CATE: Computation-aware Neural Architecture Encoding with Transformers

Shen Yan[1], Kaiqiang Song[2,3], Fei Liu[3], Mi Zhang[1]

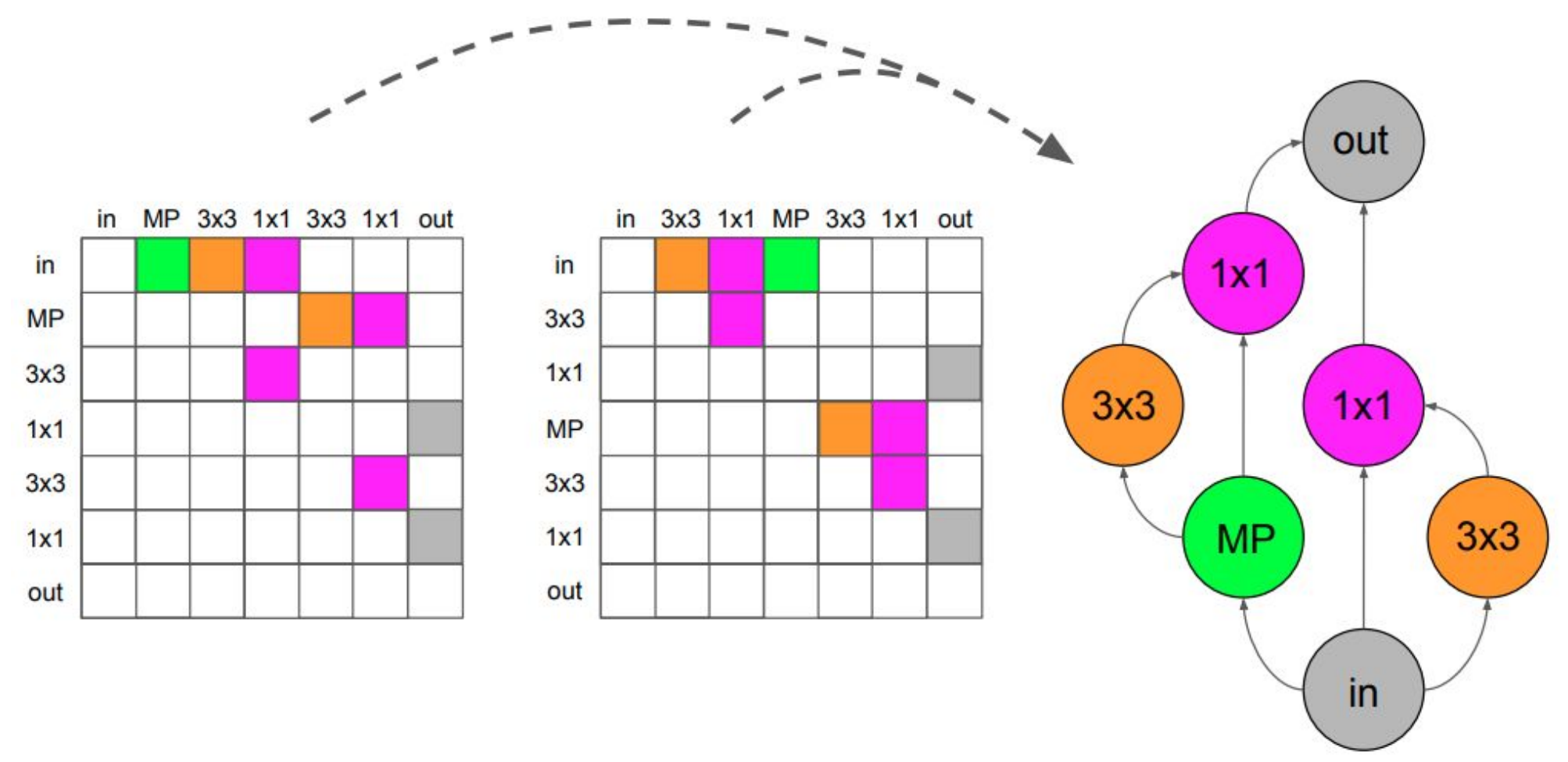
Michigan State University, Tencent AI Lab, University of Central Florida

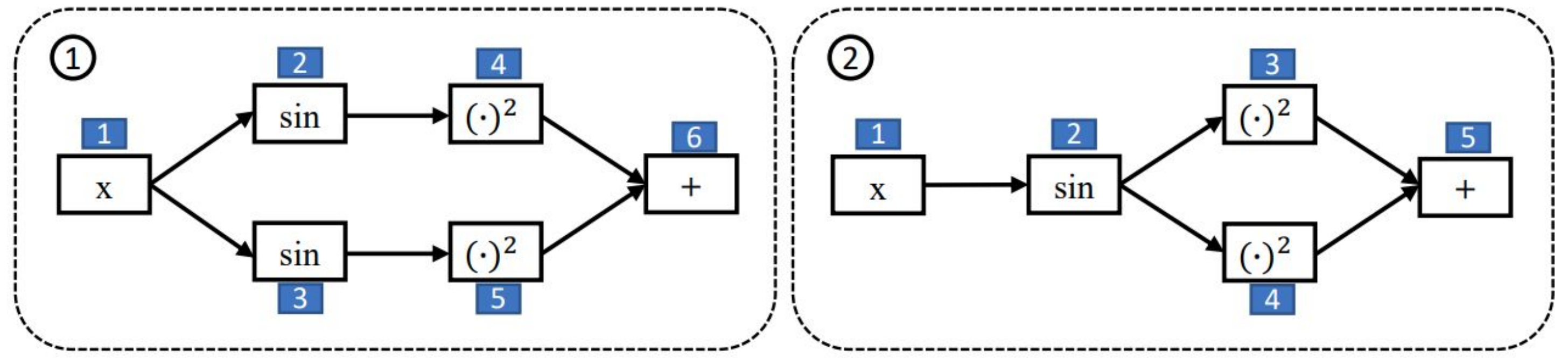
Introduction

Neural Architecture Search has recently drawn considerable attention. While majority of the prior work focuses on either constructing new search spaces or designing efficient search and evaluation methods, some of the most recent work [1,2] sheds light on the importance of ***architecture encoding*** on the subroutines in the NAS pipeline as well as overall performance of NAS.

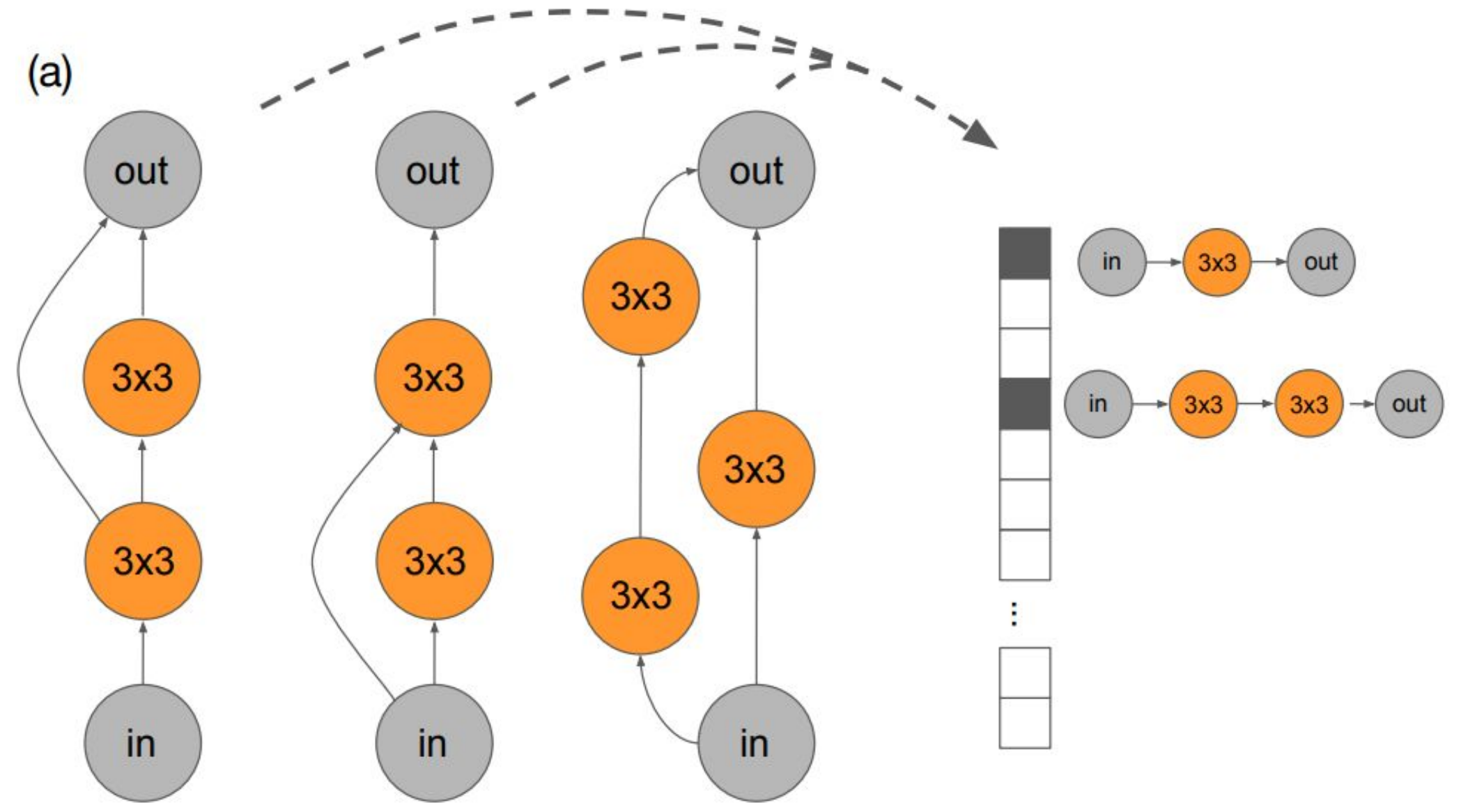
Background

- Structure-aware encodings such as adjacency matrix-based encodings ***may not be computationally unique***. A same encoding can have many different representations using adjacency matrix.



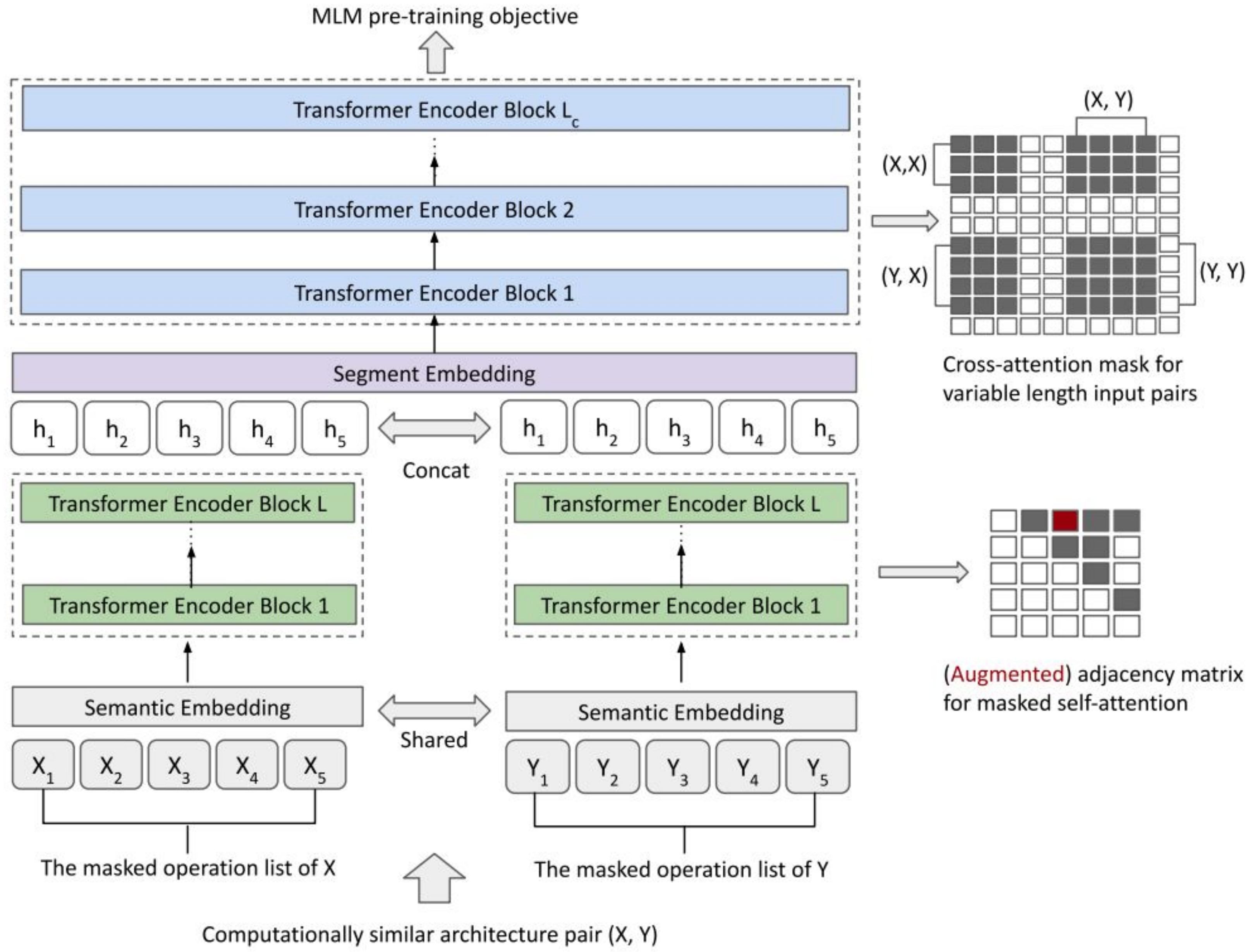


- **Computation-aware encodings** [2,3] are proposed to tackle the above drawbacks. However, it suffers scalability and generalization issues.



Our proposed method: CATE

- Use **computationally similar architecture pairs** as input. The model is trained to predict masked operators **given the joint information**. Extracted architecture encodings are used for the downstream encoding-dependent NAS subroutines.


MLM pre-training objective
Transformer Encoder Block L
Transformer Encoder Block 2
Transformer Encoder Block 1
Segment Embedding
h_1 h_2 h_3 h_4 h_5
Concat
Transformer Encoder Block L
Transformer Encoder Block 1
Semantic Embedding
Shared
X_1 X_2 X_3 X_4 X_5
Y_1 Y_2 Y_3 Y_4 Y_5
The masked operation list of X
The masked operation list of Y
Computationally similar architecture pair (X, Y)
(X, Y)
(X,X)
(Y, X)
(Y, Y)
Cross-attention mask for variable length input pairs
(Augmented) adjacency matrix for masked self-attention

Key difference with BERT:

- **No contextual bias** in the prediction as long as it is a valid graph

  Conv 3x3 -> Conv 5x5 -> ~~Conv 1x1~~ -> Max Pool -> …

  Conv 3x3 -> Conv 5x5 -> ? -> Max Pool -> …

- **Cannot use the fully-visible attention mask** as it does not reflect the single directional flow (e.g. direct, acyclic, single-in-single-out)

Algorithm 1 Floyd Algorithm

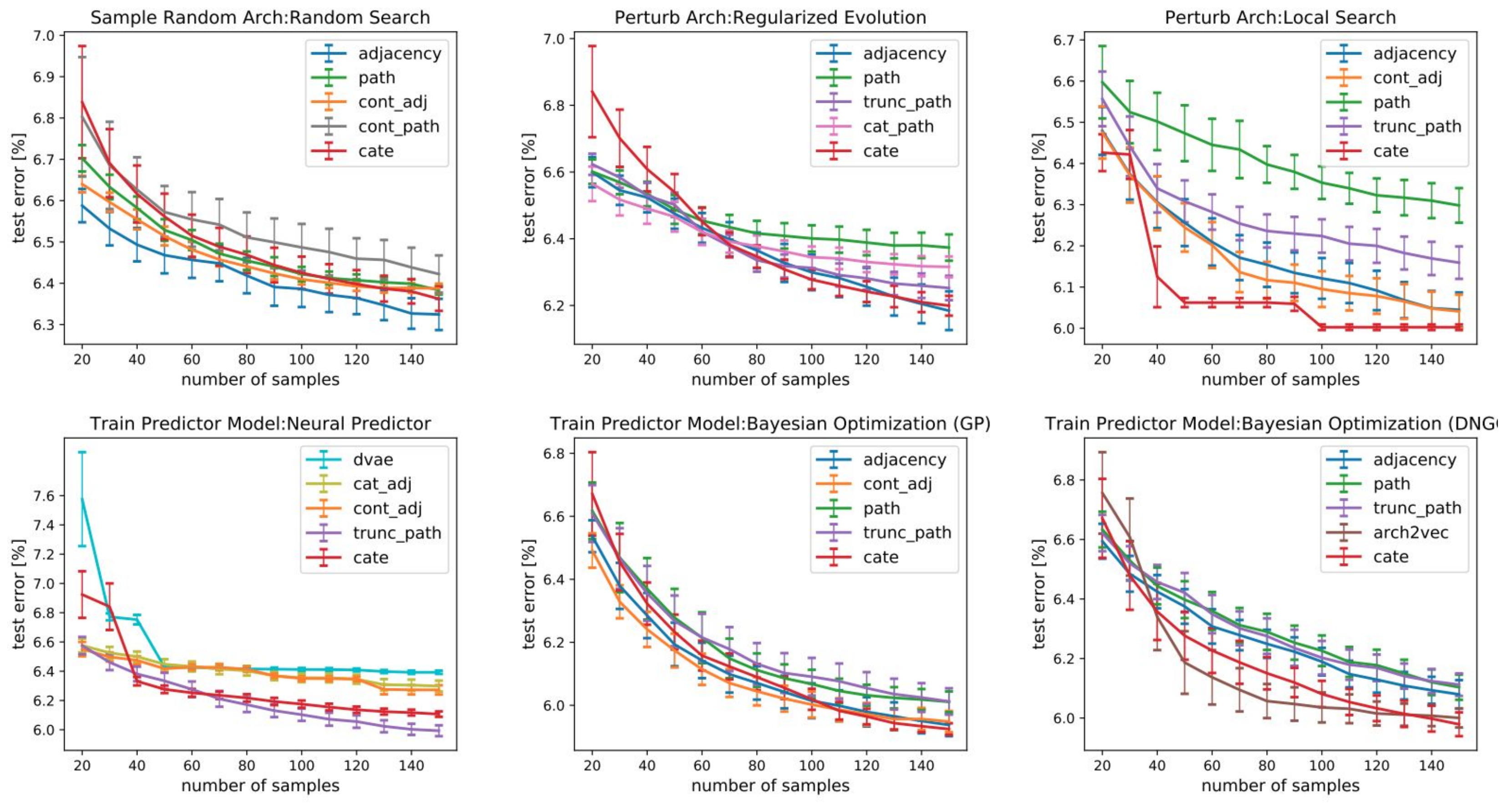
```
Input: the node set $\mathcal{V}$, the adjacent matrix $\mathbf{A}$
$\tilde{\mathbf{A}} \leftarrow \mathbf{A}$
for $k \in \mathcal{V}$ do
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{V}$ do
            $\tilde{\mathbf{A}}_{i,j} \quad |= \tilde{\mathbf{A}}_{i,k} \quad \& \quad \tilde{\mathbf{A}}_{k,j}$
Output: $\tilde{\mathbf{A}}$
```

$$\mathbf{M}_{i,j}^{Direct} = \begin{cases} 0, & \textit{if} \quad A_{i,j} = 1 \\ -\infty, & \textit{if} \quad A_{i,j} = 0 \end{cases}$$

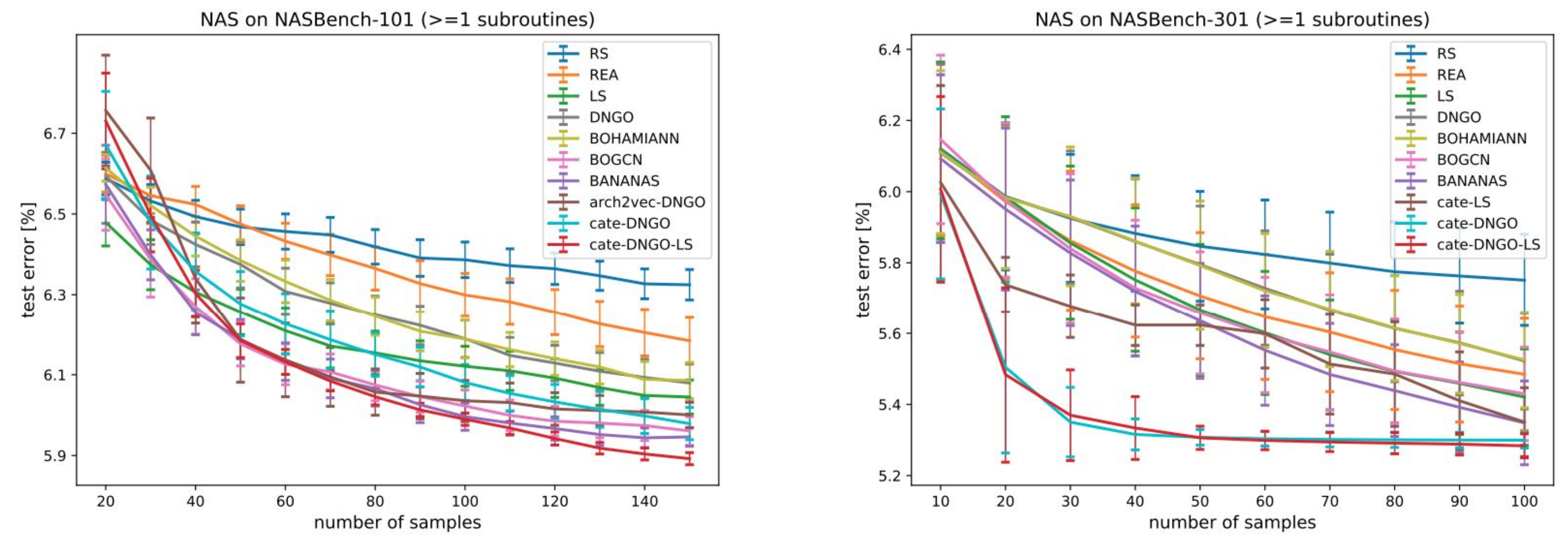
$$\mathbf{M}_{i,j}^{Indirect} = \begin{cases} 0, & \textit{if} \quad \tilde{A}_{i,j} = 1 \\ -\infty, & \textit{if} \quad \tilde{A}_{i,j} = 0 \end{cases}$$

Results

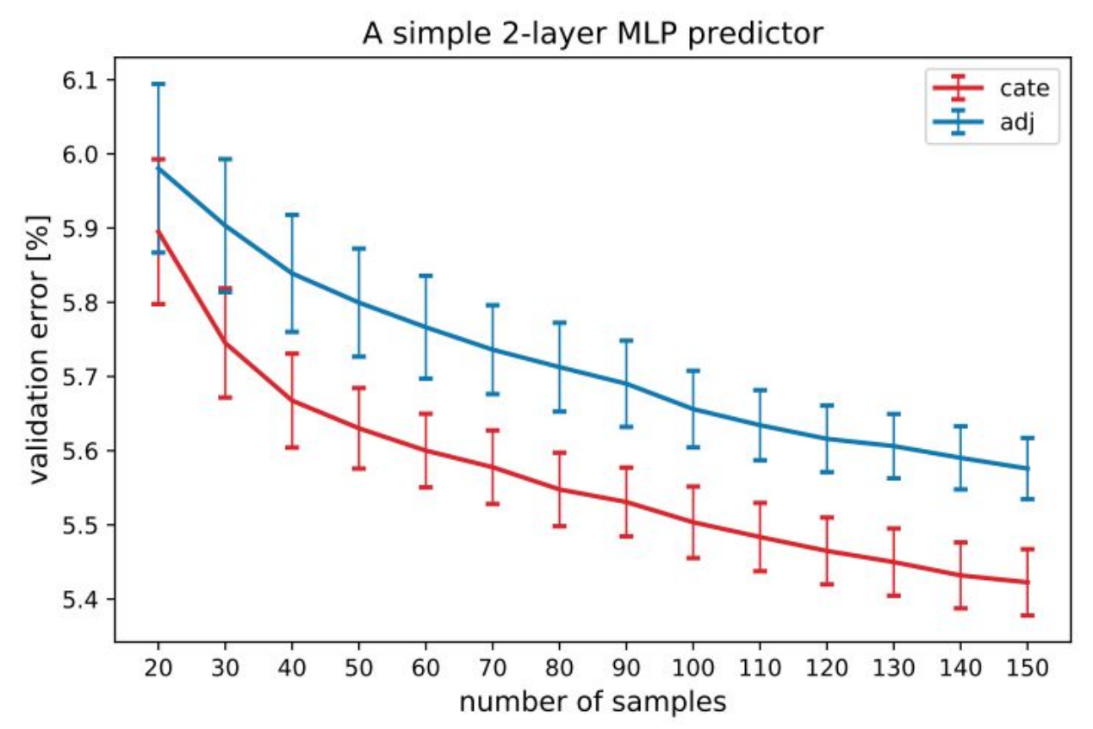
- Comparison between CATE and **other encodings under different NAS subroutines** on NAS-Bench-101:



- Comparison between CATE with **other NAS methods** on NAS-Bench-101 (left) and NAS-Bench-301 (right):



- Generalization performance on **out-of-training search space:**



NAS Methods	Avg. Test Error (%)	Params (M)	Search Cost (GPU days)
RS (Li & Talwalkar, 2019)	3.29 ± 0.15	**3.2**	4
DARTS (Liu et al., 2019a)	2.76 ± 0.09	3.3	4
BANANAS (White et al., 2021)	2.67 ± 0.07	3.6	11.8
arch2vec-BO (Yan et al., 2020)	2.56 ± 0.05	3.6	9.2
CATE-DNGO-LS (small budget)	2.55 ± 0.08	3.5	**3.3**
CATE-DNGO-LS (large budget)	$\mathbf{2.46 \pm 0.05}$	4.1	10.3

[1] Does Unsupervised Architecture Representation Learning Help Neural Architecture Search? Yan et. al., NeurIPS 2020.
[2] A Study on Encodings for Neural Architecture Search. White et. al., NeurIPS 2020.
[3] D-VAE: A Variational Autoencoder for Directed Acyclic Graphs. Zhang et. al., NeurIPS 2019.