# Enhancing HLS Performance Prediction on FPGAs through Multi-Modal Representation Learning

Longshan Shang, Teng Wang*, Lei Gong*, Chao Wang, Xuehai Zhou
*Suzhou Institute for Advanced Research, University of Science and Technology of China*
Suzhou, Jiangsu, China
{tantu}@mail.ustc.edu.cn, {wangt635, leigong0203, cswang, xhzhou}@ustc.edu.cn

***Abstract*—The emergence of Design Space Exploration (DSE) technology has reduced the cost of searching for pragma configurations that lead to optimal performance microarchitecture. However, obtaining synthesis reports for a single design candidate can be time-consuming, sometimes taking several hours or even tens of hours, rendering this process prohibitively expensive. Researchers have proposed many solutions to address this issue. Previous studies have focused on extracting features from a single modality, leading to challenges in comprehensively evaluating the quality of designs. To overcome this limitation, this paper introduces a novel modal-aware representation learning method for the evaluation of HLS design, named MORPH, which integrates information from three data modalities to characterize High-level synthesis (HLS) designs, including code, graph, and code description (caption) modality. Remarkably, our model outperforms the baseline, demonstrating a 6%-25% improvement in RMSE loss. Moreover, the transferability of our predictor has also been notably enhanced.**

***Index Terms*—HLS, Multi-modality, Design Space Exploration**

## I. Introduction

With the termination of Dennard scaling [1], Field-Programmable Gate Arrays (FPGAs) have emerged as a potential choice for data center accelerators due to their reconfigurability and energy-efficient characteristics [2]–[4]. However, their steep learning curve has hindered widespread adoption. The advent of HLS [5], [6] has alleviated the limitations of FPGAs by translating high-level code into hardware description languages and efficiently restructuring underlying microarchitectures through the insertion of pragmas [7]. However, it requires repetitive trials along with time-consuming synthesis to identify efficient pragma combinations. Therefore, an efficient and accurate prediction model is urgently needed to evaluate the current design point during DSE.

Some previous studies treated HLS tools as black boxes and focused on developing efficient heuristic methods [8]–[11] to minimize synthesis times. Despite these efforts, the DSE process is still time-consuming. Other research introduced surrogate models as alternatives to HLS tools. For instance, some studies utilized analytical models [12], [13], while others represented HLS designs as graphs and employed graph neural networks (GNNs) for quality prediction [14], [15]. However, despite the promising performance demonstrated by GNNs, they have been limited to unimodal data, hindering the exploitation of all potentially beneficial information for prediction, thereby impacting accuracy.

To address the aforementioned challenges, we propose the Modal-Aware Representation Learning model for HLS, referred to as MORPH, to quickly and accurately evaluate candidate design points generated during the HLS DSE process. The core of MORPH is Graph, Code, Caption transformer (GCC-Former) which utilized contrastive learning [16] and cross-attention mechanisms [17] to align the multi-modal features produced by encoders to generate robust embeddings that can be used for downstream prediction tasks. The experimental results demonstrate that our model architecture significantly improves prediction performance.

In summary, in this paper, we make the following contributions:

- We propose MORPH to leverage information from various modalities, enhancing the performance of design point evaluation in the DSE process to address issues of time consumption and low accuracy in the evaluation process.
- We introduce GCC-Former as an essential module in MORPH to bridge the gap between graph, code, and caption for robust feature extraction. It is a staged pre-trained query transformer consisting of representation learning and prediction learning stages.
- The experimental results demonstrate that, compared to the SOTA HARP [15] method, our approach significantly reduced the RMSE loss by 6%-25%. Additionally, through transfer learning experiments, we validated the adaptability of our proposed model architecture to various versions of HLS tools.
- Compared to HARP, our method increased the average design point performance by 21% and 11% on two datasets from varying versions of HLS tools in design space exploration experiments due to its high prediction accuracy.

## II. Related Work

As machine learning advances, it's increasingly applied across Electronic Design Automation (EDA) stages like high-level synthesis [19], logic synthesis [20], floorplanning and placement [21]. Specifically, in HLS, there is a growing trend of incorporating advanced deep learning techniques into automatic optimization processes, like leveraging reinforcement

[*] Correspondence authors are Teng Wang, Lei Gong

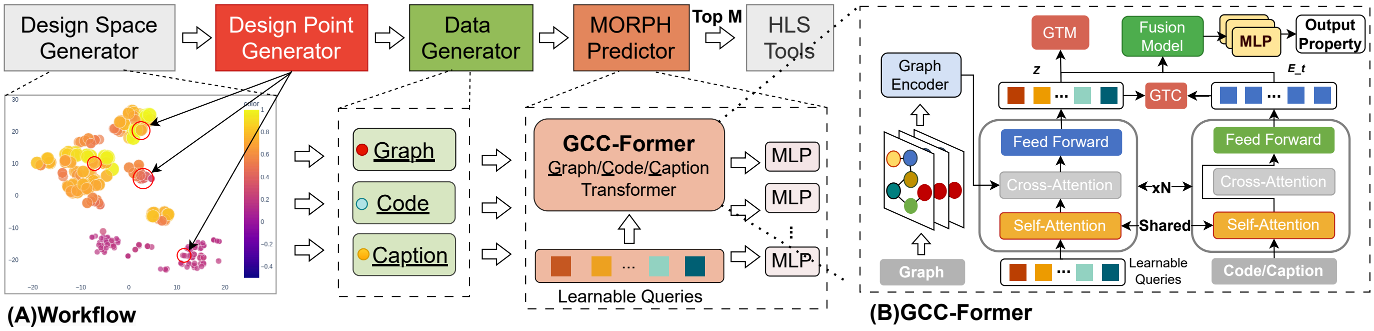


Fig. 1. Overview of MORPH: (A)The processing order of our method. (B)Model architecture of MORPH.

learning for design space exploration [22] or using machine learning algorithms to model HLS tools. One promising approach involves processing Control Data Flow Graphs (CDFGs) of input designs by employing Graph Neural Networks (GNNs) [14], [15] as surrogate of the HLS tools which can predict the latency and resource utilization for BRAM, DSP, FF, and LUT. Previous research on surrogate models predominantly utilized unimodal data for quality prediction. Although optimizing HLS designs' graph representations can improve prediction performance, unimodal information is not comprehensive enough to meet the demand for accurately predicting current design point performance during DSE. This limitation becomes apparent when facing the challenges posed by task shifts.

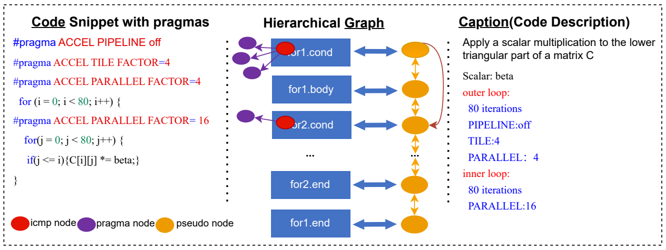
## III. MORPH Methodology

In this study, we introduce MORPH, a multi-modal model aimed at efficiently and accurately evaluating design points during HLS DSE. Inspired by Q-Fromer architecture in BLIP2 [18], we proposed GCC-Former to effectively align multiple modalities into a unified latent space, providing a comprehensive view of HLS designs.

### A. Multi-modality Data

The diversity of data modalities allows models to capture global information about HLS design, thereby enhancing representational capabilities and predictive performance. In our work, we utilized information from three modalities: graph, code, and captions.

For graph modality, we adopt a hierarchical representation augmented with pragma nodes proposed in HARP [15] to provide control and data flow information. This approach abstracts LLVM [23] Blocks into high-level pseudo-nodes to mitigate long-range dependency issues. We retrained HARP to serve as a graph encoder for encoding graph modality. For code modality, we simply replace the pragma placeholders in the original code with configuration options. Finally, since pragmas constitute only a small part of the overall design, it is challenging to extract the influence of pragmas. Therefore, we utilize TongyiLingma, a code generation model, to extract code structure, key parameters, and pragmas in order to balance the code and pragma ratio. To better suit the HLS design optimization scenario, we utilized the Jina-embeddings [24] model, which is specifically designed for code text to encode data from code and caption modalities. The multi-modality data format is illustrated in Fig 2.



```
#pragma ACCEL PIPELINE off
#pragma ACCEL TILE FACTOR=4
#pragma ACCEL PARALLEL FACTOR=4
 for (i = 0; i < 80; i++) {
#pragma ACCEL PARALLEL FACTOR= 16
  for(j = 0; j < 80; j++) {
   if(j <= i){C[i][j] *= beta;}
}
```

Apply a scalar multiplication to the lower triangular part of a matrix C

Scalar: beta
outer loop:
  80 iterations
  PIPELINE:off
  TILE:4
  PARALLEL： 4
inner loop:
  80 iterations
  PARALLEL:16

Fig. 2. A toy example of our multi-modality data

### B. GCC-Former

Inspired by BLIP2, we propose the GCC-Former, a transformer architecture designed to align data from different modalities. Although current multi-modal models primarily focus on visual and language tasks, our GCC-Former leverages information from graph and text modalities, as illustrated in Fig 1 (B). The GCC-Former consists of two sub-modules with shared self-attention layers: (1) a graph transformer, which interacts with a frozen graph encoder to extract graph features, and (2) a text transformer which acts as a text encoder. Moreover, we create a set of learnable queries as input to the graph transformer, these queries can interact with each other through self-attention layers, and interact with frozen graph features through cross-attention layers (inserted every other transformer block). The queries can additionally interact with the text through the same self-attention layers. The GCC-Former was trained in two stages: (1) the representation learning phase and (2) the predictor training phase. The representation learning stage utilizes attention mechanisms and contrastive learning methods to align data from diverse modalities. Subsequently, during the predictor training phase, we create a Multi-layer Perceptron (MLP) for each prediction target and train it on our database.

### C. Representation Learning

In the representation learning phase, we froze the graph encoder. Subsequently, the trainable GCC-Former employs self-supervised learning techniques, including matching learning and contrastive learning, as well as cross-attention mechanisms to bridge the modality gap.

**Graph-Text Contrastive Learning (GTC)** aims to align graph representations with text representations to maximize

TABLE I
TOTAL ROOT MEAN SQUARED ERROR (RMSE), MEAN ABSOLUTE ERROR (MAE), AND PERF RANKING (TAU) OF THE MODELS.

| **Name** | **Model** | **Modal** | **v1 database** | | | **v2 database** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ***Train from scratch*** | | | ***Train from scratch*** | | | ***Fine-tuned from v1*** | | |
| | | | *RMSE* | *MAE* | *perf tau* | *RMSE* | *MAE* | *perf tau* | *RMSE* | *MAE* | *perf tau* |
| M1 | HARP | graph | 0.6930 | 0.2746 | 0.8629 | 0.6386 | 0.2808 | 0.8375 | 0.6239 | 0.2729 | 0.7950 |
| M2 | Concate_Fusion | graph+code | 0.5911<br>(-14%) | 0.2519<br>(-8%) | 0.8250<br>(-4%) | 0.6628<br>(+4%) | 0.2953<br>(+5%) | 0.7841<br>(-6%) | 0.7739<br>(+24%) | 0.3138<br>(+15%) | 0.7840<br>(-1%) |
| M3 | Weight_Fusion | graph+code | 0.6066<br>(-12%) | 0.2645<br>(-4%) | 0.8163<br>(-5%) | 0.6644<br>(+4%) | 0.3035<br>(+8%) | 0.7833<br>(-6%) | 0.6404<br>(+3%) | 0.3063<br>(+12%) | 0.7761<br>(-2%) |
| M4 | MORPH_2modal | graph+code | 0.5623<br>(-19%) | 0.2440<br>(-11%) | 0.8714<br>(+1%) | 0.5695<br>(-11%) | 0.2732<br>(-3%) | 0.8359<br>(+0%) | 0.6119<br>(-2%) | 0.2724<br>(-0%) | 0.8104<br>(+2%) |
| M5 | MORPH_3modal | graph+code+caption | 0.5188<br>(-25%) | 0.2087<br>(-24%) | 0.8960<br>(+4%) | 0.5627<br>(-12%) | 0.2285<br>(-19%) | 0.8628<br>(+3%) | 0.5878<br>(-6%) | 0.2473<br>(-9%) | 0.8350<br>(+5%) |

their mutual information. Our training strategy generates positive and negative graph-text pairs within each batch through a permutation mechanism. In this work, we employ 32 learnable queries where each query has a dimension of 768, so we calculate the similarity between each query's output and the output text embedding $E_t$, selecting the highest value as the similarity of each graph-text pair. Learnable queries and text tokens are simultaneously input into a shared self-attention layer. To prevent information leakage, we employ an unimodal self-attention mask, ensuring that queries and text cannot attend to each other. We calculate $I(E_z, E_t)$ and $I(E_t, E_z)$ and compute the losses individually, taking the average as the loss. The formula is shown as follows:

$$I(E_i, E_j) = E_i \cdot E_j \tag{1}$$

$$Loss_{gtc} = \frac{1}{2} * (\sum cross\_entropy(I_i, y)) \tag{2}$$

**Graph-Text Matching (GTM)** aims to learn the fine-grained alignment between graphs and textual data, where textual data refers to data in the form of code and caption. GTM also generates positive and negative graph-text pairs in each batch. Utilizing bidirectional self-attention masks, the model facilitates mutual attentiveness between learnable queries and textual content, enabling the output query embeddings $Z$ to capture multimodal information comprehensively. These embeddings are subsequently fed into a binary classification linear layer to compute matching scores for different queries. The average of these scores is then utilized to calculate the cross-entropy loss against the true labels.

## IV. EXPERIMENTAL RESULTS

### A. Experiment Setup

Our study utilized the HARP Database [15] consisting of 40 kernels of varying complexities selected from the Machsuite benchmark and Polyhedral benchmark. The dataset comprises quality reports obtained from two versions of HLS tools (SDAccel 2018.3 and Vitis 2020.2) for various design points of different kernels. These reports encompassed DSP/BRAM/LUT/FF utilization and latency measured in cycle counts. Specifically, the datasets from the two versions of the HLS tool are denoted as the v1 and v2 databases, respectively. To deploy and train our framework, PyTorch was utilized on the NVIDIA Tesla A100 GPU.

### B. Model Accuracy

To validate our approach, we retrained HARP using the same database and hyperparameters as in this paper as baseline (M1). Subsequently, we explored two Early Fusion methods: (1)simply concatenating the embeddings of different modalities (M2) and using a learnable weight to aggregate different embeddings (M3). Finally, experiments were conducted with our proposed model for both bimodal (M4) and trimodal (M5) scenarios. Each model was trained from scratch on datasets v1 and v2 and then proceeded to fine-tune the v1 model on the v2 dataset. For transfer learning fine-tuning, we trained for an additional 200 epochs.

Table I illustrates the performance of each model on the test set. Kendall's tau metric is used to gauge the similarity between predicted rankings of design points and their ground truth rankings. It is observed that early fusion methods (M2, M3) show reduced RMSE loss on the v1 dataset. However, it performs worse on the v2 dataset, whether trained from scratch or fine-tuned. In contrast, our proposed MORPH approach, both M4 and M5, significantly reduces RMSE/MAE loss across all three scenarios and shows marked improvement in the tau metric. Furthermore, when fine-tuning the v1 model on the v2 dataset, it achieves similar performance compared to training from scratch with just a few training epochs. This demonstrates the transferability of our approach, alleviating the need to repeatedly generate large datasets for various HLS tools.

TABLE II
PERFORMANCE OF BEST DESIGN FOUND BY DSE

| Approach | **Time Limit** | **v1 kernels** | | **v2 kernels** | |
|---|---|---|---|---|---|
| | | ***avg*** | ***geo mean*** | ***avg*** | ***geo mean*** |
| GNN-DSE | 1.5h/kernel | 1x | 1x | 1x | 1x |
| HARP | 1.5h/kernel | 1.07x | 1.28x | 1.34x | 1.37x |
| [Ours] | 1.5h/kernel | 1.29x | 1.33x | 1.49x | 1.51x |

### C. Performance Results

To demonstrate the benefits of our predictor for DSE, we utilized the HARP's depth-first search method [15] for design space exploration to find low-latency design points that meet resource constraints (BRAM, DSP, FF, LUT), while evaluating candidate design points during DSE using different predictors. To facilitate this process, a classification model was employed to prune ineffective design points. The classifier first determines the effectiveness of the current design point and then a regression model was used to evaluate the quality of the current design point. We evaluated the assessment time for all kernels, with an average evaluation time of 50ms per design point. We set the DSE time limit to 1.5 hours per kernel, which allows us to explore approximately 108,000 design points within this time frame. Following the completion of DSE, we synthesize the top ten design points to get their true performance for comparison.

As shown in Table II, our proposed MORPH for DSE achieved significant improvements in both the average and geometric mean of the optimal perf values for 35 kernels in dataset v1 and 27 kernels in dataset v2. We present the performance improvement factors of HARP and our MORPH model relative to GNN-DSE. Compared to the state-of-the-art HARP, our method achieved an average performance improvement of 21% on v1 kernels and a 11% acceleration on v2 kernels.

## V. Conclusion

In this work, we developed a multi-modal surrogate model for High-Level Synthesis (HLS) to address the challenges of long evaluation time and low evaluation accuracy in HLS design point assessment. We proposed the GCC-Former module, which aligns and integrates information from multiple modalities using contrastive learning and cross-attention mechanisms to enhance the model's performance. In the future, we intend to incorporate reinforcement learning techniques to enable efficient automatic design space exploration in HLS.

## Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grants 2022YFB4501600 and 2022YFB4501603, in part by the National Natural Science Foundation of China under Grants 62102383, 61976200, and 62172380, in part by Jiangsu Provincial Natural Science Foundation under Grant BK20210123, in part by Youth Innovation Promotion Association CAS under Grant Y2021121.

## References

[1] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, ”Design of ion-implanted MOSFET's with very small physical dimensions,” IEEE Journal of Solid-State Circuits, vol. 9, no. 5, pp. 256–268, 1974.

[2] C. Wang, L. Gong, X. Li and X. Zhou, ”A Ubiquitous Machine Learning Accelerator With Automatic Parallelization on FPGA.” in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 10, pp. 2346-2359, 1 Oct. 2020.

[3] D. Danopoulos, K. Anagnostopoulos, C. Kachris and D. Soudris, ”FPGA Acceleration of Generative Adversarial Networks for Image Reconstruction,” 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 2021, pp. 1-5.

[4] C. Wang, L. Gong, X. Ma, X. Li and X. Zhou, ”WooKong: A Ubiquitous Accelerator for Recommendation Algorithms With Custom Instruction Sets on FPGA.” in IEEE Transactions on Computers, vol. 69, no. 7, pp. 1071-1082, 1 July 2020.

[5] Cong, Jason, et al. “High-Level Synthesis for FPGAs: From Prototyping to Deployment.” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Apr. 2011, pp. 473–91.

[6] G. Martin and G. Smith, ”High-Level Synthesis: Past, Present, and Future,” in IEEE Design and Test of Computers, vol. 26, no. 4, pp. 18-25, July-Aug. 2009.

[7] Cong, Jason, et al. “FPGA HLS Today: Successes, Challenges, and Opportunities.” ACM Transactions on Reconfigurable Technology and Systems, Dec. 2022, pp. 1–42.

[8] C. H. Yu, P. Wei, M. Grossman, P. Zhang, V. Sarker and J. Cong, ”S2FA: An Accelerator Automation Framework for Heterogeneous Computing in Datacenters,” 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2018, pp. 1-6.

[9] L. Ferretti, G. Ansaloni and L. Pozzi, ”Lattice-Traversing Design Space Exploration for High Level Synthesis,ICCD, Orlando, FL, USA, 2018, pp. 210-217.

[10] Sohrabizadeh, A., Yu, C.H., Gao, M. and Cong, J., 2022. AutoDSE: Enabling software programmers to design efficient FPGA accelerators. ACM Transactions on Design Automation of Electronic Systems (TODAES), 27(4), pp.1-27.

[11] Sun, Q., Chen, T., Liu, S., Chen, J., Yu, H. and Yu, B., 2022. Correlated multi-objective multi-fidelity optimization for HLS directives design. ACM Transactions on Design Automation of Electronic Systems (TODAES), 27(4), pp.1-27.

[12] Zhao, Jieru, Liang Feng, Sharad Sinha, Wei Zhang, Yun Liang, and Bingsheng He. ”COMBA: A comprehensive model-based analysis framework for high level synthesis of real applications.” In 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 430-437. IEEE, 2017.

[13] Zhong, Guanwen, Alok Prakash, Yun Liang, Tulika Mitra, and Smail Niar. ”Lin-analyzer: A high-level performance analysis tool for FPGA-based accelerators.” In Proceedings of the 53rd Annual Design Automation Conference, pp. 1-6. 2016.

[14] Sohrabizadeh, A., Bai, Y., Sun, Y. and Cong, J., 2022, July. Automated accelerator optimization aided by graph neural networks. In Proceedings of the 59th ACM/IEEE Design Automation Conference (pp. 55-60).

[15] Sohrabizadeh, A., Bai, Y., Sun, Y. and Cong, J., 2023, October. Robust GNN-based representation learning for HLS. In 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) (pp. 1-9). IEEE.

[16] Khan, Adnan, Sarah AlBarri, and Muhammad Arslan Manzoor. ”Contrastive self-supervised learning: a survey on different architectures.” In 2022 2nd International Conference on Artificial Intelligence (ICAI), pp. 1-6. IEEE, 2022.

[17] Wei, Xi, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. ”Multi-modality cross attention network for image and sentence matching.” In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10941-10950. 2020.

[18] Li, J., Li, D., Savarese, S. and Hoi, S., 2023, July. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning (pp. 19730-19742). PMLR.

[19] Kim, R.G., Doppa, J.R. and Pande, P.P., 2018, November. Machine learning for design space exploration and optimization of manycore systems. In 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-6). IEEE.

[20] Haaswijk, W., Collins, E., Seguin, B., Soeken, M., Kaplan, F., Süsstrunk, S. and De Micheli, G., 2018, May. Deep learning for logic optimization algorithms. In 2018 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-4). IEEE.

[21] Xie, Z., Huang, Y.H., Fang, G.Q., Ren, H., Fang, S.Y., Chen, Y. and Hu, J., 2018, November. RouteNet: Routability prediction for mixed-size designs using convolutional neural network. In 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE.

[22] Wu, N., Xie, Y. and Hao, C., 2021, June. Ironman: Gnn-assisted design space exploration in high-level synthesis via reinforcement learning. In Proceedings of the 2021 on Great Lakes Symposium on VLSI (pp. 39-44).

[23] C. Lattner and V. Adve, ”LLVM: a compilation framework for lifelong program analysis and transformation,” International Symposium on Code Generation and Optimization, 2004. CGO 2004., San Jose, CA, USA, 2004, pp. 75-86

[24] Günther, Michael, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. ”Jina embeddings: A novel set of high-performance sentence embedding models.”