Painting on Placement:

Forecasting Routing Congestion using Conditional GANs

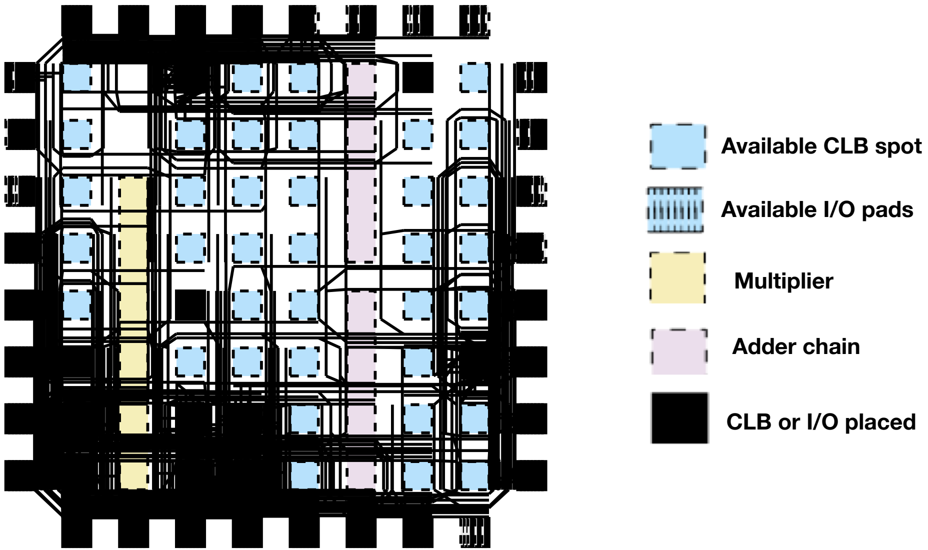
Cunxi Yu and Zhiru Zhang
ECE
Cornell University

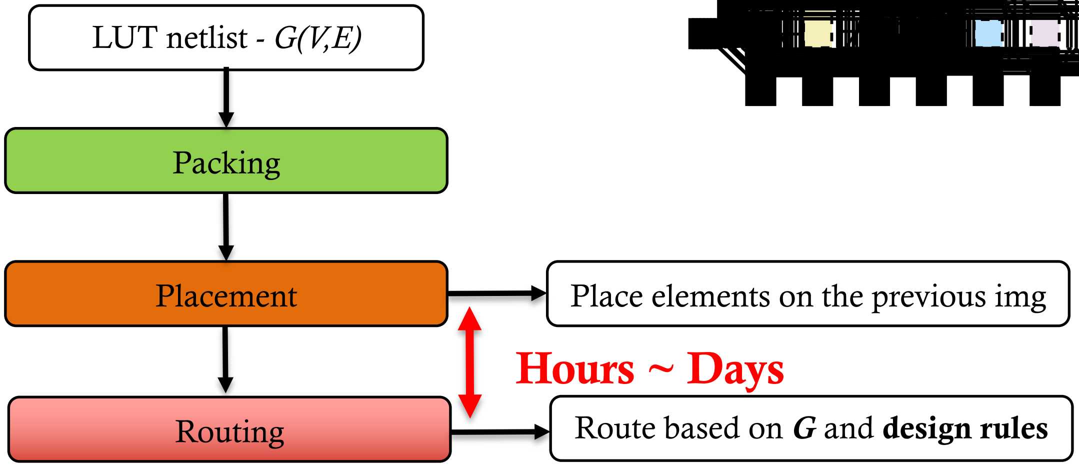
Cornell University

CSL

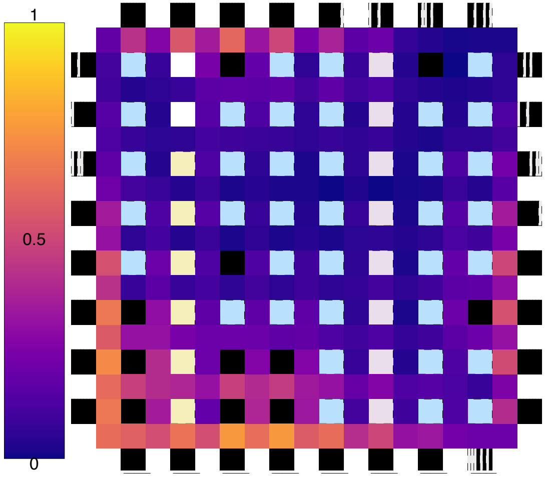
Motivation

- Physical design
  - Logic netlist to layout
    - Input: Directed graph
    - Output: physical netlist



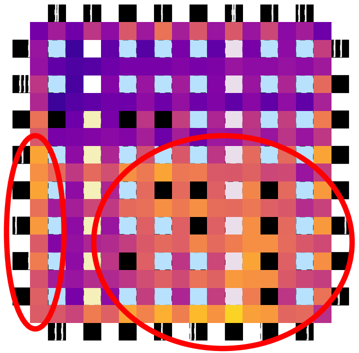


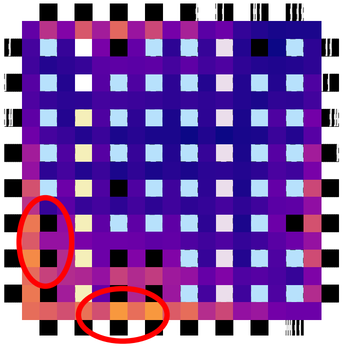
Routing Congestion



- Why congestion is important?
  - Routability
  - Timing closure
- How is congestion evaluated?
  - Utilization of routing channels
- What makes differences?
  - **Placement**

Very time-consuming to get post-routing congestion.





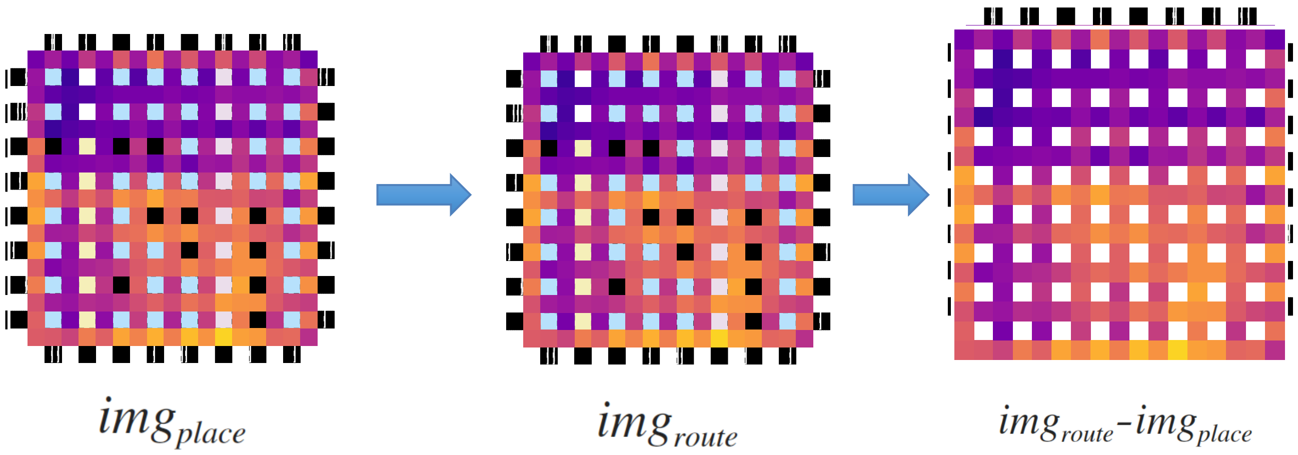
Related Works (Todo)

- ML for EDA
  - Front-end applications
    - Synthesis [Dai et al., FCCM'18] [Haaswijk ISCAS'18]
    - Design flows [Yu et al., DAC'18] [Ustun et al., FCCM'19]
  - Back-end applications
    - Place and route [Xie ICCAD'18] [Pui ICCAD'17][Huang et al., DATE19]
    - Manufacturability [Xu et al., TCAD'18] [Ye et al., DAC'19]
  - Analog IC design
    - Design and layout of analog designs [Wang et al., arxiv][Xu et al., DAC'19]

- **Limitations**
  - Hotspots/partial congestion estimation only [Xie ICCAD'18] [Pui ICCAD'17] [Yang et al., TCAD'18]
  - Require early routing information as features [Xu et al., TCAD'18] [Huang et al., DATE19]

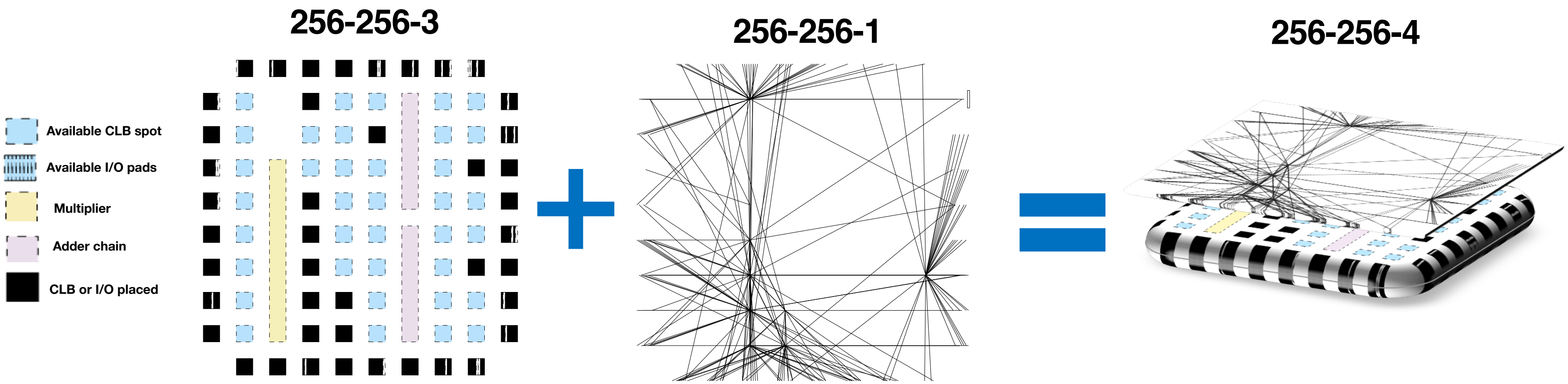
Approach – Painting on Placement

- Forecast congestion from placement
  - Learning a image-to-image mapping [Isola et al, CVPR'17]
    - Congestion and placement can be represented as image
- Forecast as *image colorization*
  - **Colorizing the routing channels of the placement image**
    - Exact the same underlying structure



Features

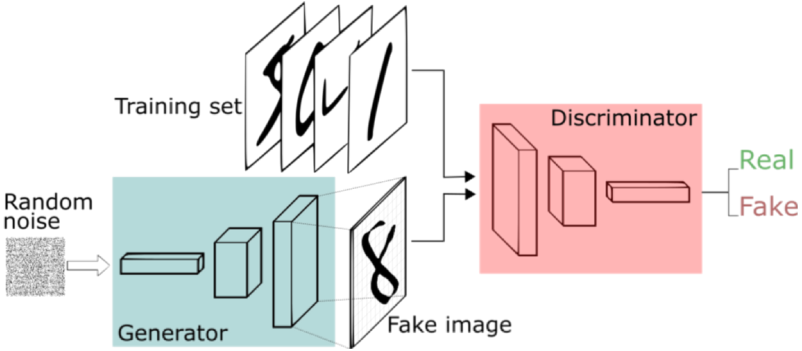
- Embedding of placement and G(V,E)
  - Placement image (RGB)
  - Visualization of netlist after placement (grayscale)
  - Final input: stack two images as a four-channel matrix
    - e.g., 256-256-4



- Tips for constructing feature images
  - **Color code** to distinguish different types of cells
  - Adjust **resolution** to distinguish each cell as a objective

Generative Adversarial Nets

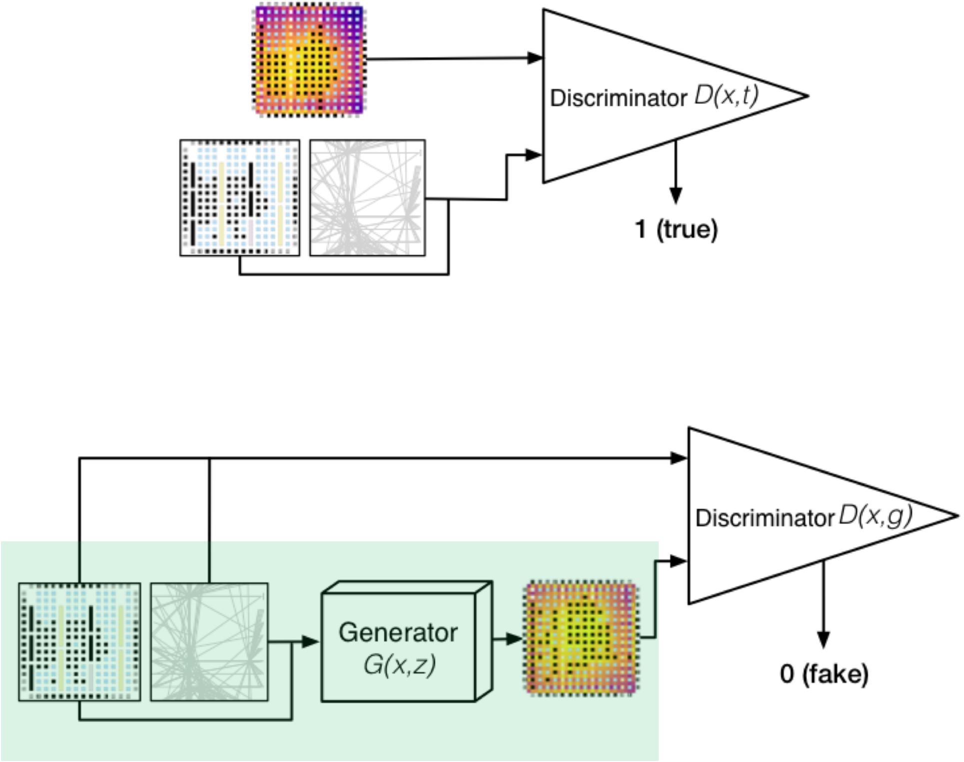
- Generative Adversarial Nets (GAN) [Goodfellow NeurIPS14]
  - Discriminator: learns to **classify** true or fake
  - Generator: learns to **fool** discriminator
  - No control over modes of the data to be generated



- **Conditional GAN (cGAN)** [Mirza arxiv14]
  - Adding the additional parameter to control the generator
  - Model used in this work
    - G: Fully Convolutional Networks (FCN) with 7 down/up-sampling layers
    - D: CNN based binary classifier

Training and Inference

- Training
  - Discriminator
    - Learns to **classify** true or fake
  - Generator
    - Learns to **fool** the discriminator
  - Input-output pair
    - Stacked matrix
      - 256-256-4
    - Congestion heatmap
      - 256-256-3
- Inference
  - Run generator



Dataset and Results

- Image generator implemented based on VPR
  - VPR configs: seed, ALPHA_T, INNER_NUM, place_algorithm
  - Training: 1x1080Ti, < 3 hours
  - Inference: < 0.2 s on 1x1080Ti (batch=1)
- Min-congestion exploration
  - e.g., target design = *dcsg*

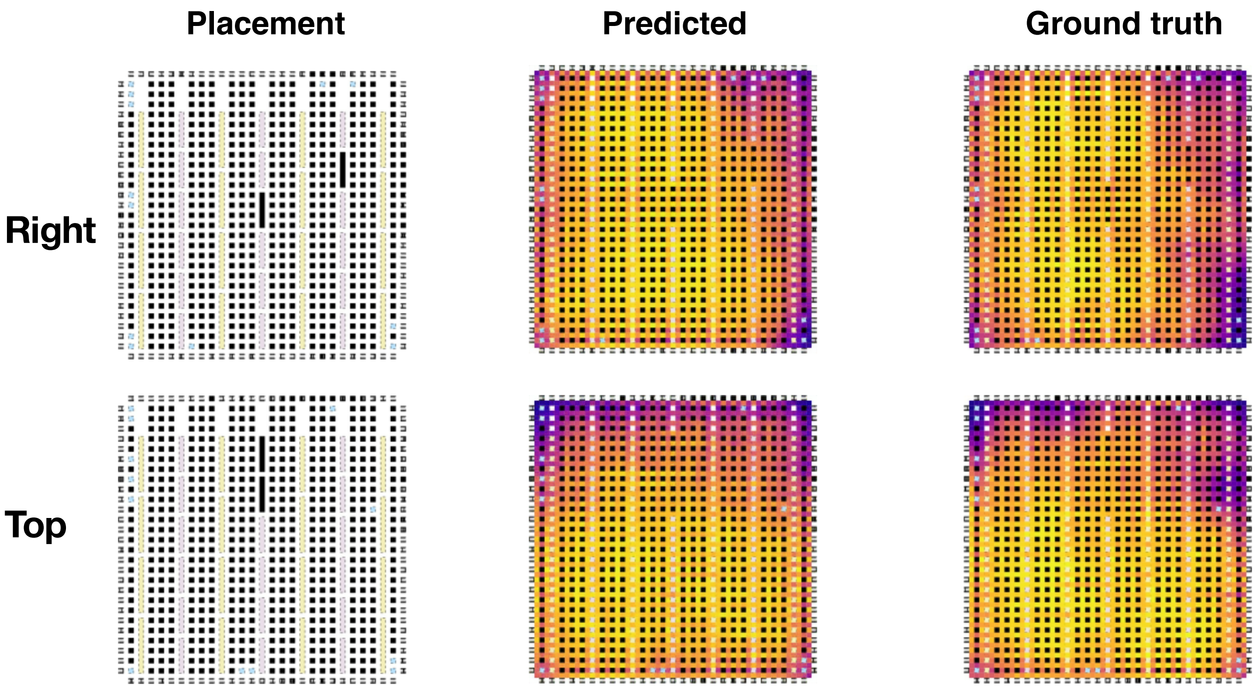
Design	# LUTs	# FFs	# Nets	# samples
diffeq1	563	193	2059	200
diffeq2	419	96	1560	200
SHA	2501	1047	5023	200
OpenRISC	2823	911	10910	200
ODE	5488	670	12336	150
bfly	9503	1748	38582	150
dcsg	9088	1618	36912	150

Training Set

10 samples

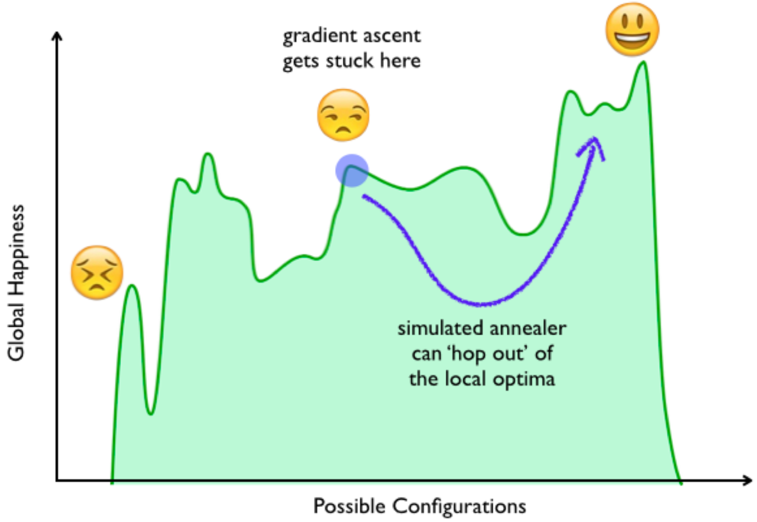
Results – Placement Exploration

- Constrained placement exploration
  - Placement with biased routing congestion region



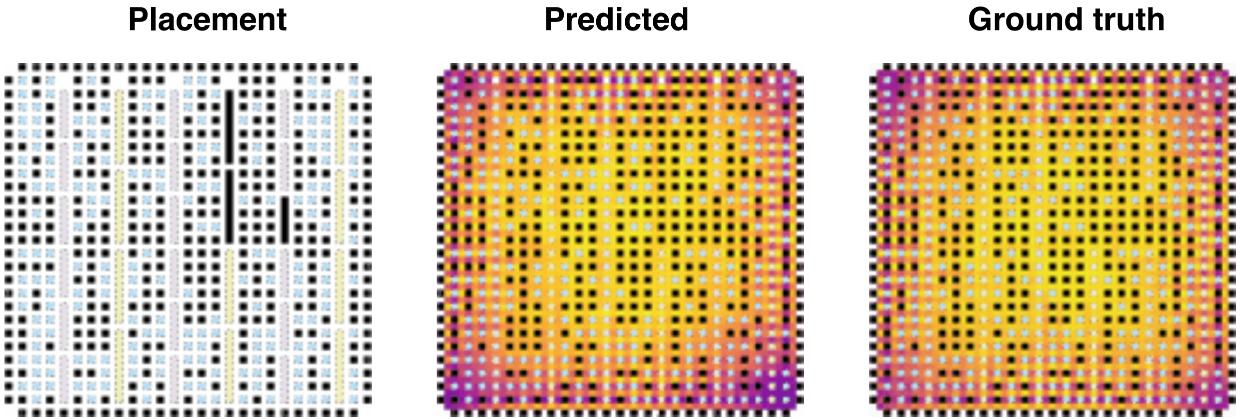
Results – Real-time Forecast

- Visualization of *simulated annealing* placement
  - Swap locations iteratively
  - Every 200 iterations



Results – Real-time Forecast

- Visualization of *simulated annealing* placement
  - Swap locations iteratively
  - Every 200 iterations



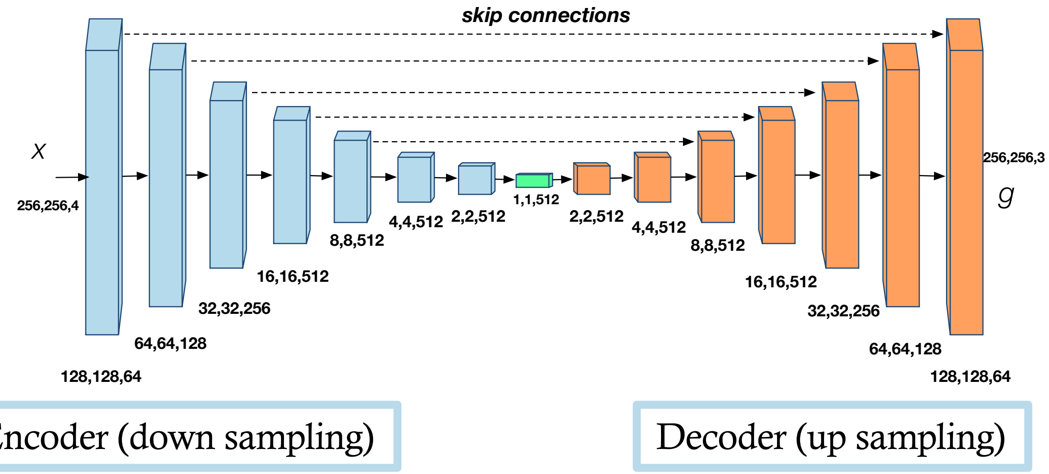
Conclusion

- Forecasting Routing Congestion using Conditional GANs
  - Estimate routing utilization of all routing channels
  - Generate high-quality full routing congestion heatmap
    - Image-to-image mapping as image colorization
- **Limitations**
  - Model is device specific
  - Still require 5-10 image pairs for new design
  - Black-box inference is inefficient while is integrated with PnR tools
- **Future work**
  - Accelerate timing closure by packing/placement exploration
  - Integrate with Yosys-NextPnR flow and VTR(ABC-VPR) flow

Thank you!

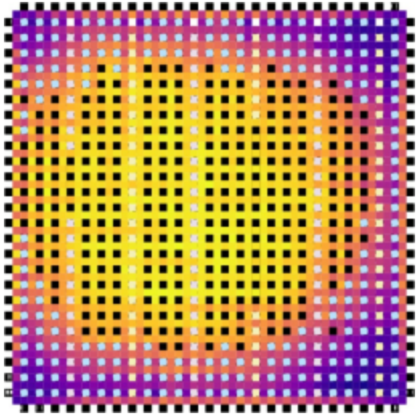
Conditional Generative Adversarial Nets

- Why conditional GAN (cGAN) ?
  - Condition on the input image
  - **Input and output has similar underlying structure**
- Generator
  - Fully Convolutional Networks (FCN)
- Discriminator
  - CNN binary classifier

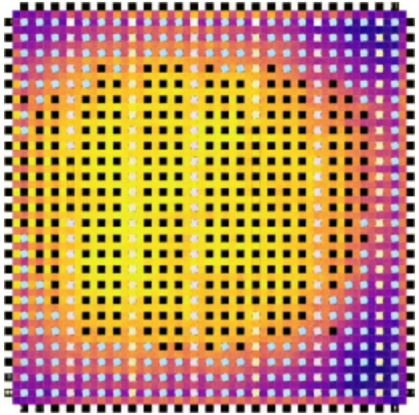


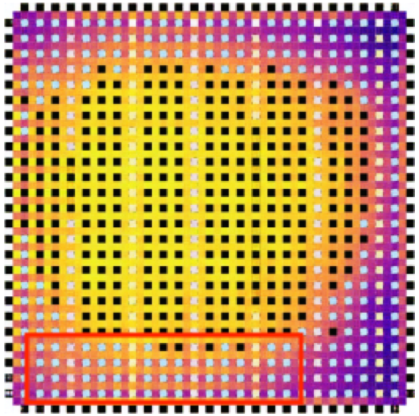
Results - Analysis of Skips and L1 Loss

Ground Truth

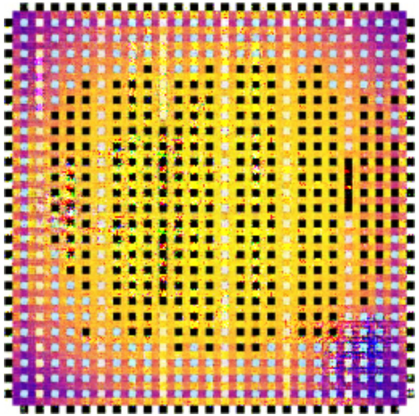


Predict (L1+skips)





Predict (skips)



Predict (single skip+L1)

Results - Analysis of Skips and L1 Loss