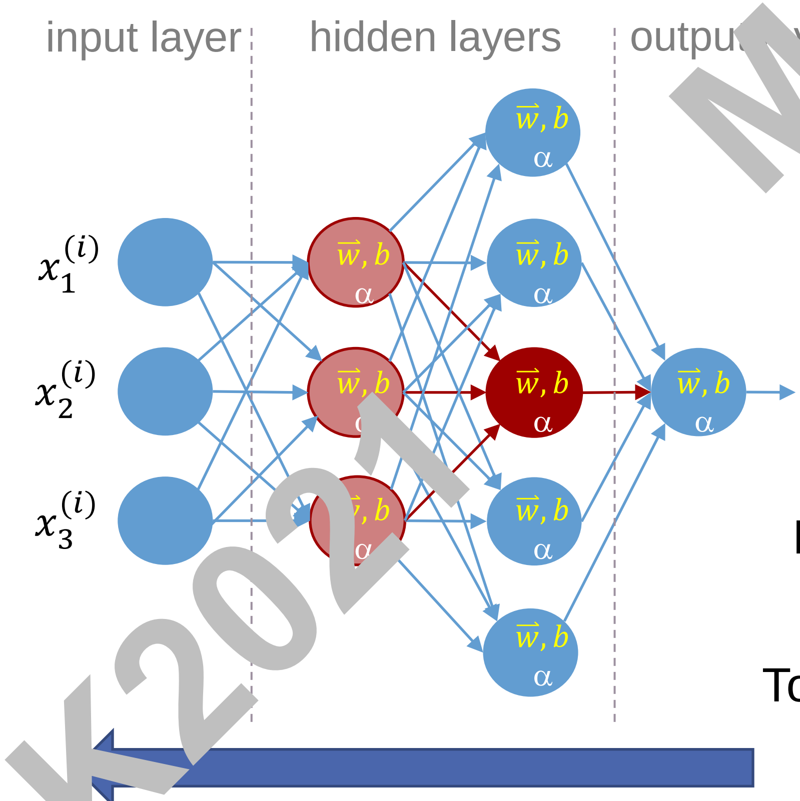
Factor that adds non-linearity



[Adapted from https://www.coursera.org/specializations/deep-learning]

Neural network basics – Backward propagation

- Data set: $\left(\begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix}, y^{(1)}\right), \left(\begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \end{pmatrix}, y^{(2)}\right), \ldots, \left(\begin{pmatrix} x_1^{(m)} \\ x_2^{(m)} \\ x_3^{(m)} \end{pmatrix}, y^{(m)}\right)$



Backward propagation

$\hat{y}^{(i)} \leftrightarrow y^{(i)}$

quantify accuracy

Loss function: $L(\hat{y}^{(i)}, y^{(i)})$

Total loss: $J = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$

Update $\vec{w}, b$ of each node to minimize J

At each node,

$w_1 := w_1 - \alpha \cdot dw_1$
$w_2 := w_2 - \alpha \cdot dw_2$
$w_3 := w_3 - \alpha \cdot dw_3$
$b := b - \alpha \cdot db$

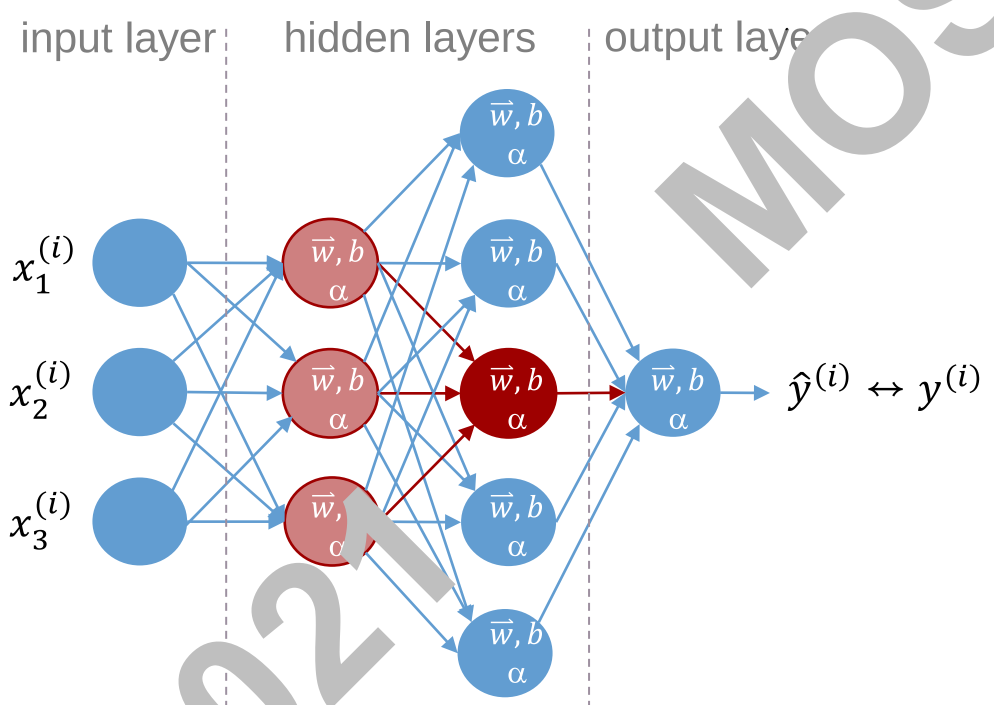
learning rate α: determines magnitude of change

[Adapted from https://www.coursera.org/specializations/deep-learning]

Neural network basics - Training

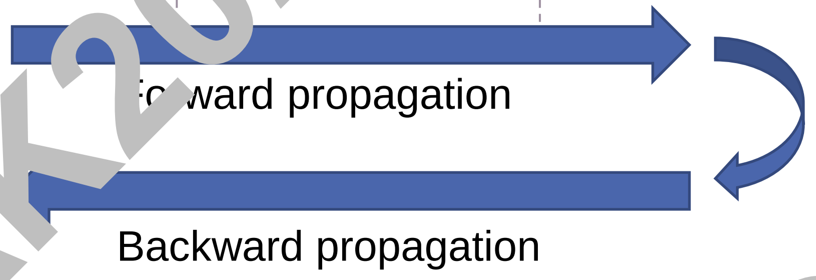
- Repeat forward & back propagation until desired accuracy is reached

  → fix weight & bias - **Arbitrary function can be expressed based on user data**



Parameters:
- weight & bias of each node

Hyperparameters (parameters that control parameters):
- # of hidden layers, # of nodes at each layer
- choice of activation function
- learning rate
- # of epoch



1 epoch: 1 forward & backward propagation for all data

[Adapted from https://www.coursera.org/specializations/deep-learning]

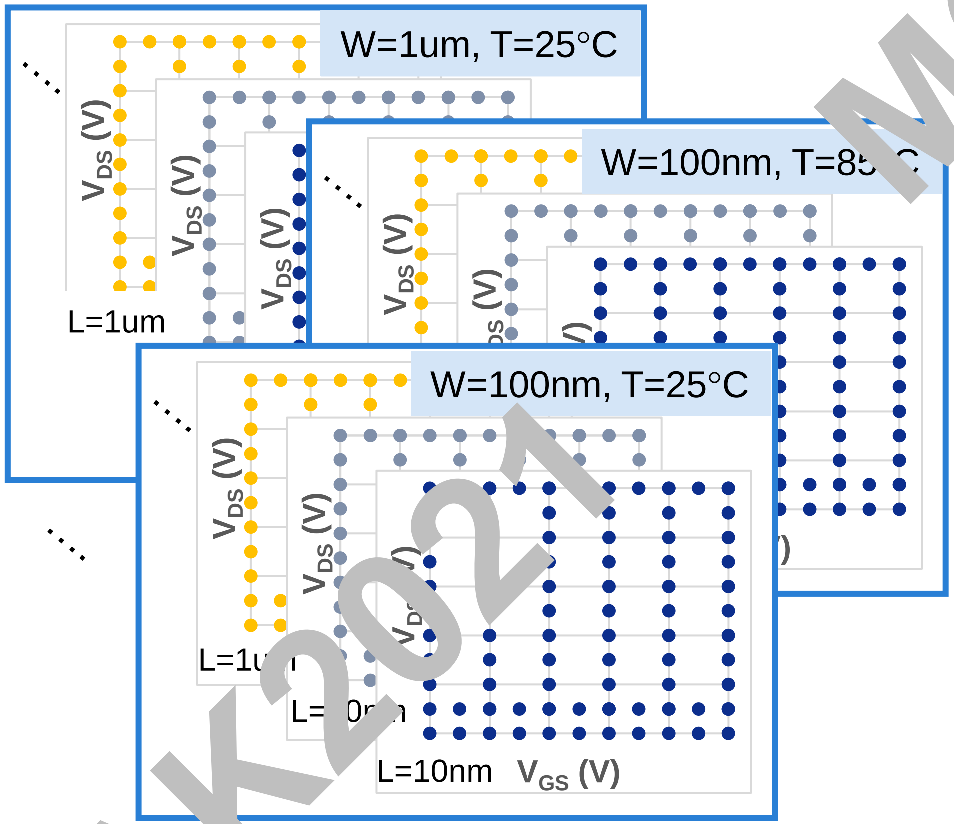
Data-based neural network device model

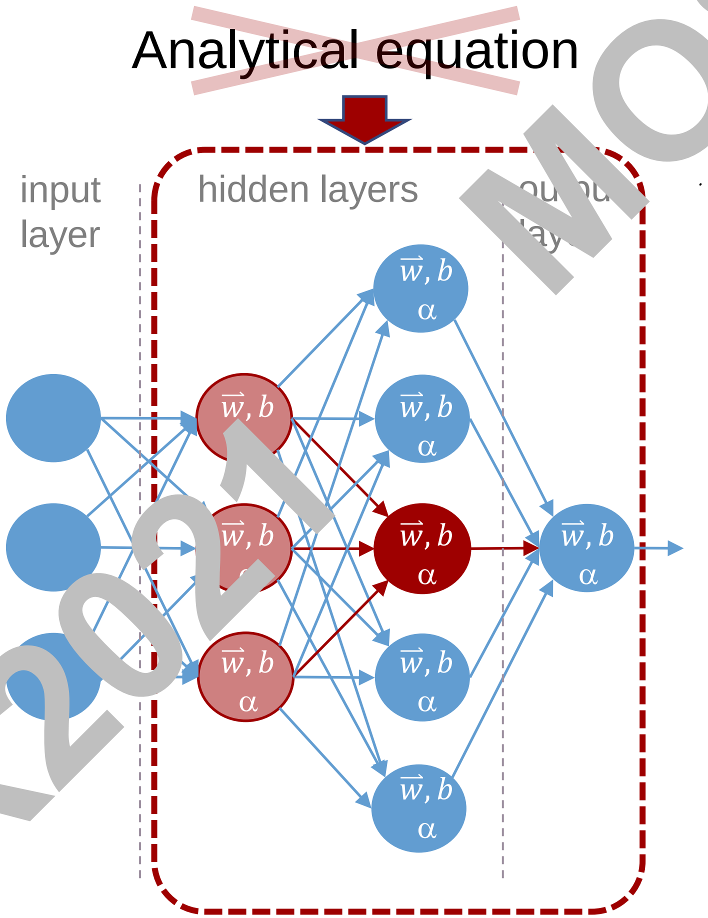
Example case

Goal: Generate device model for transistors for any {W, L, T}

W: channel width
L: channel length
T: temperature

Available measured data:
I-V data for *x* different combinations of {W, L, T}





Voltages (e.g. V_{GS}, V_{DS})
Parameters (e.g. W, L, T)

Currents (e.g. I_D)

Neural compact model
Fixed NN trained with measured data

Outline

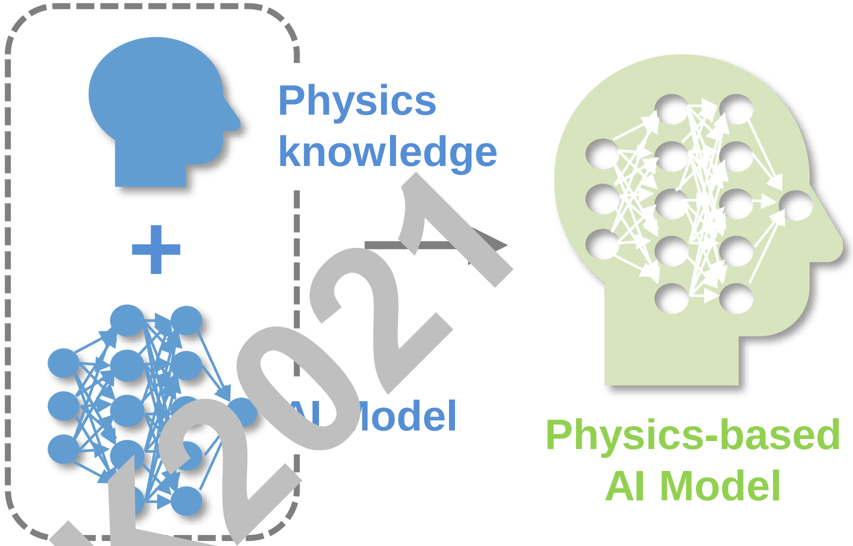
- Motivation
- Introduction to neural network model
- Knowledge-based NN model
  - Training with inductive bias
  - Building prior knowledge from multiple tasks
- End-to-end autonomous technology development

Simple data-based regression is not sufficient

- Can violate the laws of physics

→ Train with inductive bias

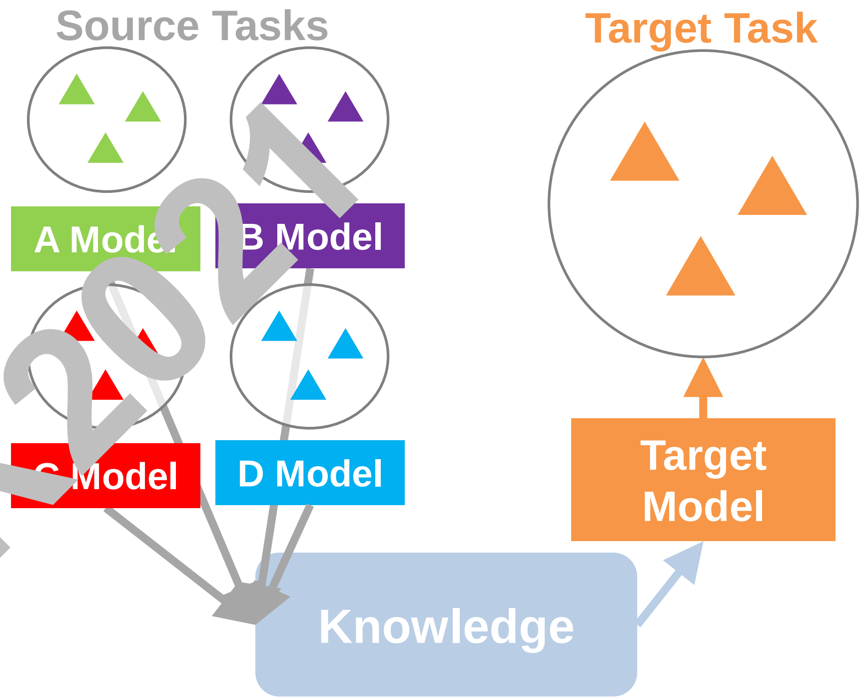
- Domain knowledge based constraints
- Physics-based NN



- Amount of data may not be sufficient

→ Build prior knowledge from multiple tasks

- Multi-task learning
- Meta learning

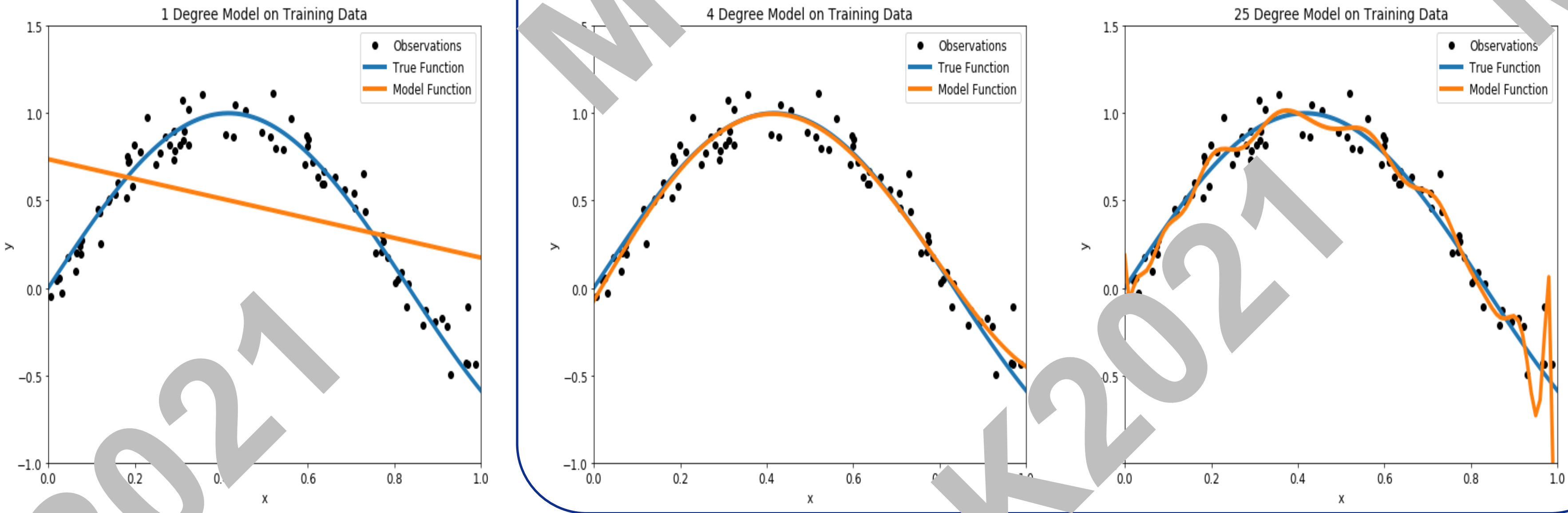


Domain knowledge based constraints - Concept

- <u>Reduce degree of freedom</u> by incorporating human-knowledge of the domain

Add constraints to avoid overfitting (regularization)

Underfitting | Good fit | Overfitting

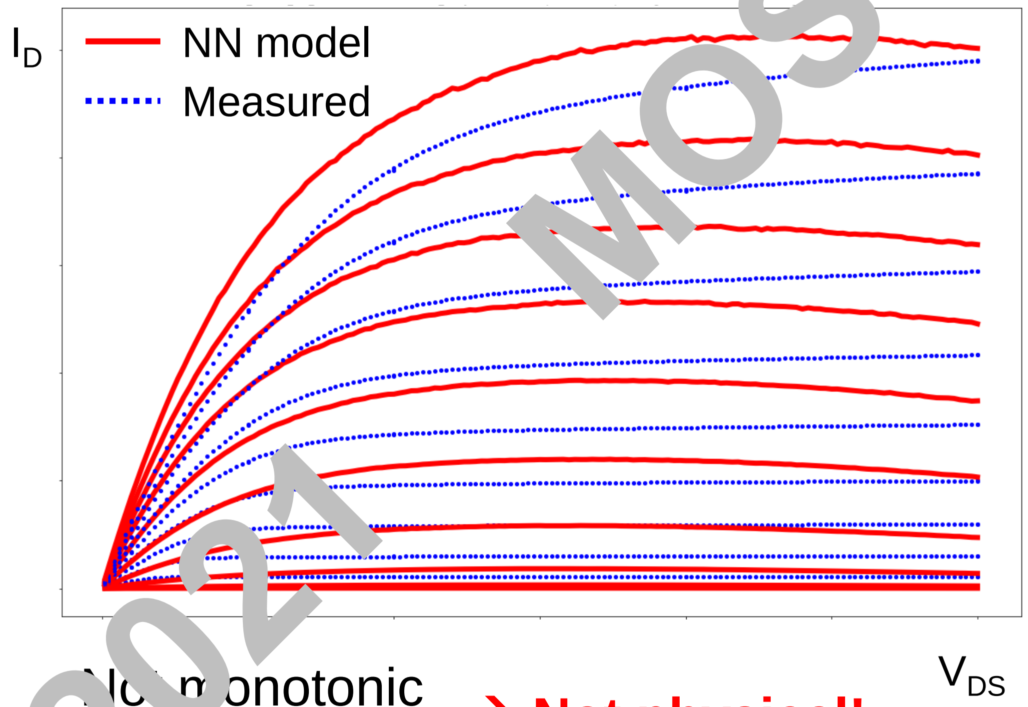


[https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765]

Domain knowledge based constraints – Device example

- Monotonicity
  - Knowledge: Increasing voltage increases current
  - Constraint: Additional term in cost function that increases when increasing voltage decreases current
- Smoothness
  - Knowledge: Increasing voltage increases current smoothly
  - Constraint: Additional term in cost function that increases when I-V curve is not smooth

w/out domain knowledge based constraints



Not monotonic
Not smooth

→ **Not physical!**

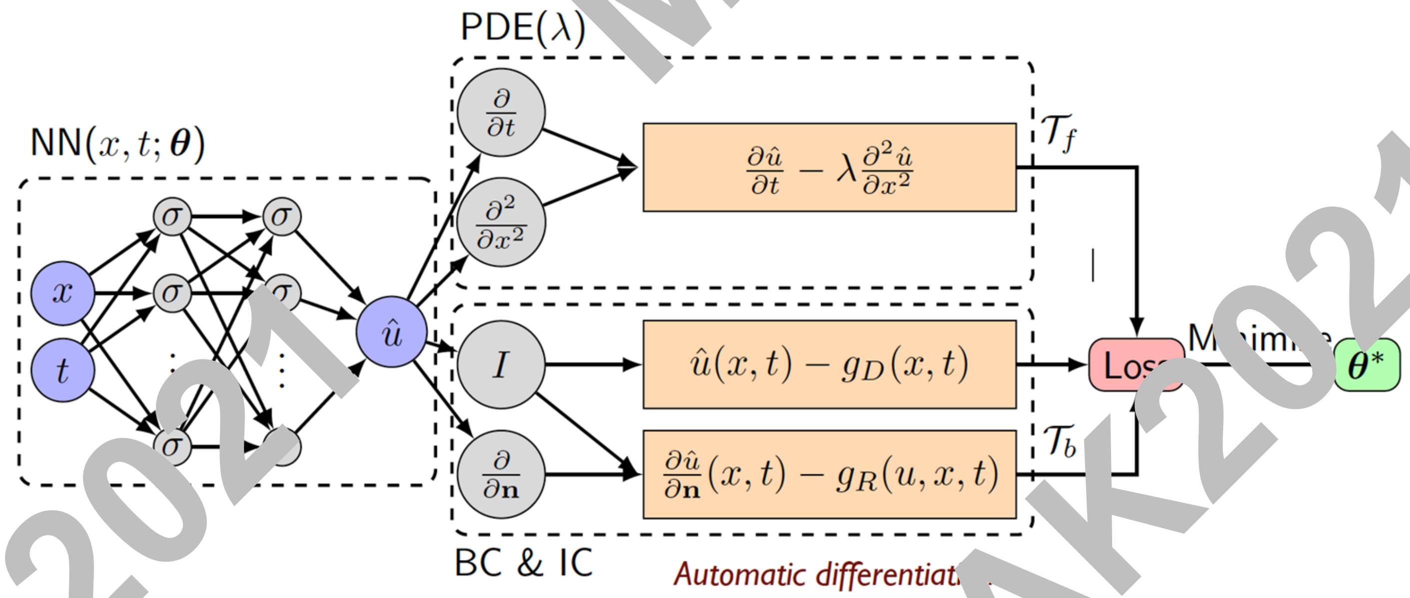
Physics-based NN – Concept

- Modify NN training scheme or architecture by incorporating underlying physics

Example

Train NN so that governing PDE, boundary condition, and initial condition is satisfied

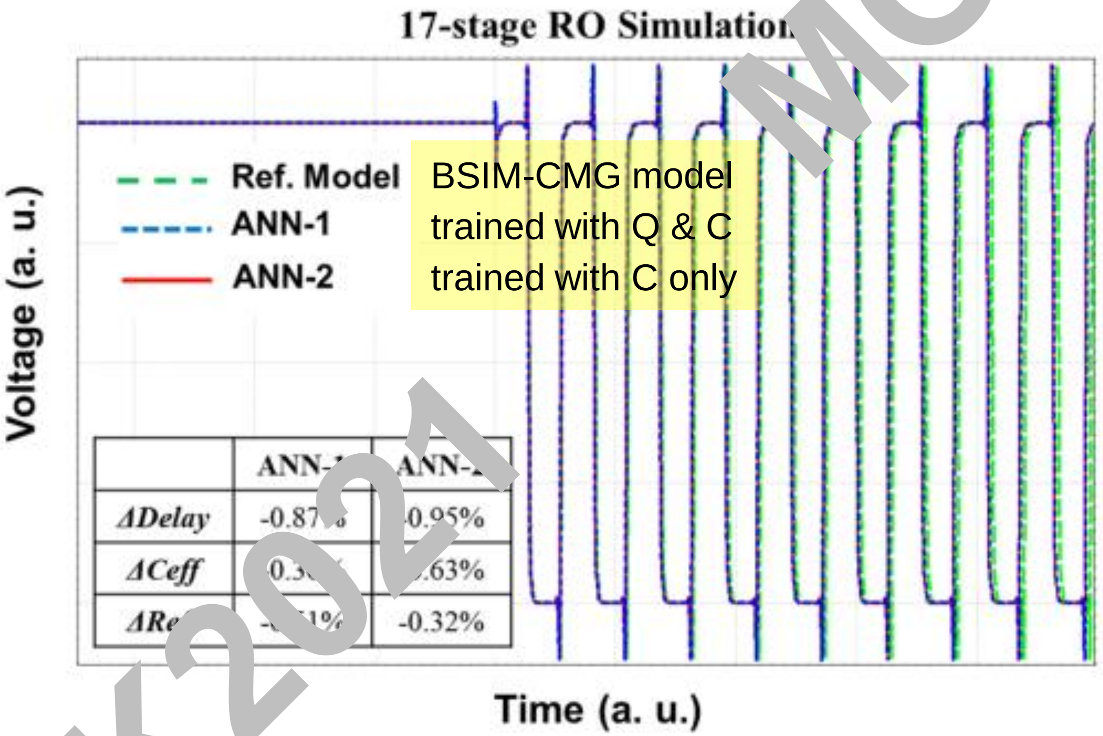
Physics-Informed Neural Network (PINN)



[Lu et al., "DeepXDE: A deep learning library for solving differential equations," *SIAM Rev.*, 2021]

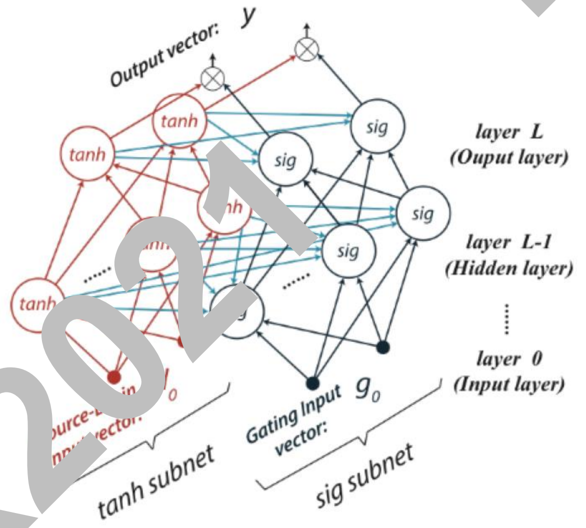
Physics-based NN – Device example

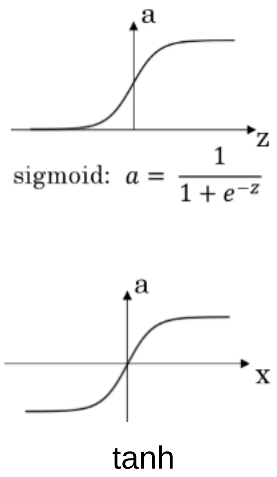
- For modeling Q, use loss function to fit only C (data more readily available than Q)
  - Use knowledge that for simulations, although Q is needed, C (=dQ/dV) is what matters



[J. Wang et al., "Artificial neural network-based compact modeling methodology for advanced transistors," TED, 2021]

- Select activation functions for NN considering known I-V behavior
  - Choose tanh for V_{DS} input branches (I_D=0 @ V_{DS}=0) and sigmoid for V_{GS} input branches (I_D≈0 @ V_{GS}=0)

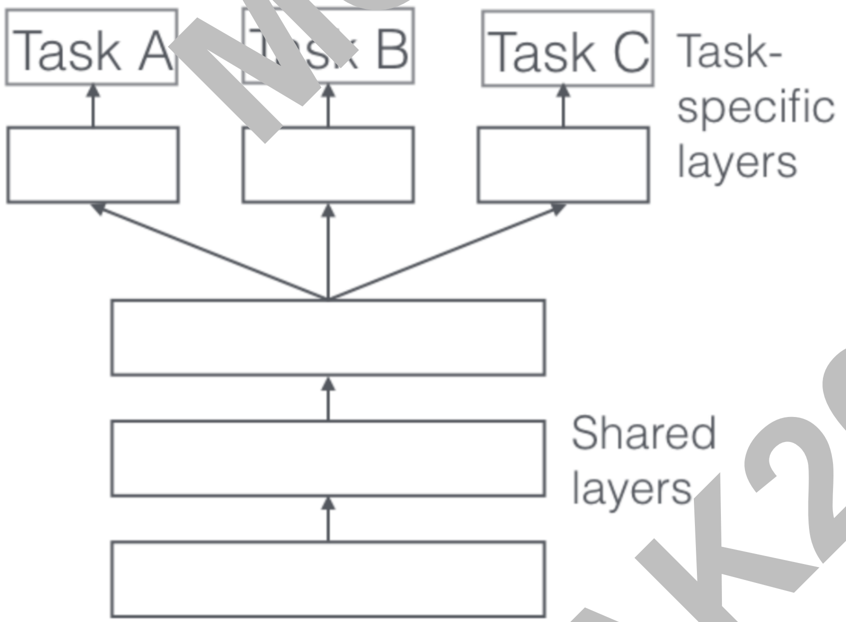




[M. Li et al., "Physics-inspired neural networks for efficient device compact modeling," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, 2016]

Multi-task learning – Concept

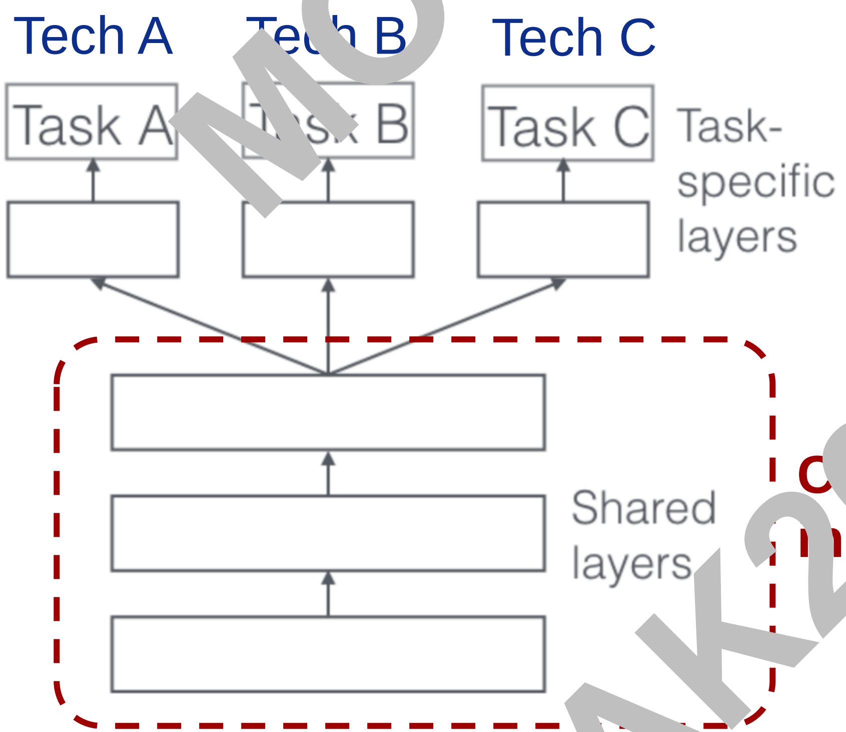
- Learn generalized representation by training with related tasks simultaneously
  - Shared layers with general properties + task-specific layers



[S. Ruder, “An Overview of Multi-Task Learning in Deep Neural Networks,” arXiv:1706.05098v1 [cs.LG]]

Multi-task learning – Device example

Example case:
- Goal – Generate device model for transistors for any {W, L, T} fabricated with <u>technology A</u>
- Available measured data:
  - I-V data for *x* different combinations of {W, L, T} with <u>technologies A, B, C</u>

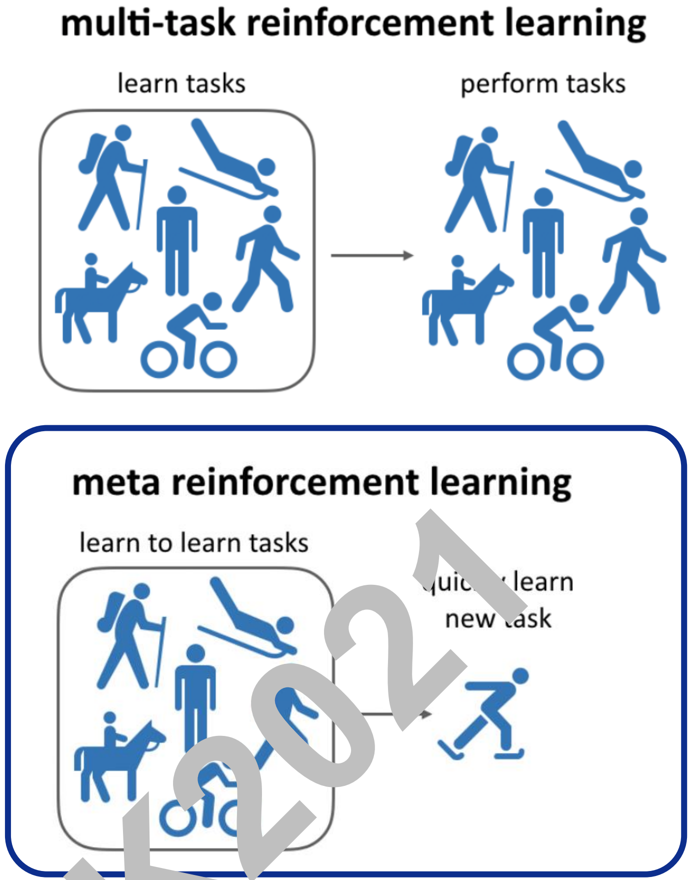


Contains general technology-independent properties

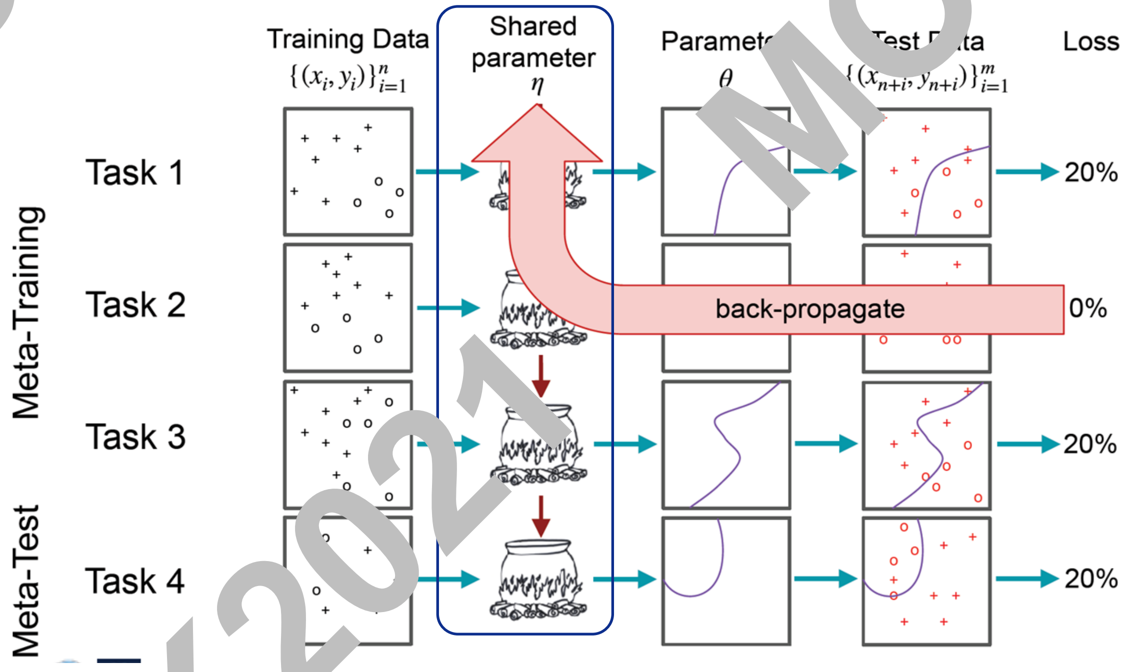
[S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv:1706.05098v1 [cs.LG]]

Meta learning – Concept

- Extract prior knowledge by "learning to learn" by training with numerous related tasks



[https://meta-world.github.io/]



[Y. W. Teh, "Meta-Learning – an idiosyncratic tutorial, MLSS, 2020]

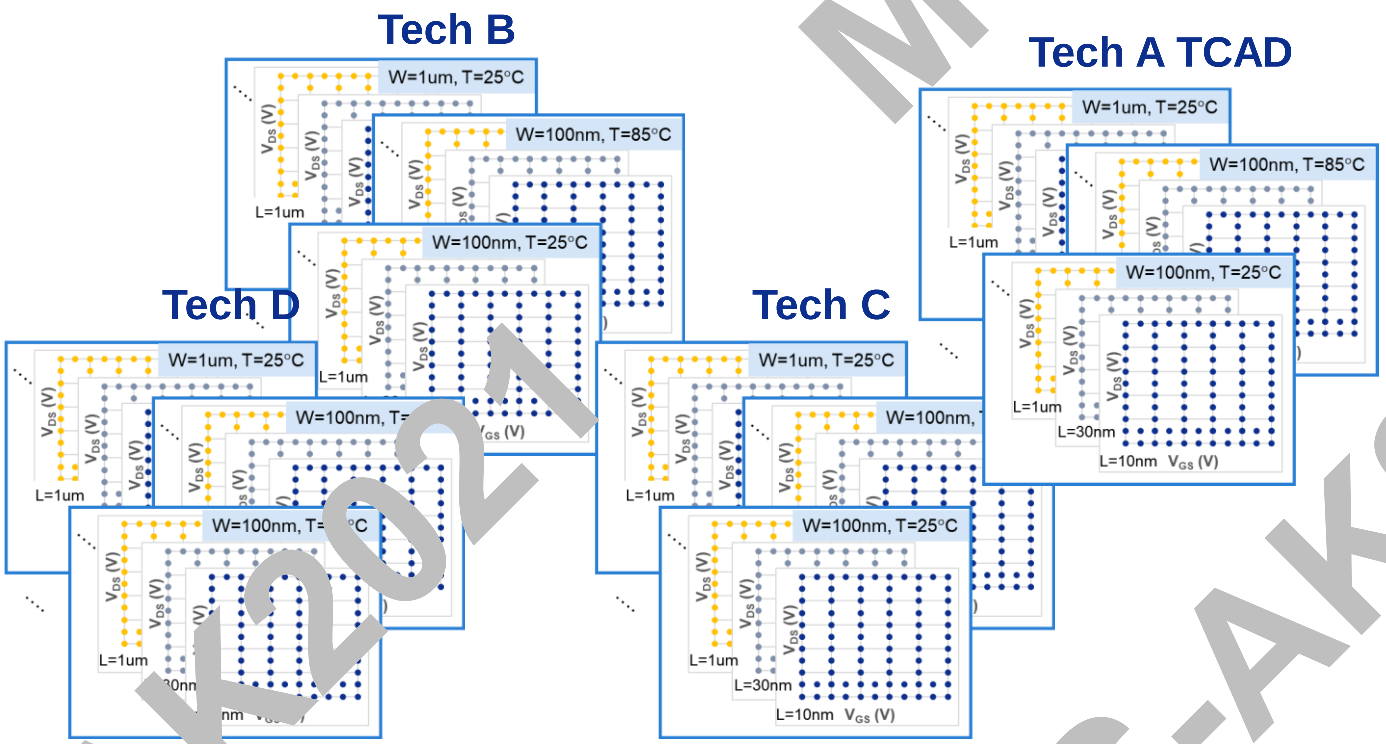
1. Base-learning: Solves individual tasks
2. Meta-learning: Individual tasks updated to improve generalization performance

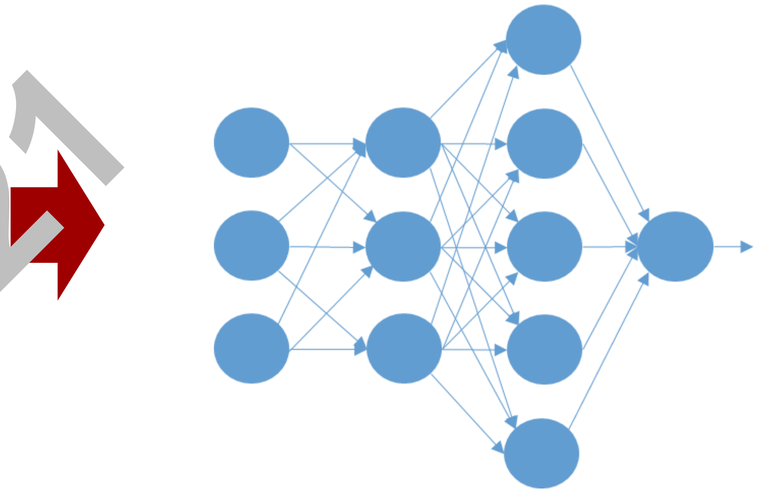
Meta learning – Device example

Example case:
- Goal – Generate device model for transistors for any {W, L, T} fabricated with <u>technology A</u>
- Available measured data:
  - I-V data for *x* different combinations of {W, L, T} with <u>technologies A, B, C, D, etc.</u>

Learn to learn tasks w/ other technology data and/or TCAD data

Quickly learn tech A





Outline

- Motivation
- Introduction to neural network model
- Knowledge-based NN model
  - Training with inductive bias
  - Building prior knowledge from multiple tasks
- End-to-end autonomous technology development

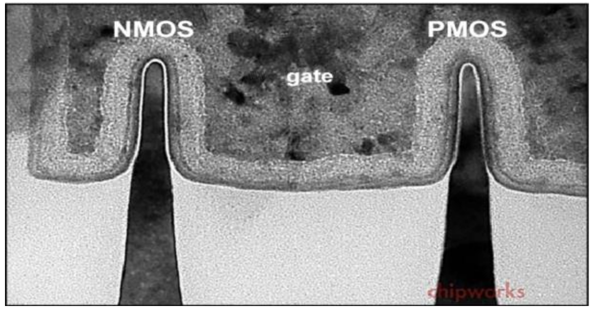
End-to-end autonomous technology development

New EDA market

Focus of conventional EDA market

FAB

Device hardware

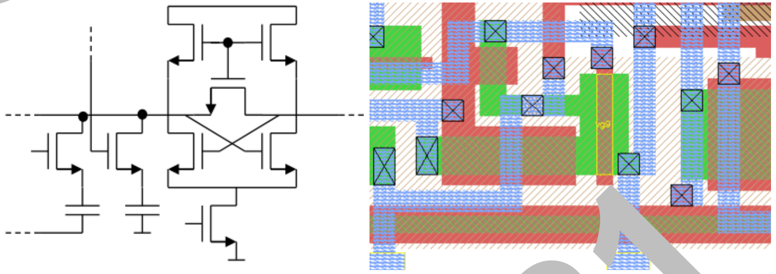


[chipworks / TSMC 16nm finFET]

Device model

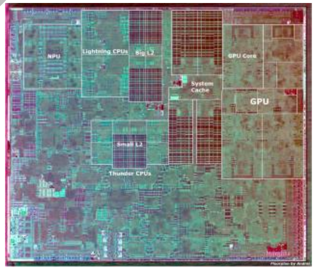
IC DESIGN

Circuit simulation & chip design



FAB

Chip hardware



[AnandTech modified TechInsights Apple A13 die shot]

<Chip manufacturing process>

Full automation from device design to chip design

- A new paradigm in the global semiconductor industry

Thank you

Alsemy