Vision Transformer-Based Routability Prediction with DINOv2 Transfer Learning Using the CircuiteNet2.0 Dataset

Shundan Xiao
Stanford
sxiao2@stanford.edu

Hangfei Lin
Stanford
hangfei@stanford.edu

Abstract

Routability prediction is vital in electronic design automation (EDA), traditionally tackled with heuristic methods. Leveraging the CircuiteNet2.0 dataset, we propose a Vision Transformer (ViT)-based approach, enhanced with transfer learning using DINOv2, to improve prediction accuracy. Our ViT-Hybrid model combines transformer encoding for capturing global spatial relationships and CNN-based decoding for detailed reconstruction. Evaluated on over 10,000 chip design samples, our model shows performance comparable to existing baselines. This study demonstrates the potential of integrating Vision Transformers and DINOv2 in EDA, setting the stage for more efficient and accurate routability prediction.

1. Introduction

The scaling down of semiconductor devices poses unprecedented challenges in electronic design automation (EDA), particularly in routability prediction, which ensures that chip designs are free from physical and electrical violations before manufacturing. This task has become more complex with the increase in circuit density and design intricacies. Routability prediction is vital as it impacts the manufacturability, performance, and yield of the final chip products.

Traditionally, routability has been addressed through heuristic methods or simplified computational models, which often fall short in handling the complexities of modern integrated circuit (IC) designs. Recent advancements in machine learning, especially deep learning, have opened new avenues for addressing these challenges more effectively. The CircuitNet2.0[4] dataset provides a rich resource for training and validating new predictive models, offering comprehensive multi-modal data from over 10,000 chip design samples. In our

Our paper introduces an innovative adaptations of the VisionTransformer architecture designed specifically for predicting routability from high-dimensional, image-based data representations of IC layouts. We aim to improve prediction accuracy over the baseline.

1.1. Routibility Prediction

Routability prediction is a crucial aspect of the electronic design automation (EDA) process that involves predicting the feasibility of routing electrical connections on a chip without causing physical or electrical violations. This process is essential for ensuring that the design of integrated circuits (ICs) can be successfully manufactured and will operate as intended.

Routability prediction assesses the likelihood that the interconnects (wires) within an IC design can be placed without exceeding the available routing resources or violating design rules. These rules are essential to prevent issues such as signal interference, delay faults, and other electrical problems that can degrade the performance of the final product or even render it non-functional.

2. Related Work

2.1. Machine Learning for Routability Prediction

In the past decade, there has been a growing focus on utilizing machine learning for the design automation of ICs. Specifically, for routability prediction, the layout of a chip's physical placement can be viewed as an image, making it natural to model this problem using existing CNN-based image-to-image translation methods. RouteNet [9] and GPDL [8] are two of the most notable studies that leverage Fully Convolutional Neural Networks for routability prediction. Both employ an encoder-decoder architecture, where convolutional (CONV) and pooling (POOL) layers are used for downsampling, and transposed-convolutional (TRANS) layers are used in the decoder for upsampling, ensuring the final output matches the input size. In this paper, we use GPDL as the base model to compare model performance, as our approach is also CNN-based.

Other research efforts have explored the topological correlation among circuit netlists by modeling the netlist as

graph nodes and relational edges, leveraging Graph Neural Networks (GNNs) to learn latent relations [11] [6].

Combining CNNs and GNNs has also become a popular research area recently, as it can take advantage of both geometric and topological information [12].

2.2. Vision Transformer Models

Transformers [10] were initially proposed for machine translation and have quickly become the state-of-the-art in most Natural Language Processing (NLP) tasks. Unlike recurrent networks, Transformers do not suffer from memory limitations due to their unique self-attention mechanisms. There have been attempts to utilize transformer-like self-attention architectures with CNNs)for computer vision tasks.

Vision Transformers (ViTs) [3] represent a significant adaptation of Transformer architecture for computer vision tasks. ViTs operate by splitting an image into patches, flattening these patches, and then producing lower-dimensional linear embeddings from the flattened patches. After adding positional embeddings, these sequences are fed into a Transformer encoder. ViT models are typically pre-trained on large datasets and then fine-tuned on smaller datasets for specific tasks, such as image classification. ViTs are highly adaptable and capable of efficiently processing both local and global features, making them suitable for applications that involve analyzing large-scale chip design images.

2.3. Transfer Learning with DINOv2

Transfer learning has emerged as a powerful technique in machine learning, allowing models pre-trained on large datasets to be adapted for specific tasks with smaller datasets. DINOv2 [1] is a state-of-the-art self-supervised learning model trained on a vast corpus of internet images. It has shown excellent performance in various computer vision tasks, including segmentation, classification, and depth estimation.

In our research, we explored the use of DINOv2 for routability prediction. By integrating DINOv2's robust feature extraction capabilities into our Vision Transformer-based model, we aimed to leverage its pre-trained knowledge to enhance the performance of our predictive model. This approach aligns with recent trends in using transfer learning to bridge the gap between large-scale pre-training and domain-specific fine-tuning, particularly in fields with limited data availability like EDA.

2.4. Public Dataset for EDA

The EDA industry has traditionally been a closed environment compared to the broader software development world, primarily due to business concerns that necessitate keeping data private. Hardware specifications vary significantly, and data collection in this field is both time-consuming and computationally intensive. The EDA industry requires an "ImageNet moment" to foster open data sharing and advance research.

CircuitNet [2] represents the first large-scale open-source AI dataset in this field. Its successor, CircuitNet2.0, offers even better coverage, addressing a wider range of design targets and prediction tasks [4].

3. Methods

In this section, we start by outlining our task and describing the baseline model used for comparison. Following this, we detail our efforts to enhance the baseline model.

3.1. Task Description

Our task is to predict routability by estimating the congestion level during the placement stage in EDA design. Congestion measures the density of routed wires on the layout and is often used to optimize the positions of macros and cells during floorplanning and placement. Since obtaining precise congestion data requires time-consuming global routing, a fast prediction model is desirable to accelerate the design process. Thus, the goal of congestion prediction models is to use data from the placement stage to predict the congestion map after global routing, saving valuable time in the design cycle. An example of a congestion map can be seen in Figure 4.

3.2. Baseline Method

Congestion prediction can be framed as an image-to-image translation problem. Given input feature images, a congestion map is generated and compared to the ground truth. For detailed examples of features and predicted results, please refer to Section 4.2.

Our baseline model, GPDL, is adopted from [8]. It is a generative model that utilizes a Fully Convolutional Network (FCN) encoder-decoder architecture to translate image grid features into a congestion map. Figure 6 illustrates the model architecture described in [8]

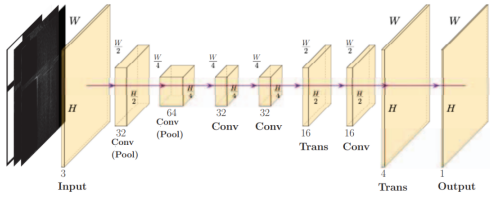


Figure 1: GPDL Model Architecture

3.3. Our Approaches

GPDL is a compact model particularly effective for congestion prediction in the hardware design domain. This ef-

fectiveness largely stems from the relatively small training dataset typical in chip design, which discourages the use of deeper CNN models due to the increased risk of overfitting. Below, we identify some of GPDL's limitations before suggesting potential improvements.

1. **Limited Receptive Field:** The use of small (3x3) convolution kernels in GPDL restricts its ability to examine distant topological relationships. Due to its limited depth and compact structure, GPDL has a constrained receptive field, hampering its capacity to capture extensive global spatial relationships.

2. **Loss of Low-Level Information:** GPDL employs pooling for down-sampling, which helps the model learn robust representations against minor spatial variations. However, this method can result in the loss of crucial low-level details during the phase of reconstructing the output image.

To address these challenges, we propose a hybrid encoder-decoder model Vit-Hybrid that integrates vision transformers with CNNs. This model employs a vision transformer to encode the image into patches, processes these patches through transformer encoder layers equipped with multihead attentions, and finally utilizes CNN upsampling techniques to reconstruct the image from feature vectors.

Leveraging the transformer attention framework enhances our model's ability to capture global spatial relationships while maintaining a relatively shallow network architecture, thus offering an unlimited receptive field.

To mitigate the loss of low-level information, we have incorporated skip connections from the transformer encoder layers to the corresponding CNN decoder layers. This allows for direct transmission of information from the encoder to the decoder, preserving essential details.

As shown in Figure 2, our model consists of two main parts: a ViT-based encoder and a decoder comprising multiple TransConv layers.

1. **Encoder:** Three features are concatenated in the channel dimension into a (N, 3, 256, 256) tensor, which is passed into a PatchEncoder layer. This layer divides the image into patches, applies a linear transformation, and adds positional embeddings before passing the tensor to the transformer encoder. A standard set of five transformer encoder layers is used to generate the latent feature output, with the size of (N, num_patches, latent_feature_dimension) for decoding.

2. **Decoder:** The latent feature is then reshaped to (N, num_patches, patch_size, patch_size), which in turn goes through five different TransConv layers to return to the original image size by doubling in size per layer.

The resulting tensor is passed through another CONV layer before entering the Sigmoid layer to produce the final prediction. There are optional skip connections between corresponding transformer encoder layers and the TransConv layers which are not demonstrated in the figure.

3.4. Transfer Learning with DINOv2

Inspired by the Cryo-Transformer (Cryo-ViT) presented in our lectures, we explored the effectiveness of transfer learning using DINOv2 for our routability prediction task. DINOv2 is a self-supervised learning model trained on a large corpus of internet images, offering state-of-the-art performance in various computer vision tasks such as segmentation, classification, and depth estimation. Given its robust feature extraction capabilities, we integrated DINOv2 into our model pipeline to enhance feature representation.

Implementation: Initially, we appended the latent features extracted from DINOv2 to the input images and passed them through our encoder-decoder network. However, this approach significantly slowed down training due to the high dimensionality of the DINOv2 output features (768 dimensions), which restricted the batch size.

To optimize this, we modified our approach by directly concatenating the DINOv2 features with the outputs of our encoder, followed by a convolution layer and a Sigmoid activation. This method leverages DINOv2's robust feature extraction while maintaining computational efficiency.

3.5. Implementation

Our code is built on top of the `CircuitNet2.0` codebase, which provides the code to train the GPDL baseline model. We implemented the ViT-Hybrid model and the Dinov2 version of ViT-Hybrid as follows:

- `ViT-Hybrid model: https://github.com/sherryxiao1988/routability/blob/main/routability_ir_drop_prediction/models/transformer.py`

- `Dinov2 version of ViT-Hybrid: https://github.com/sherryxiao1988/routability/blob/main/routability_ir_drop_prediction/models/dinov2.py`

Additionally, we wrote code to use `optuna` for hyperparameter space search:

- `Hyperparameter search code: https://github.com/sherryxiao1988/routability/blob/main/routability_ir_drop_prediction/train_transformer.py`

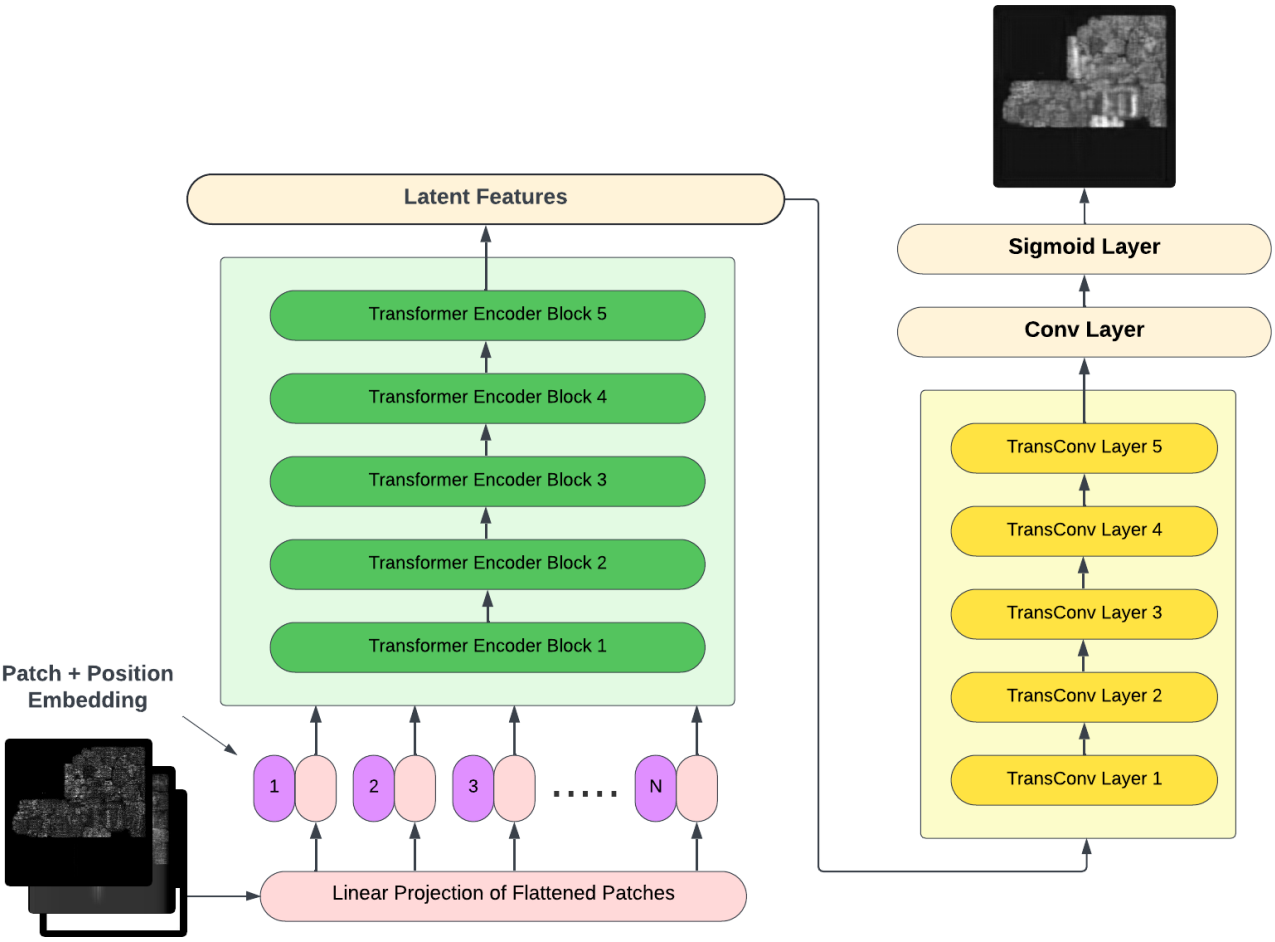


Figure 2: ViT-Hybrid Architecture.

4. Dataset and Features

4.1. Dataset

We utilize the CircuitNet2.0 dataset [4], which includes over 10,000 chip design samples for CPUs and GPUs. These samples are generated using commercial 14nm and 28nm FinFET process design kits (PDKs) and electronic design automation (EDA) tools. The dataset provides comprehensive multi-modal data, including netlist connectivity, cell placement, routing congestion, IR-drop, timing, power, and more. In this research, we focus on training with the 28nm FinFET data. Detailed statistics of our dataset are shown in Table 1

Generating these samples is both time-consuming and computationally expensive. For instance, creating a single data sample for the smallest design, *zero-riscy*, takes about 2 hours, whereas the largest design, *NVDLA-large*, requires nearly 1 week to generate a single data sample. This presents a significant challenge in the industry.

4.2. Features

For routability prediction problem, in nature, it's a pixel-level regression, or sometimes called image-to-image translation. The inputs and output are as follows.

Inputs are images produced from the circuit design process. We use 3 images as input features:

Macro Region: An image or feature map indicating the placement and distribution of larger circuit components (macros) across the chip. This helps in understanding the spatial constraints and opportunities for routing interconnects.

RUDY: A feature map representing the estimated wiring demand in different regions of the chip. This metric helps predict areas where the density of required interconnects might exceed the available routing resources.

RUDY pin: Similar to RUDY, this feature focuses

Design	#Cells	#Nets	#Macros	#Pins	#IOs	#Samples
RISCY	46,184	47,233	3	180,069	563	3,456
RISCY-FPU	65,464	66,903	3	252,390	563	3,456
zero-riscy	35,969	36,225	3	138,569	563	3,456
OpenC910-1	754,981	766,436	32	3,062,504	1,341	96
Vortex-large	1,018,221	1,107,255	376	3,731,139	1,242	74
Vortex-small	113,961	124,058	43	433,449	1,234	96
NVDLA-large	1,478,865	1,637,556	80	5,705,108	1,734	68
NVDLA-small	270,072	285,465	108	1,039,571	538	89

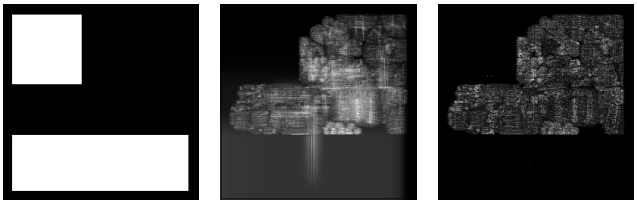
Table 1: Statistics of samples in CircuitNet 2.0.

specifically on the density related to pin connections, providing a finer granularity of prediction where pin-related congestion might occur.

We combine them along the channel dimension to form a tensor with dimensions (N, 3, 256, 256). Here, N represents the batch size, 3 indicates the number of channels, and the last two dimensions, 256 and 256, correspond to the height and width of the image, respectively.

The output is a congestion map with dimensions of 256x256 pixels, where each pixel represents the congestion level in that area. The congestion is measured on a scale from 0 to 1, with higher scores indicating more congestion.

An example of these features is demonstrated in Figure 3.



(a) Macro Region (b) RUDY (c) RUDY Pin

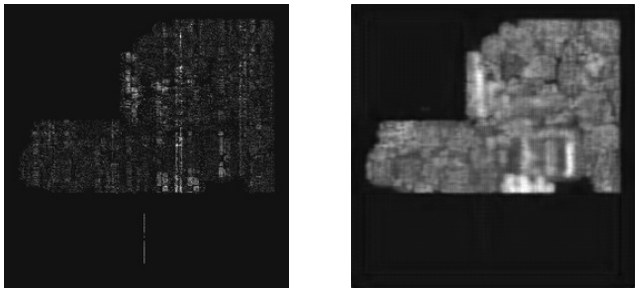
Figure 3: Examples of image features used in the model: macro region, RUDY, and RUDY pin.

The corresponding label for the above three features and the congestion map predicted is demonstrated in 4.

4.3. Data Augmentation

Data augmentation is crucial for enhancing the robustness and generalizability of the model. For our research on routability prediction, we explored several augmentation strategies to assess their impact on model performance.

Flipping: Flipping is studied in [4]. It applied both horizontal and vertical flipping to the input images as a baseline. Flipping helps the model become invariant to the orientation of the layout, which can be particularly useful given that certain circuit designs might exhibit symmetrical properties.



(a) Label (b) Prediction

Figure 4: Examples of label and prediction congestion maps.

Rotation: Images were rotated by angles of 90, 180, and 270 degrees. While rotation can help the model generalize better to different orientations, we observed that excessive rotation could introduce artifacts due to the non-uniform nature of circuit layouts. This effect suggests that some orientations might not be equally valid or prevalent in the dataset, potentially leading to performance degradation.

Translation: We translated the images in both x and y directions by up to 10 pixels. To handle the areas that become vacant due to translation, we filled these regions with black pixels (value 0). This method of handling the translated areas might introduce artificial edges, which could adversely affect the model's learning process. Future work could explore more sophisticated filling techniques to mitigate this issue.

5. Results

We conducted 10 trial training sessions to train our ViT-Hybrid models over 3000 epochs, utilizing Optuna for hyperparameter optimization. The configurable parameters included learning rate, weight decay, and batch size. As shown in Figure 5, the optimal combination of these parameters was found to be a batch size of 32, a learning rate of approximately 3×10^{-4}, and a weight decay of about

5×10^{-6}. Among these, the learning rate emerged as the most critical hyperparameter influencing training accuracy.

Figure 5: Hyperparameter Search with Optuna

Even after training for more than 3000 epochs, our model exhibited overfitting; the training loss continued to decrease, but the validation results began to worsen. Therefore, we chose to limit our experiments to 3000 epochs with Optuna. This decision helped balance the training process and prevent further overfitting.

5.1. Model Configuration

The Vit-Hybrid model was configured with an input size of $3 \times 256 \times 256$, utilizing 8×8 patches, resulting in a total of 1024 patches per image. The latent feature size in the transformer was set to 64, with the MLP layer size at 128. The model architecture included five transformer encoder layers and five TransConv layers. We used Mean Squared Error (MSE) as the loss criterion.

5.2. Evaluation Metrics

We evaluated the model performance using two metrics:

- **Normalized Root Mean Square Error (NRMSE)**: This metric assesses the average magnitude of errors between the predicted and actual values, normalized by the range of the dataset. Lower NRMSE values indicate closer agreement between predictions and the true data, reflecting better model accuracy.
- **Structural Similarity Index Measure (SSIM)**: SSIM evaluates the visual similarity between the predicted and actual images, focusing on aspects like luminance, contrast, and structural integrity. Higher SSIM values suggest better visual quality and accuracy in the predictions.

5.3. Summary of Results

The evaluation results are detailed in Table 3, which compares the performance metrics across different models and the baseline which has random weights.

Table 2: Experimental results on routing congestion prediction

Dataset	Model	NRMSE	SSIM
CircuitNet2.0	Random Weight	0.3868	0.3349
CircuitNet2.0	GPDL	0.0473	0.7691
CircuitNet2.0	ViT-Hybrid	0.0503	0.7751

We achieved results that are very close to those of the GPDL model for both metrics. However, we do not consider this an outperformance of the base model since the difference is not significant. We believe there are several reasons for this outcome:

1. ViT-based models are known to perform best when pre-trained with a backbone model and then fine-tuned for downstream tasks. This approach is particularly beneficial when the available data for downstream tasks is limited compared to datasets like ImageNet used for training backbone models. Our dataset is also quite limited in size (thousands of samples), making the adoption of a pre-trained backbone model advantageous. Although we have implemented the necessary code, we lacked the time and computing resources to fully tune the DINOv2 version of the ViT-Hybrid model to achieve better results.

2. Our findings may also suggest that while DINOv2 excels in tasks such as segmentation, its pre-trained features may not directly translate to improvements in routability prediction due to the unique nature of the problem. This outcome might be attributed to the DINOv2 model being pre-trained on segmentation tasks, which differs from our pixel-level regression task for circuit data. However, due to time constraints, further tuning and adaptations were not explored to confirm this hypothesis.

5.4. Other Attempts

5.4.1 Feature Pyramid Networks

We also explored other FCN-based methods, such as Feature Pyramid Networks (FPN) [7], to enhance the feature representations between encoders and decoders. The code for this approach can be found here. However, this method did not yield significant improvements, so we did not include it in the core method section.

FPN is a relatively large model, and training it took much longer compared to the more compact models like GPDL and ViT-Hybrid. Despite extensive training (up to 40,000 epochs), FPN did not overfit, but it showed only minimal improvement. We believe the limited size of our dataset makes it unsuitable for large models like FPN.

5.4.2 Concatenating Coordinate Information

The ViT model divides image inputs into patches, causing the pixel position information to be lost due to the linear projection embedding layers. We attempted to concatenate normalized coordinate information to the input as two additional channel features (code). However, this approach did not yield significant performance improvement. While exploring this idea, we found [5], which applied the same concept to ViT-based speech emotion recognition with good results.

In data listed in chronological order, time information is essential. In images preprocessed from sound data, the passage of time can be inferred from the x-axis coordinate information, allowing ViT to determine the order of the divided patches from the x- and y-coordinate information. Concatenating the coordinate information directly to the input image retains pixel position information and transfers it to ViT.

We suspect that while this method is beneficial for speech emotion recognition, where the temporal order is crucial, it does not apply as effectively to our domain. In our case, the spatial relationships in the congestion map do not carry the same inherent sequential significance as temporal data does in speech recognition. Thus, the added coordinate information does not provide the same advantage in predicting congestion levels.

5.5. Data Augmentation Analysis

Table 3: Test result for different data augmentations

Data augmentation	NRMSE	SSIM	EMD
With Flip on GPDL	0.0617	0.8182	0.0029
With Rotation on GPDL	0.0635	0.8111	0.0031
With Translation on GPDL	0.0620	0.8146	0.0032
With 3 data aug on GPDL	0.0630	0.8137	0.0033

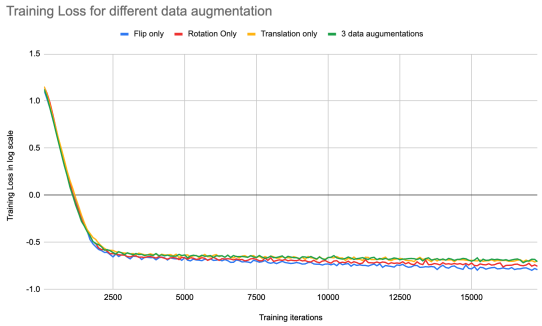


Figure 6: Training loss for different data augmentations

Effects of Augmentation: Our experiments indicated that while augmentation can potentially enhance model robustness, it also led to slower training and slightly worse performance metrics. This outcome may be attributed to the specific characteristics of circuit images, where the spatial relationships and orientations are critical. Translation, in particular, seemed to disrupt the inherent structure of the circuit layouts. Flip augmentation remains the most effective data augmentation technique. Although the initial results showed a slight degradation in performance with more aggressive augmentation, further tuning of augmentation parameters and methods might yield better outcomes.

6. Conclusion and Future Work

6.1. Conclusion

In this work, we demonstrated the potential of Vision Transformers (ViTs) in the domain of routability prediction within electronic design automation (EDA). Our ViT-Hybrid model, combining a ViT encoder for global spatial relationships with a CNN decoder for refined reconstruction, achieved performance comparable to the established GPDL baseline on the CircuiteNet2.0 dataset. This confirms the viability of adapting transformer architectures for specialized EDA tasks.

While our ViT-Hybrid model didn't significantly outperform the GPDL baseline, several factors contributed to this:

- **Limited Dataset Size:**: The small dataset size typical of chip design (in the thousands) might not be sufficient for the ViT to fully leverage its ability to learn complex patterns.
- **Transfer Learning Challenges**:Our preliminary experiments with DINOv2-based transfer learning showed promise but didn't yield substantial gains. The pre-trained DINOv2 features, optimized for segmentation, might not be ideally aligned with the nuances of routability prediction.
- **Potential for Inaccurate Topology Representation**:In highly congested layouts with complex topological relationships, our model might struggle to accurately represent the underlying circuit structure, or represent long-range effects. This could lead to less precise predictions of routability issues.

6.2. Future Directions

With additional resources and time, we would explore several avenues to enhance our approach:

- **Detailed Result Analysis**: We would conduct a thorough analysis of the model's performance on individual samples, identifying specific design characteristics or circuit features that lead to higher or lower accuracy. This would involve examining the model's predictions

at a macro and pixel level to pinpoint areas of strength and weakness.

- **Deeper Transfer Learning Integration**: More in-depth exploration of the DINOv2 features, possibly involving fine-tuning of the DINOv2 layers themselves, could lead to better utilization of the pre-trained knowledge. We would also investigate alternative pre-trained models with task types more closely aligned with routability prediction, such as models designed for image-to-image translation or regression. This might offer better transferability and enhance the effectiveness of transfer learning.

- **Hybrid Architectures**:Combining graph neural networks (GNNs) with our ViT-Hybrid model could capture both geometric and topological relationships, leading to a more comprehensive understanding of routability challenges.

- **Refine Data Augmentation**: Developing more sophisticated data augmentation strategies tailored to circuit layouts could enhance model robustness and generalization.

- **Real-World Validation**:Evaluating the model on real-world industrial chip designs would provide invaluable insights into its practical utility and potential for real-world impact.

By addressing these aspects, we believe that transfer-learning-enhanced, transformer-based models hold significant promise for advancing routability prediction and accelerating the EDA design process.

References

[1] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[2] Z. Chai, Y. Zhao, W. Liu, Y. Lin, R. Wang, and R. Huang. Circuitnet: An open-source dataset for machine learning in vlsi cad applications with improved domain-specific evaluation metric and learning strategies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(12):5034–5047, 2023.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] X. Jiang, Z. Chai, Y. Zhao, Y. Lin, R. Wang, and R. Huang. Circuitnet 2.0: An advanced dataset for promoting machine learning innovations in realistic chip design environment. In *Proceedings of the International Conference on Learning Representations*, Virtual Conference, 2024. Poster session.

[5] J.-Y. Kim and S.-H. Lee. Coordvit: A novel method of improve vision transformer-based speech emotion recognition using coordinate information concatenate. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4, 2023.

[6] R. Kirby, S. Godil, R. Roy, and B. Catanzaro. Congestionnet: Routing congestion prediction using deep graph neural networks. *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 217–222, 2019.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection, 2017.

[8] S. Liu, Q. Sun, P. Liao, Y. Lin, and B. Yu. Global placement with deep learning-enabled explicit routability optimization. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1821–1824, 2021.

[9] K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio. Routenet: Leveraging graph neural networks for network modeling and optimization in sdn. *IEEE Journal on Selected Areas in Communications*, 38(10):2260–2270, Oct. 2020.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[11] B. Wang, G. Shen, D. Li, J. Hao, W. Liu, Y. Huang, H. Wu, Y. Lin, G. Chen, and P. A. Heng. Lhnn: Lattice hypergraph neural network for vlsi congestion prediction, 2022.

[12] Y. Zhao, Z. Chai, Y. Lin, R. Wang, and R. Huang. Hybridnet: Dual-branch fusion of geometrical and topological views for vlsi congestion prediction, 2023.