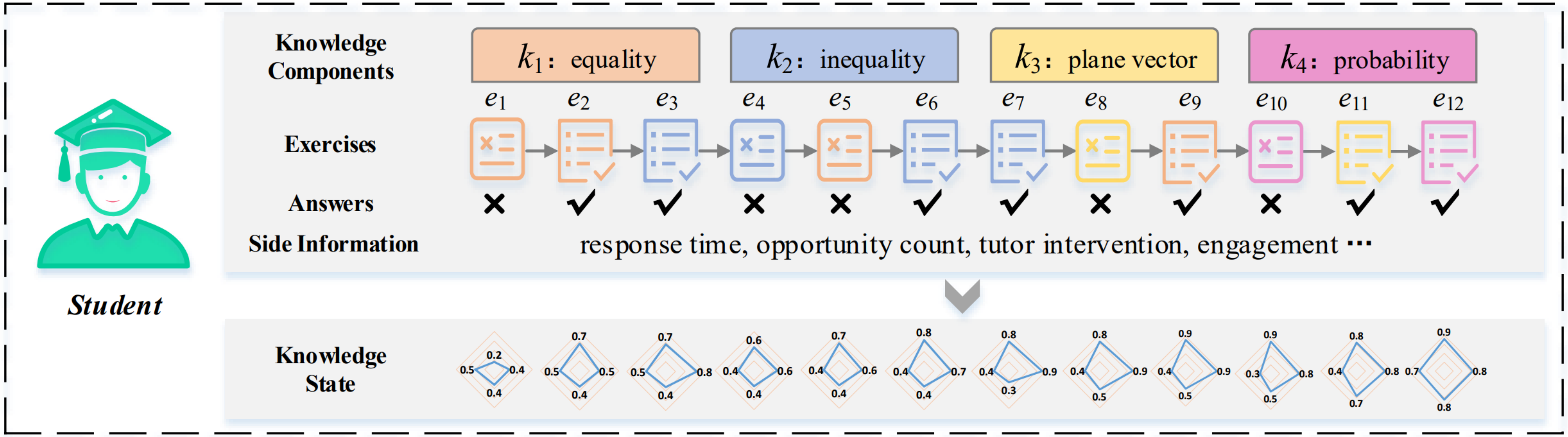
# Evolutionary Neural Architecture Search for Transformer in Knowledge Tracing

Presenter：Shangshang Yang

# ●Background- KT task

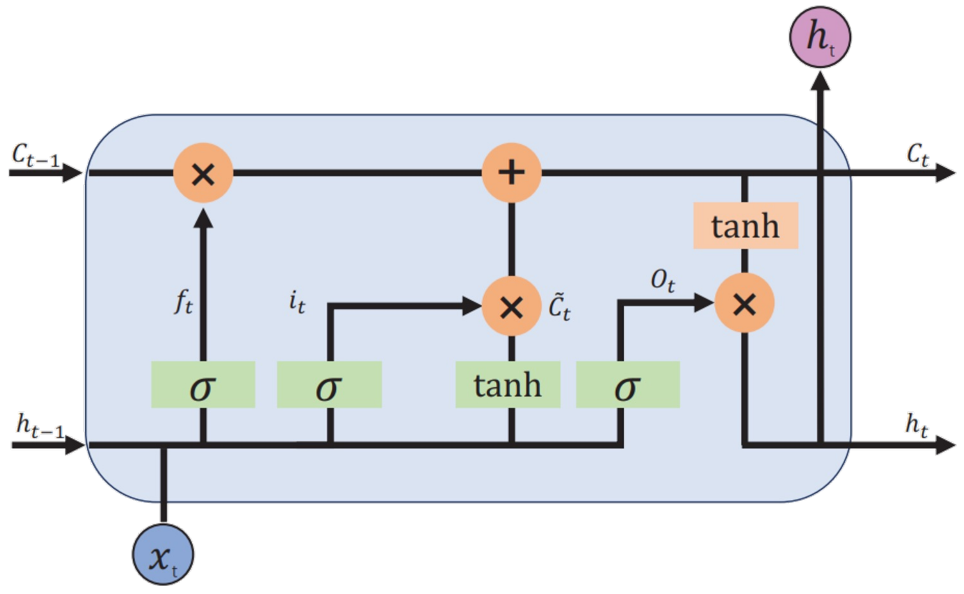
## ◆ Knowledge Tracing (KT) Task



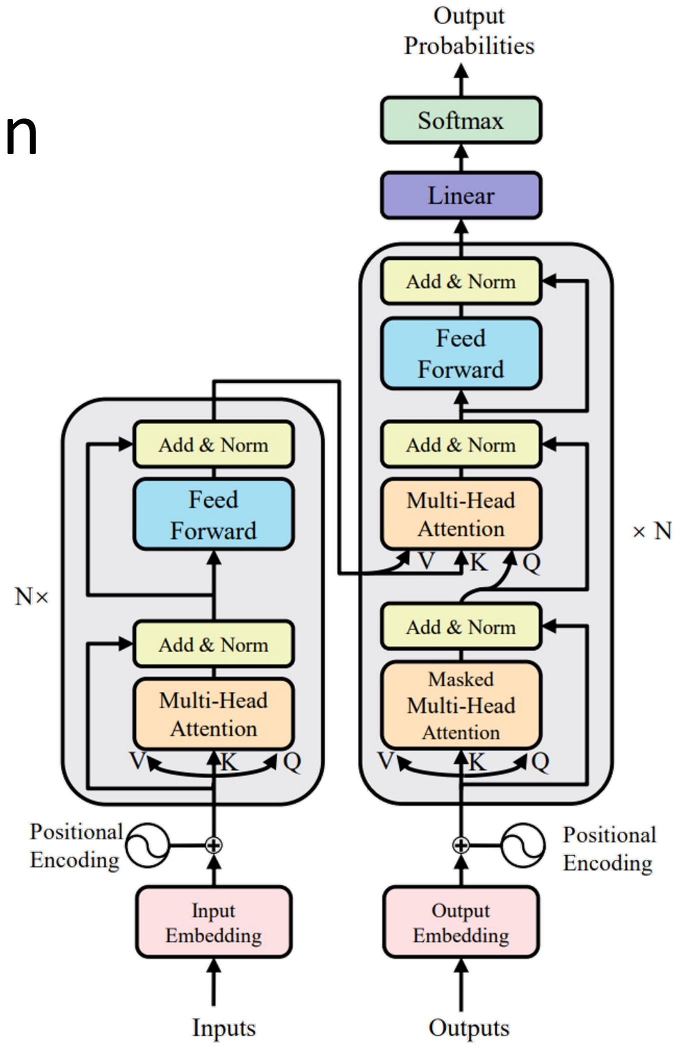

- ➢ KT aims to reveal the student's mastery on each knowledge concept after he/she completed each exercise;
- ➢ Existing approaches (based on probabilistic or logistic models and DNNs) solve KT tasks as **a sequence prediction task**, where student's knowledge states are implicit in the hidden vectors.

# ●Motivation & Idea

Current knowledge tracing (KT) models are based on LSTMs or Transformers




LSTM network




Transformer

**Strengths:**

- Significantly **better performance** than other DNN-based KT approaches

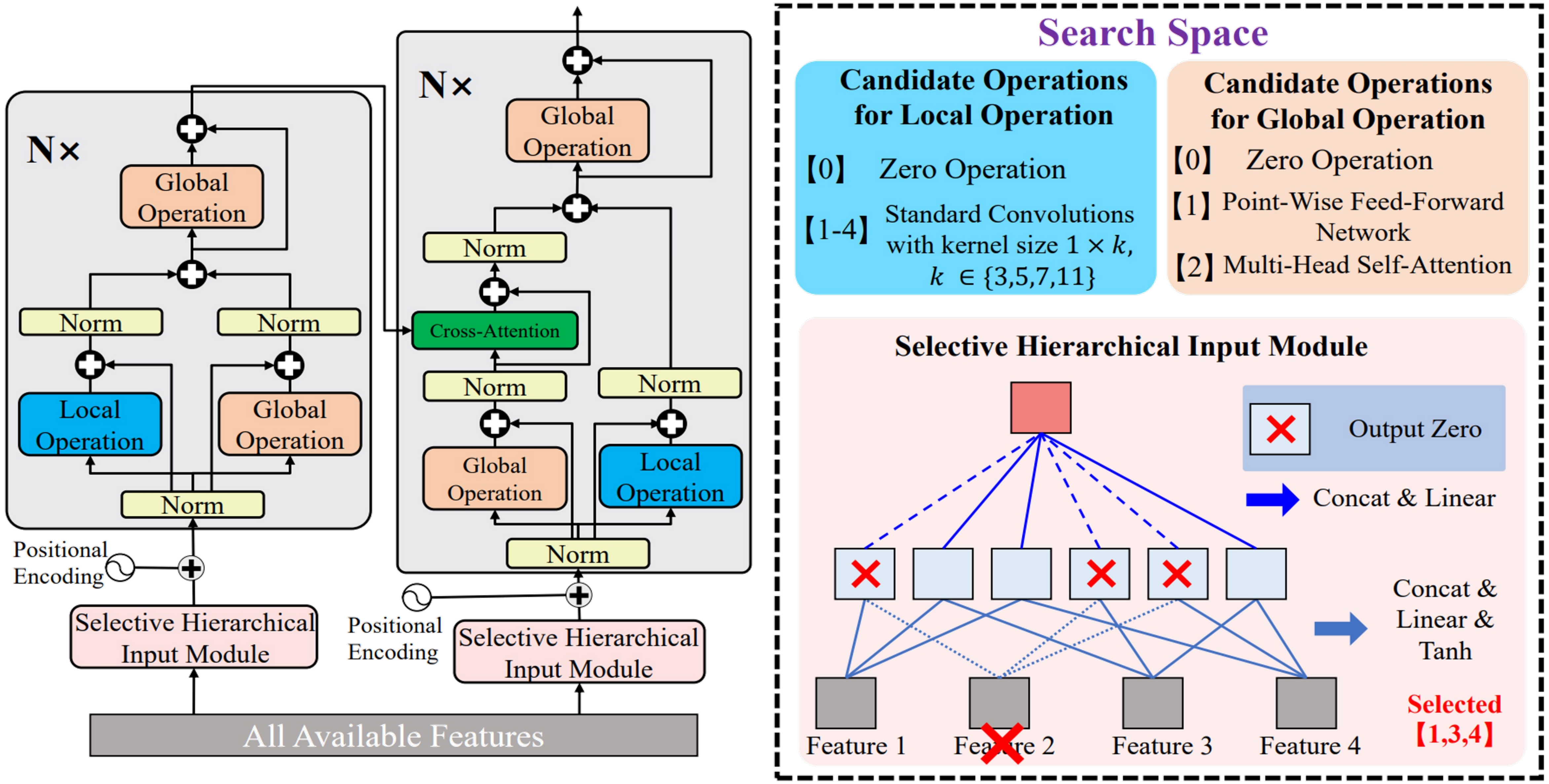
**Weakness:**

- **Manually-selected** input **features**
- **Simple input fusion** methods (Add, Concat)
- Directly employing **vanilla Transformer**
- Lacking **architecture design** for *forgetting behavior modeling*

**Research Motivation**:

- Current KT models **directly** employ existing DNNs architectures （especially, **Transformer-based** KT models show **significantly good performance**）, overcome some problems (such as student's forgetting behavior ) **only from the model inputs** (manually fuse inputs);
- Never considering the influence of **model architectures** to improve performance**;**
- Besides， existing NAS approaches **cannot be directly applied** to KT, due to the search space difference.

# ●Search Space Design

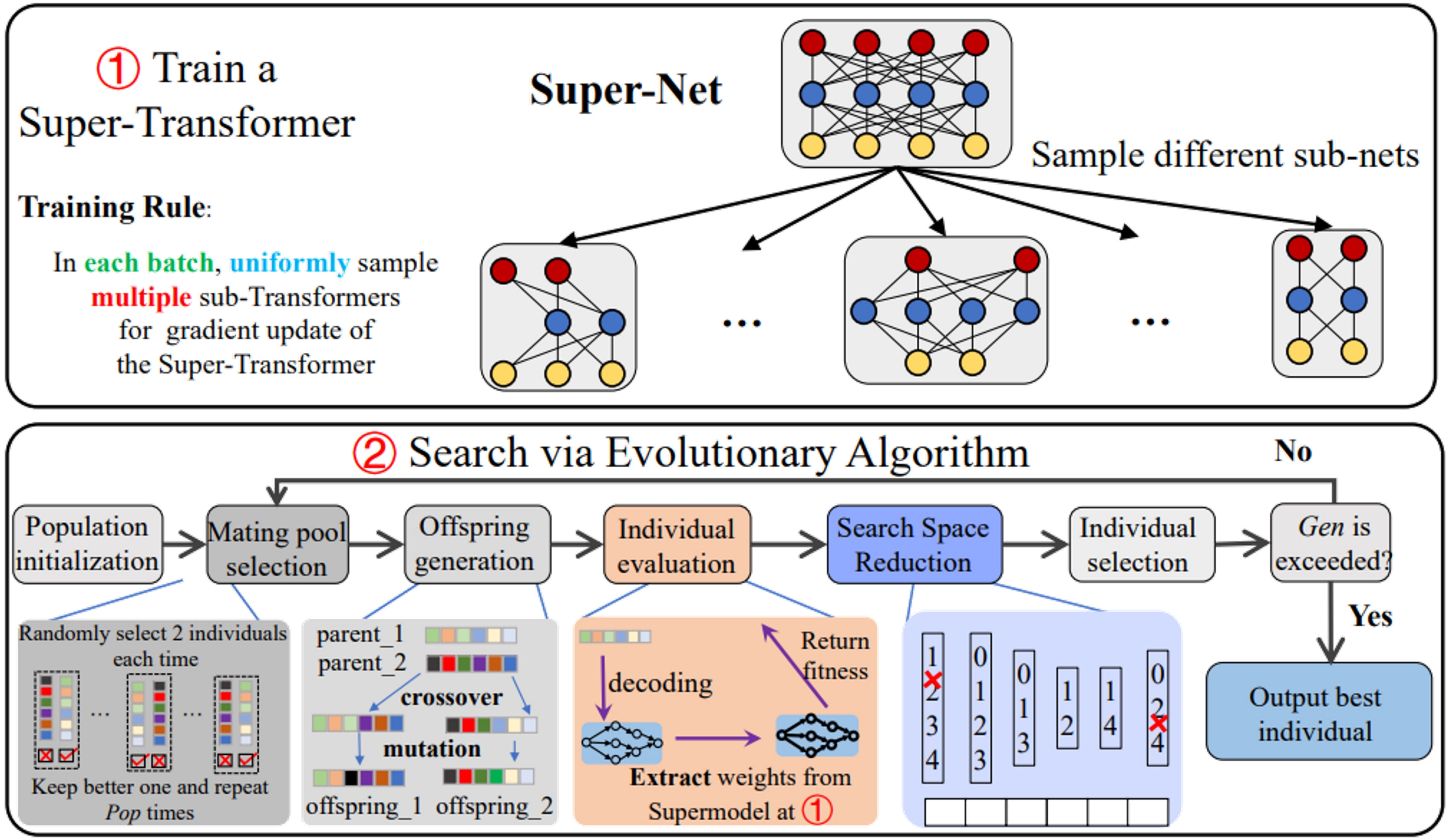



**Transformer-based search space**

Main Design

- **Introducing** convolution operation-based local context modelling : balance attention-based global context modelling, enhance the modelling for different learning behaviors（such as students' forgetting behaviors）
- Replace MHSA and FFN with a global operation module: increase the diversity of contained model architectures
- Design a selective hierarchical input module for **automatically** selecting input features

# ●Overall Framework




**Main strategy**:

1. **Supernet-based evaluation**:

   train a super-Transformer for subsequent evaluation, reducing the search cost

2. **Search Space Reduction Strategy**:

   progressively delete some worse operations, accelerating the convergence

# Experiments-overall comparison

安徽大學

**Table 1 Overall Performance Comparison in terms of AUC and ACC**

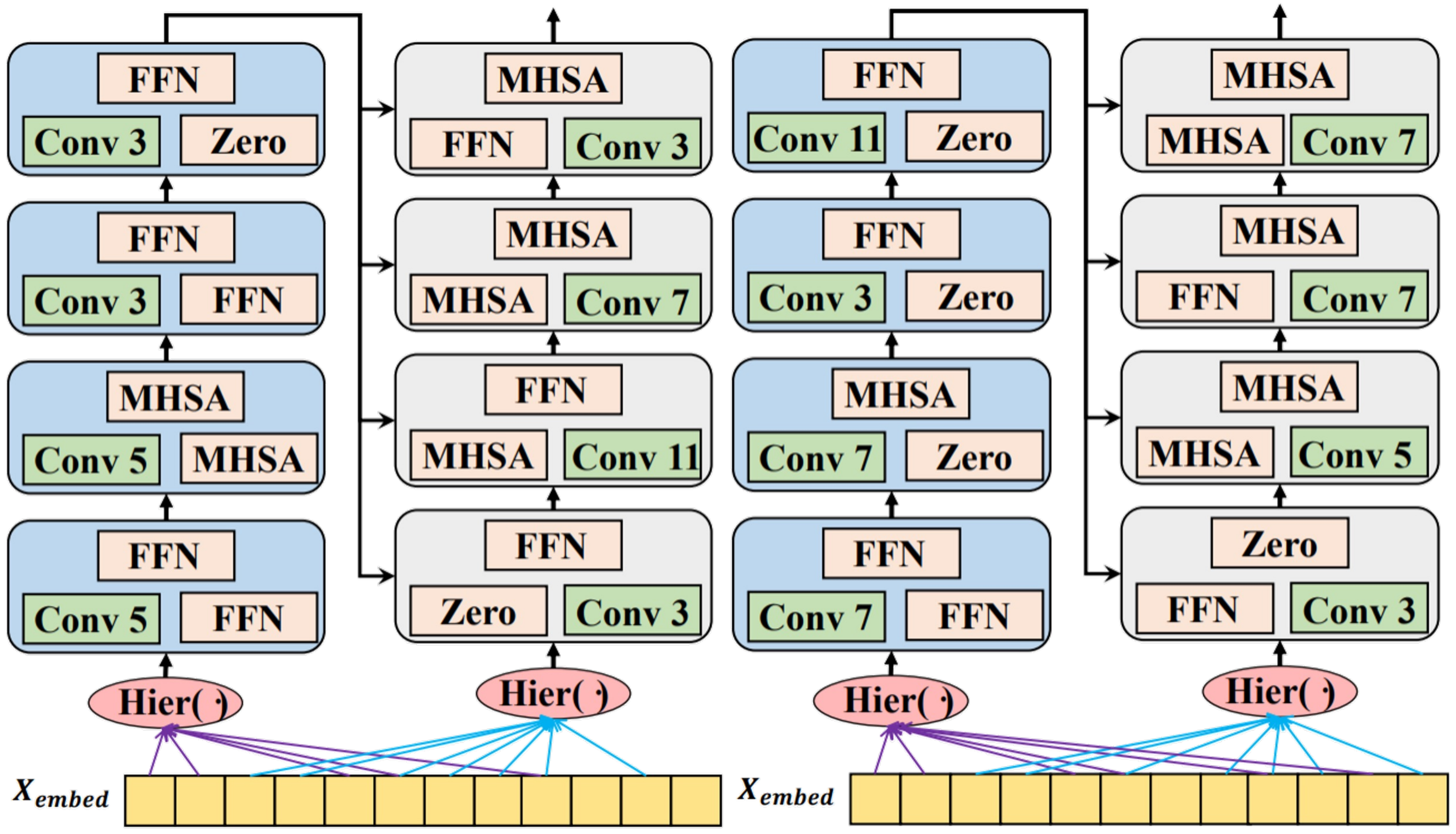
| Dataset | Metric | DKT | HawkesKT | CT-NCM | SAKT | AKT | SAINT | SAINT+ | NAS-Cell | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| Param.(M) | EdNet | 0.13495 | **0.019578** | 1.9974 | 2.0864 | 1.2330 | 2.7492 | 3.1862 | 1.8692 | 3.8232 |
| | RAIEd2020 | 0.13531 | **0.019932** | 2.0431 | 2.1317 | 1.2335 | 2.7945 | 3.2315 | 1.9145 | 4.1262 |
| EdNet | RMSE ↓ | 0.4653 | 0.4475 | 0.4364 | 0.4405 | 0.4399 | 0.4322 | 0.4285 | 0.4345 | **0.4209** |
| | ACC ↑ | 0.6537 | 0.6888 | 0.7063 | 0.6998 | 0.7016 | 0.7132 | 0.7188 | 0.7143 | **0.7295** |
| | AUC ↑ | 0.6952 | 0.7487 | 0.7743 | 0.7650 | 0.7686 | 0.7825 | 0.7916 | 0.7796 | **0.8062** |
| RAIEd2020 | RMSE ↓ | 0.4632 | 0.4453 | 0.4355 | 0.4381 | 0.4368 | 0.4310 | 0.4272 | 0.4309 | **0.4196** |
| | ACC ↑ | 0.6622 | 0.6928 | 0.7079 | 0.7035 | 0.7076 | 0.7143 | 0.7192 | 0.7167 | **0.7313** |
| | AUC ↑ | 0.7108 | 0.7525 | 0.7771 | 0.7693 | 0.7752 | 0.7862 | 0.7934 | 0.7839 | **0.8089** |
| +/-/≈ (six results totally) | | 6/0/0 | 6/0/0 | 6/0/0 | 6/0/0 | 6/0/0 | 6/0/0 | 6/0/0 | 6/0/0 | - |

Overall Comparison
On two datasets

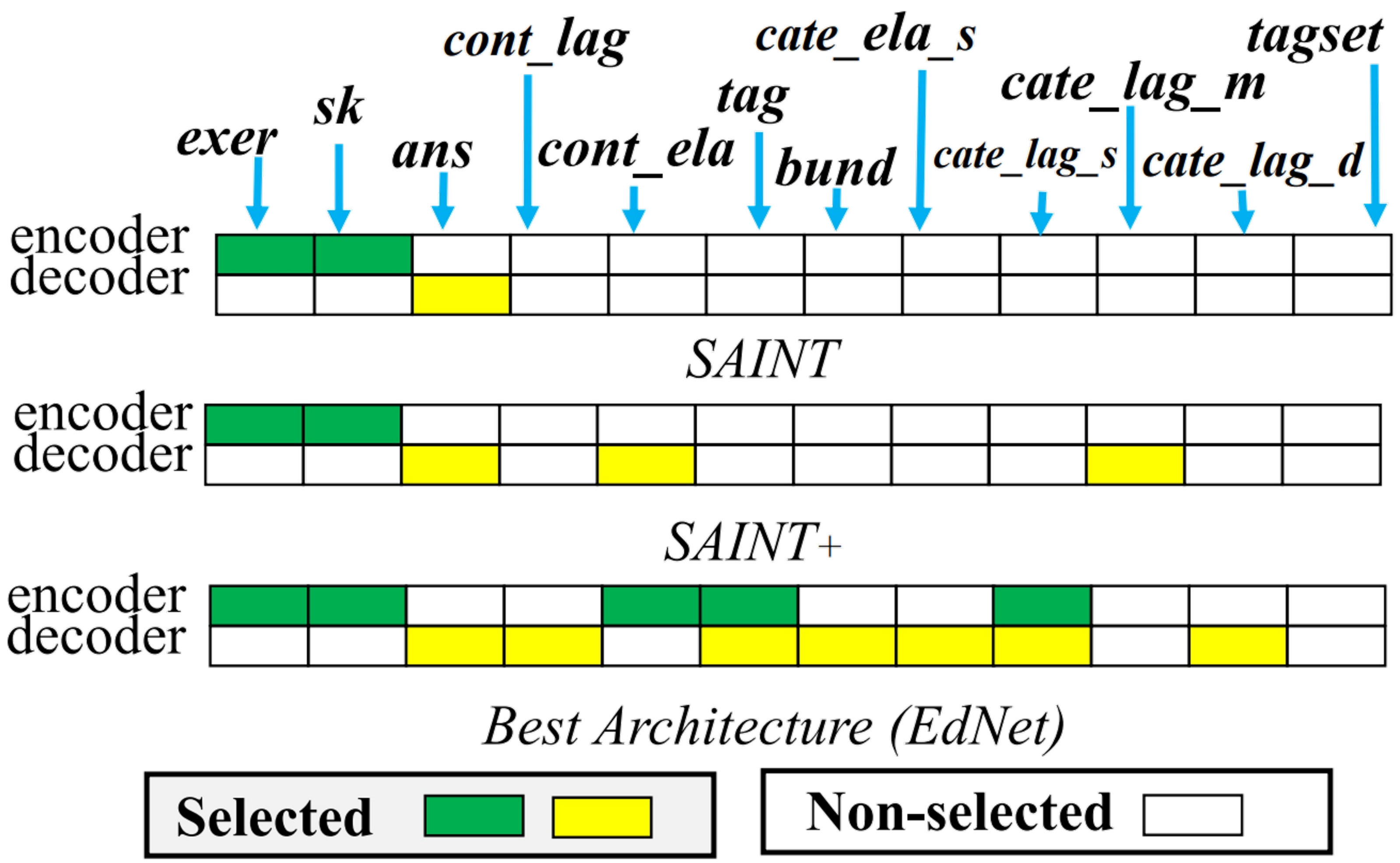
# Experiments- Visualization

## Found Architecture Visualization

**Best-found architectures on two datasets**




**The selected features in the best-found architecture**




**Some insightful observations**

- Prefer local operations like convolution when **close to the input**
- Prefer global operations (such as MHSA & convolution with larger kernel size) when **close to the output**
- **Automatically selected features** contain manually-selected features, also contain others

# ●Experiments-Ablation study

安徽大學

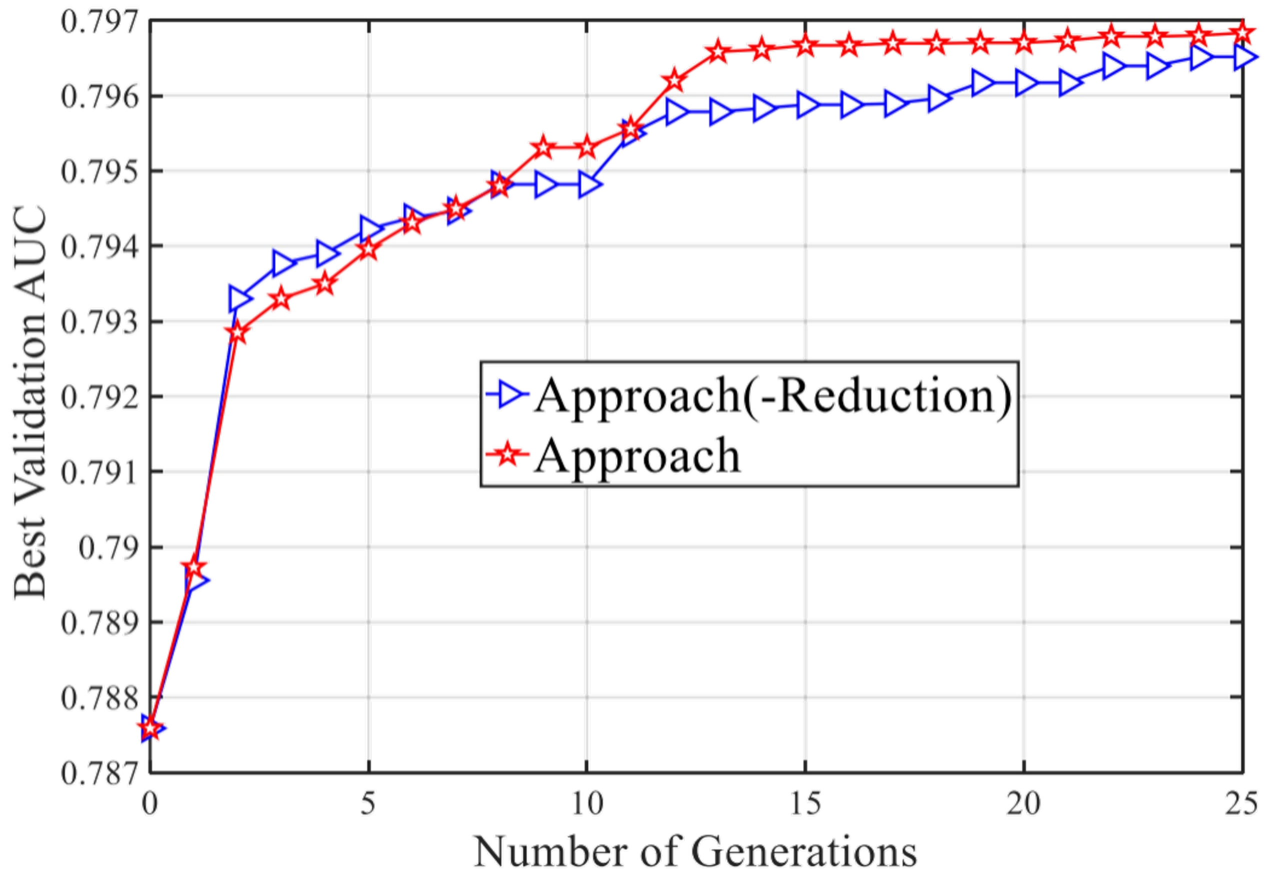
## Effectiveness of the devised modules

| Method | RMSE↓ | ACC↑ | AUC↑ |
|---|---|---|---|
| SAINT+ | 0.4285 | 0.7188 | 0.7916 |
| ***A***: All Features + Concat | 0.4276 | 0.7203 | 0.7937 |
| ***B***: Selected Features + Concat | 0.4262 | 0.7217 | 0.7958 |
| ***C***: Selected Features + Hierarchical | 0.4250 | 0.7236 | 0.7987 |
| ***D***: *C*'s Input + Convolution | 0.4235 | 0.7253 | 0.8012 |
| ***E***: **Ours** (without Hierarchical Fusion, with Concat) | 0.4223 | 0.7269 | 0.8041 |
| ***F***: **Ours** (without the Selected Features, with All Features) | 0.4221 | 0.7260 | 0.8030 |
| ***G***: **Ours** (without Selected Features & Hierarchical, with SAINT+'s input) | 0.4238 | 0.7249 | 0.8008 |
| ***H***: **Ours** (without the Searched Architecture, with SAINT+'s model), i.e., ***C*** | 0.4250 | 0.7236 | 0.7987 |
| **Searched** by ENAS-KT(f) (under a small Supernet with fewer training: embedding size 64, epoch 30), retrain under size 128, taking 9.1 hours totally | 0.4224 | 0.7271 | 0.8036 |
| **Ours** | **0.4209** | **0.7295** | **0.8062** |

### The followings' effectiveness can be validated：

- The selected (searched) features
- The devised hierarchical input module
- The necessary of introducing convolution
- The devised evolutionary search approach

## Effectiveness of search space reduction




(a) Convergence curve on EdNet

The reduction strategy can indeed accelerate the convergence, leading to better convergence results

# Thanks！！