



MEGVII 旷视

Feedback-Based Restormer

2022 MegCup炼丹大赛（小模型盲降噪）——第三名方案

Team: Feedforward



靳鑫 (队长)
Xin Jin



武睿祺
Rui-Qi Wu



李震
Zhen Li



李重仪 (指导老师)
Chongyi Li

团队分工



靳鑫 (队长)
Xin Jin

- 基本代码框架搭建
- 数据处理
- 损失函数设计
- 部分model相关代码撰写
- 完成Pytorch代码到
MegEngine代码转换



武睿祺
Rui-Qi Wu

- 基本代码框架搭建
- 尝试各种优化策略
- 数据统计
- 尝试各种炼丹trick



李震
Zhen Li

- 算法调研与指导
- 部分model代码撰写
- 数据分析



李重仪 (指导老师)
Chongyi Li

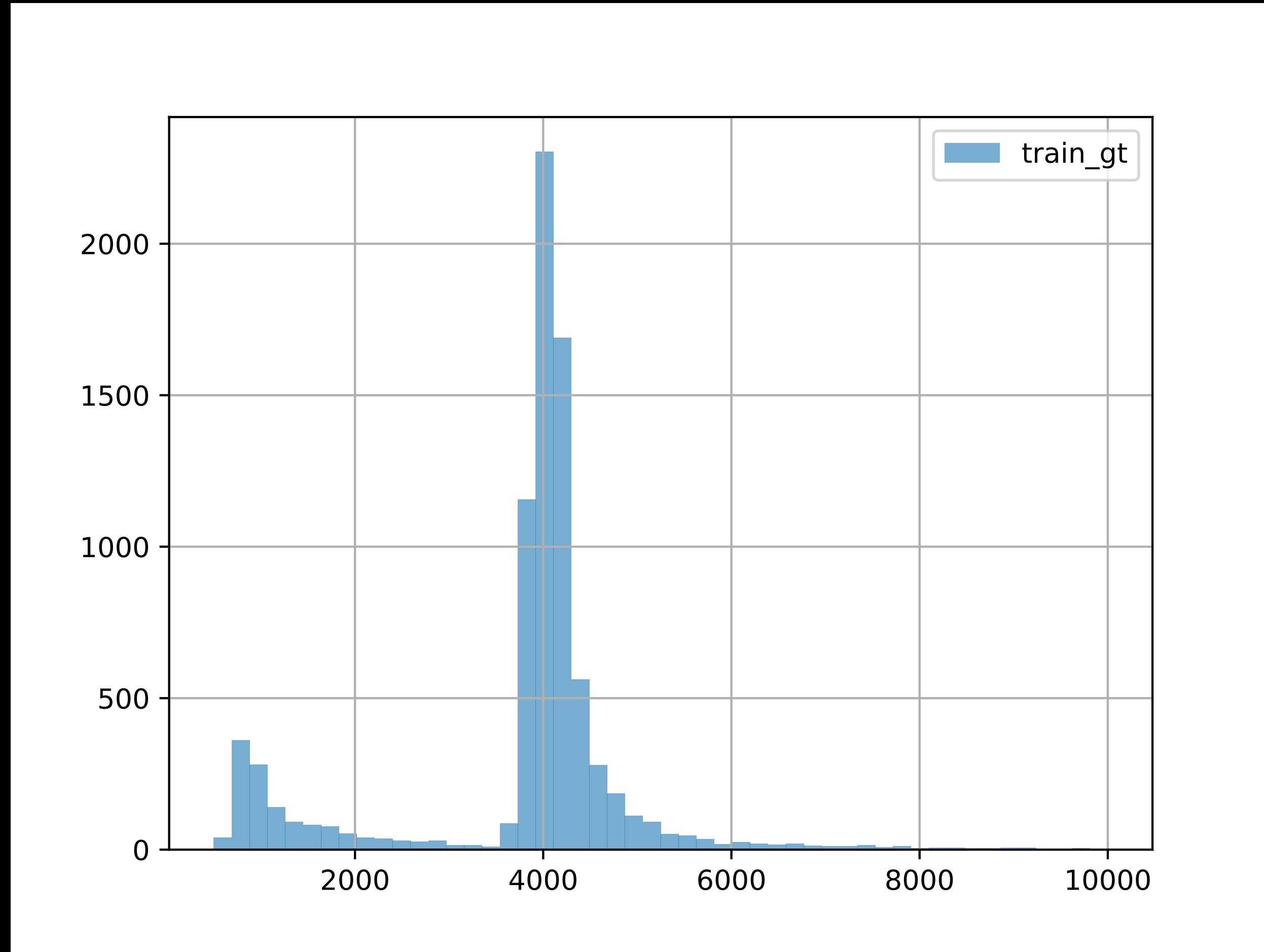
- 算法调研与指导
- 组织讨论

More than Model

- 黑电平
 - 统计
 - 标准化
- Pixel-wise Focal Loss

More than Model

- 黑电平
 - 统计



More than Model

- 黑电平

$$\frac{(X + \beta) - E(X + \beta)}{\sqrt{D(X + \beta)}} = \frac{X - E(X)}{\sqrt{D(X)}}$$

- 标准化

More than Model

$$L_{focal} = \left\| \exp(-y) \cdot |x - y| \right\|_1$$

- Pixel-wise Focal Loss

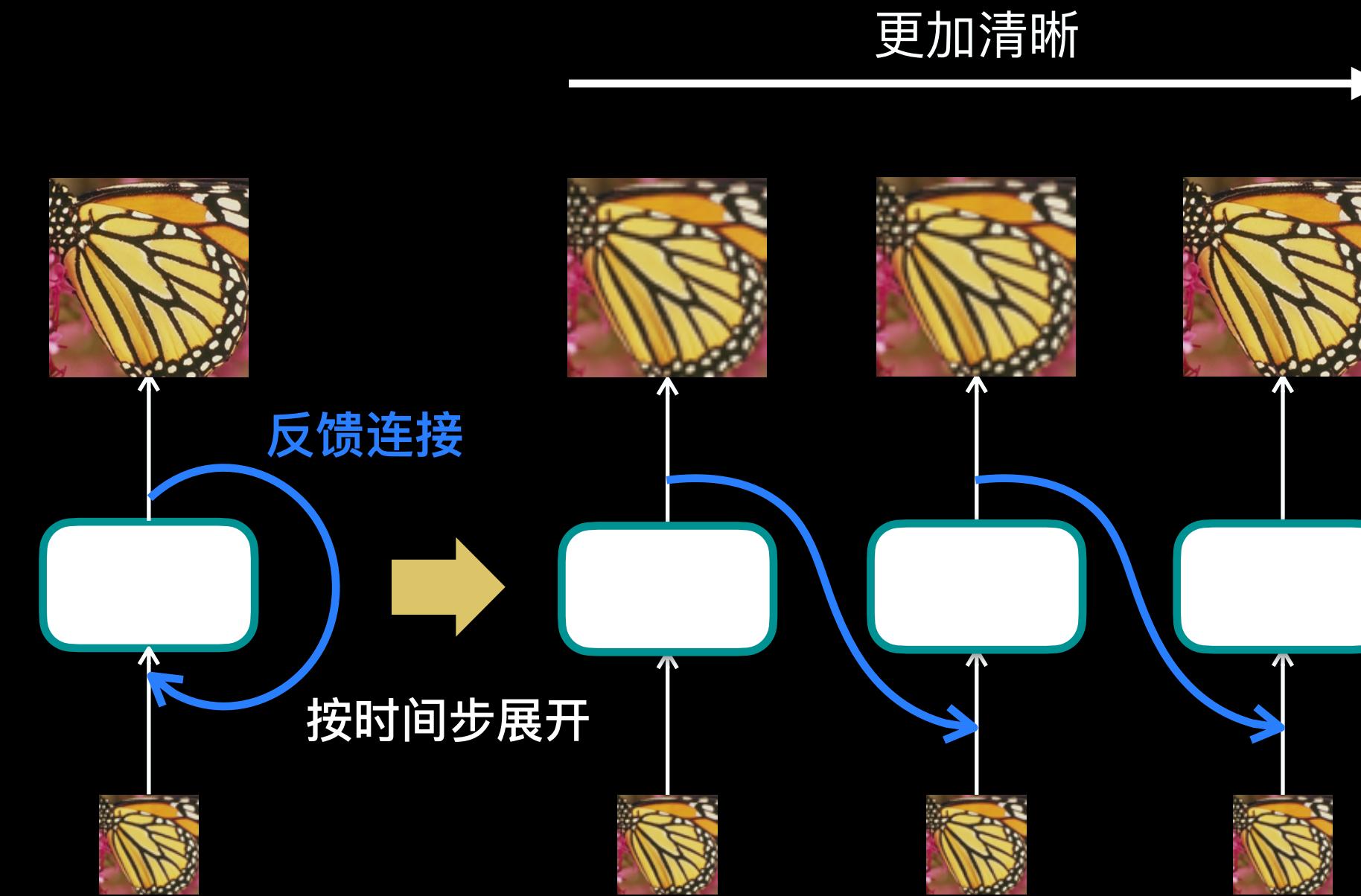
More than Model

- 黑电平
 - 统计
 - 标准化
- Pixel-wise Focal Loss

Failure Case

- 黑电平
 - 统计
 - 标准化
- Pixel-wise Focal Loss

背景

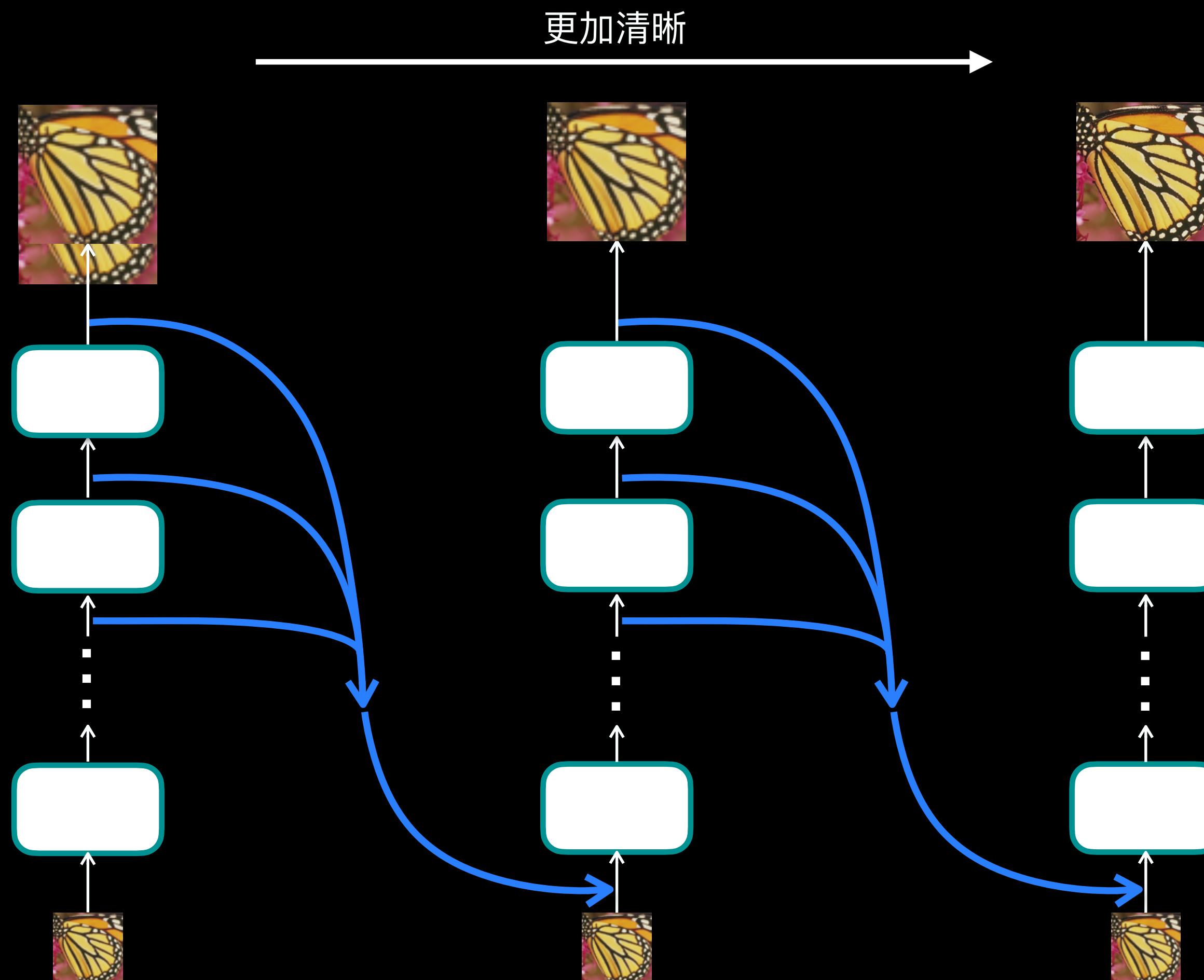


核心思想：高质量图像的信息可以帮助低质量图像进行更好的复原

反馈系统的三个条件：

1. 使用**循环**结构（以实现迭代过程）
2. 每个时间步都有**loss**做监督（以使反馈连接包含高级特征信息）
3. 每个时间步都提供**输入**（以保证每个时间步都有待被调制的低级信息）

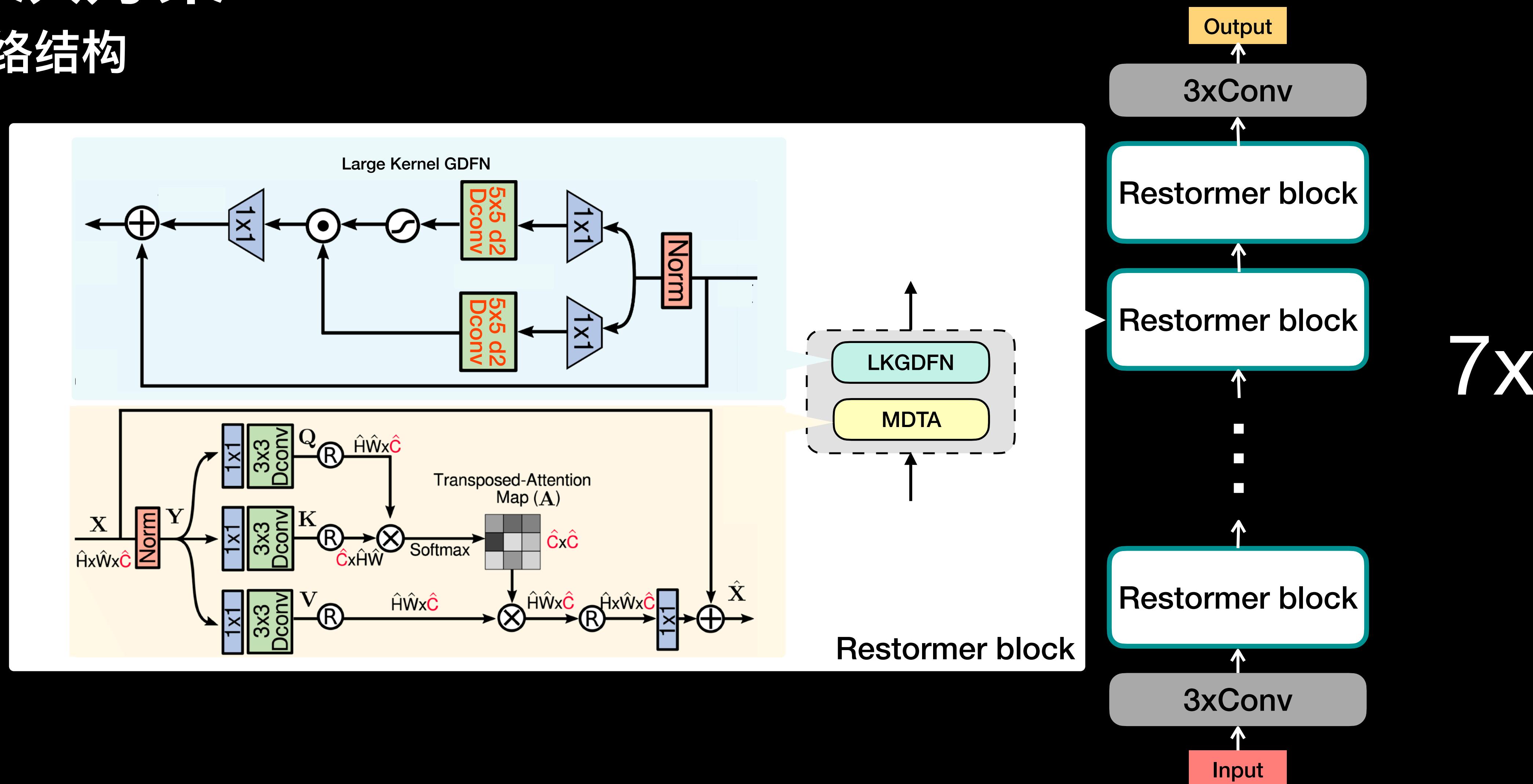
背景



更多的反馈：将最顶层的几组特征反馈至最底层

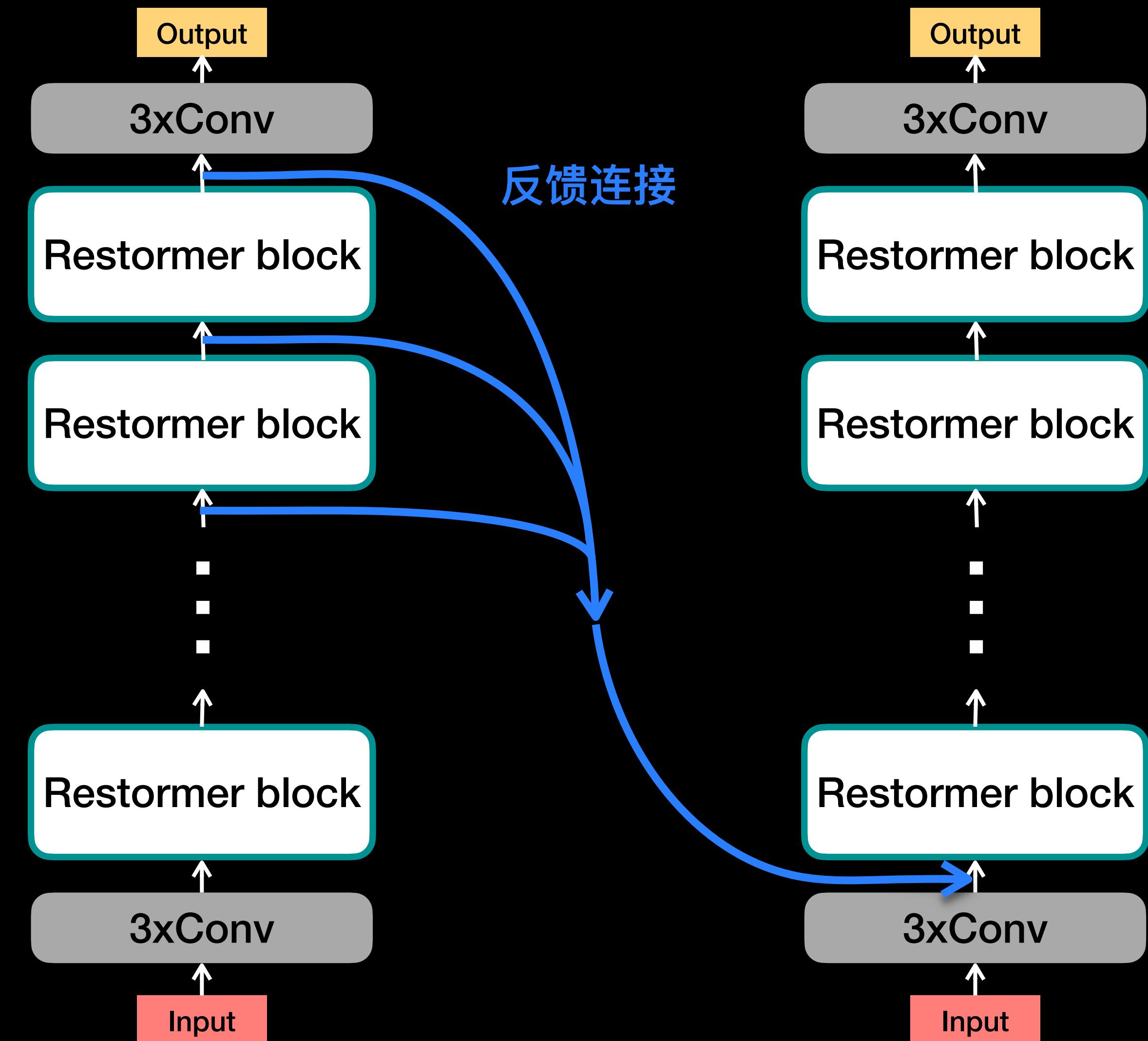
解决方案

网络结构



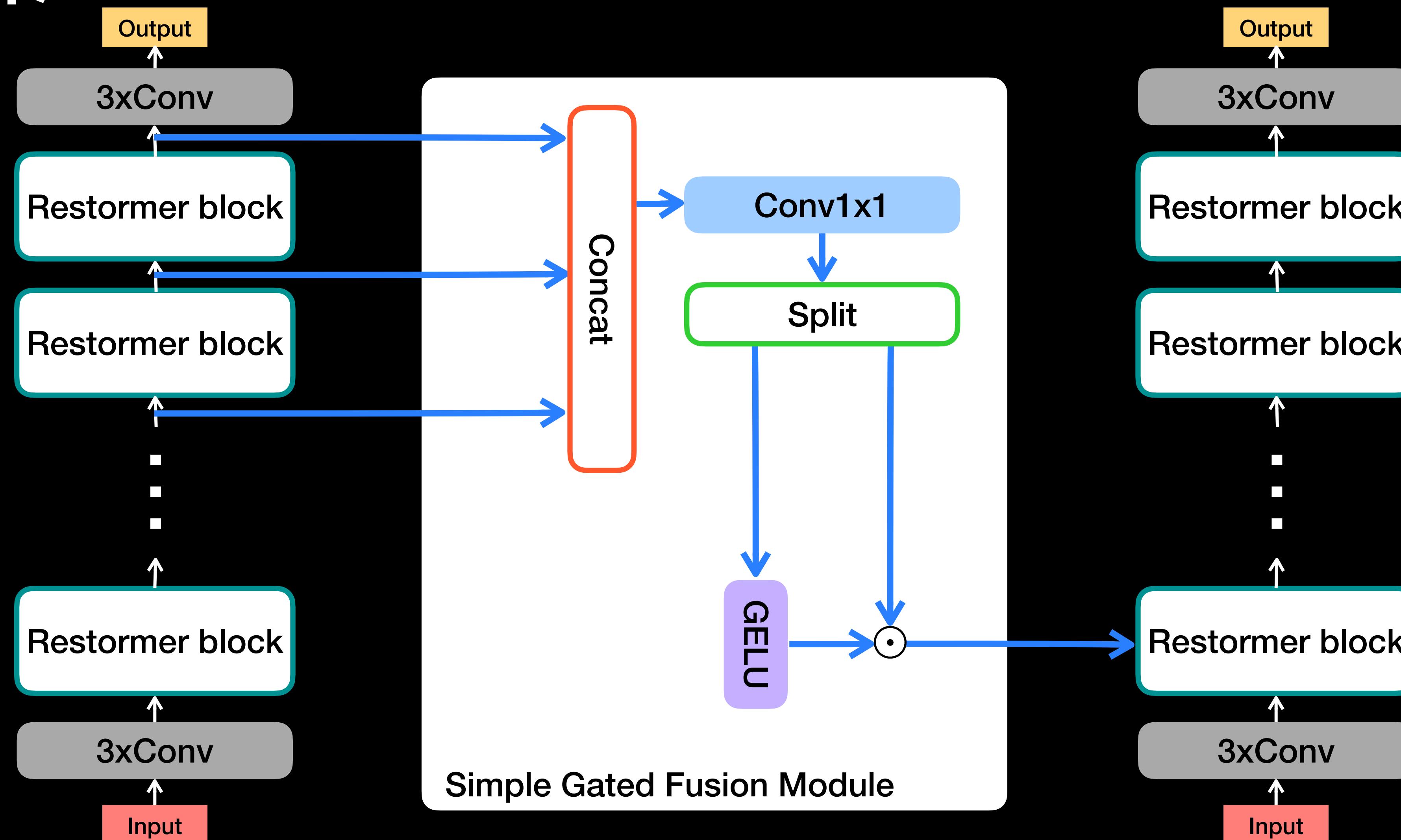
解决方案

网络结构



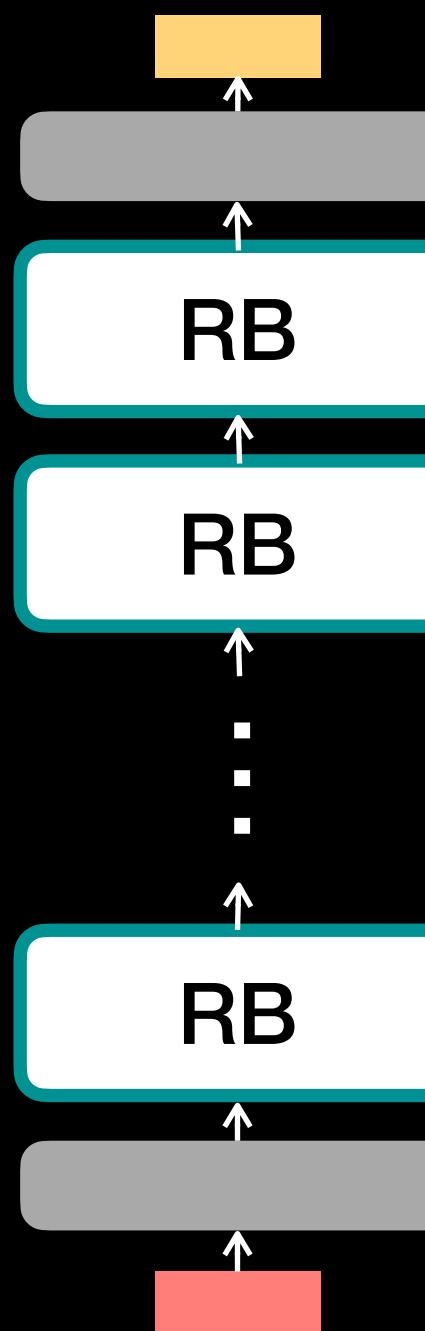
解决方案

网络结构



实验结果

Feedback相关实验



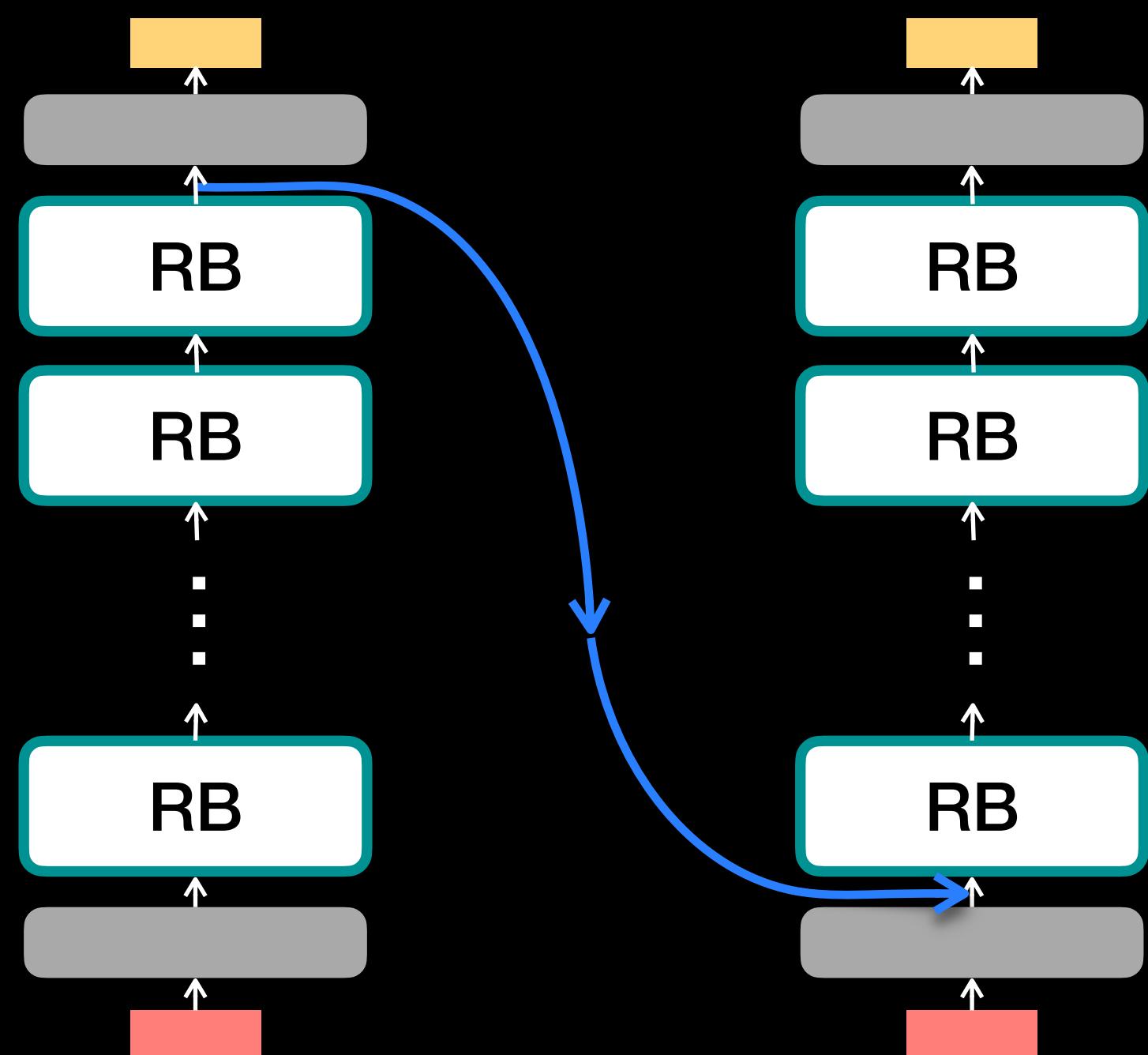
RB: Restormer Block

*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
无循环结构	91.912K	8.9709

实验结果

Feedback相关实验



RB: Restormer Block

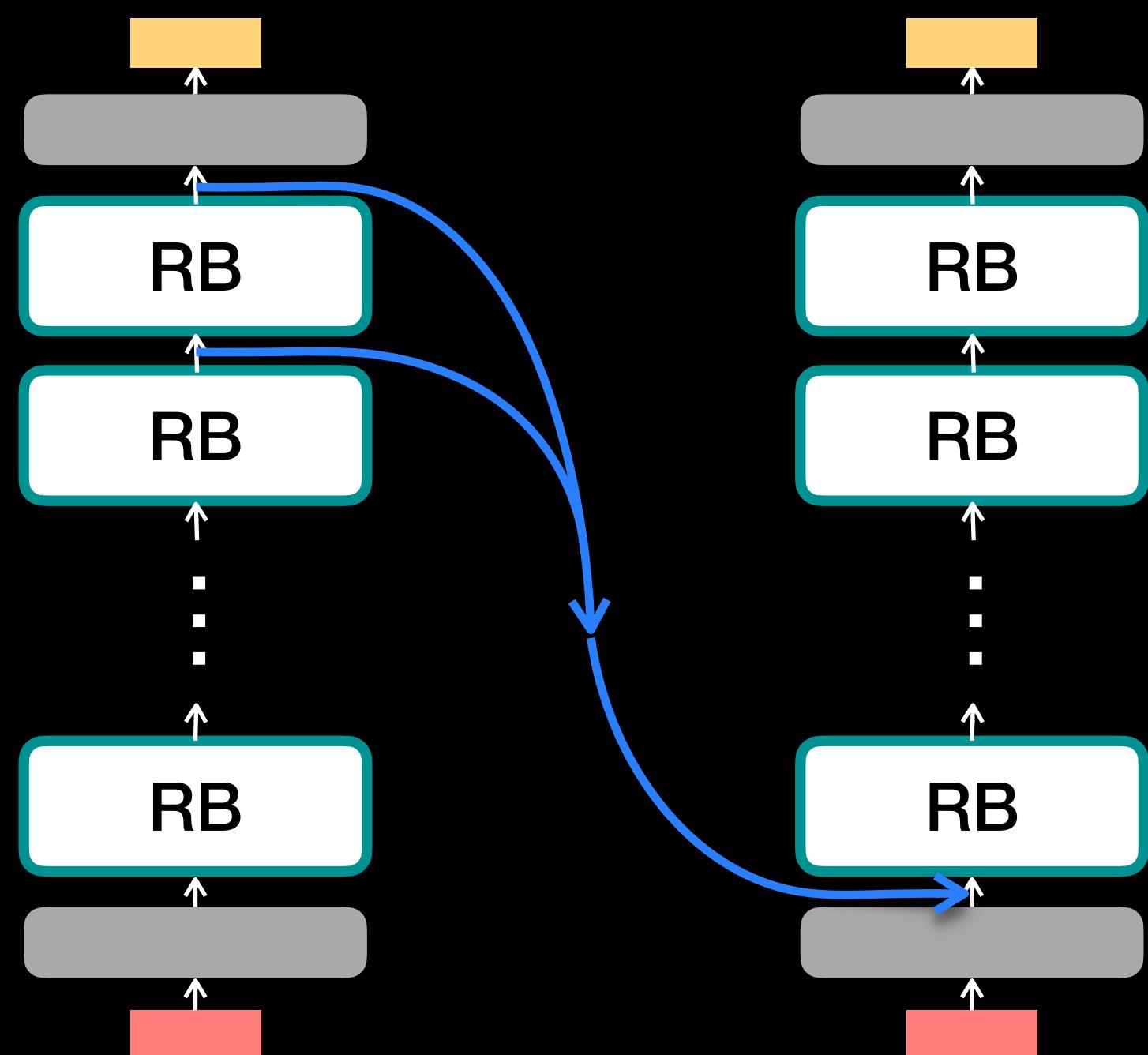
*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
无循环结构	91.912K	8.9709
用最顶层的一组特征反馈一次	95.060K	8.9749

↑ 0.4%

实验结果

Feedback相关实验



RB: Restormer Block

*验证集按9:1选取，验证分数比提交分数低约0.007

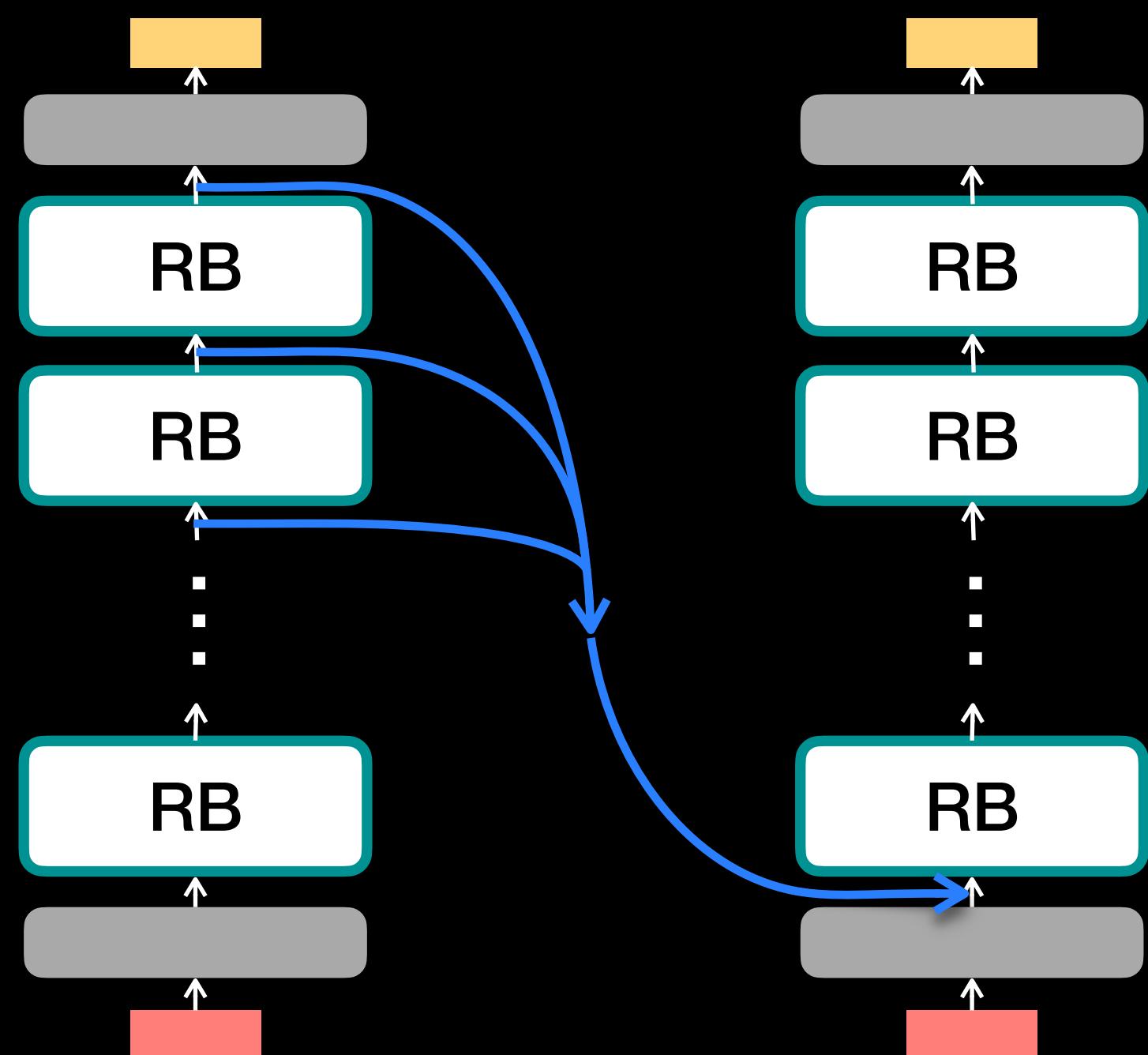
模型	参数量	验证集分数
无循环结构	91.912K	8.9709
用最顶层的一组特征反馈一次	95.060K	8.9749
用最顶层的两组特征反馈一次	97.108K	8.9806

0.4%

1.0%

实验结果

Feedback相关实验



RB: Restormer Block

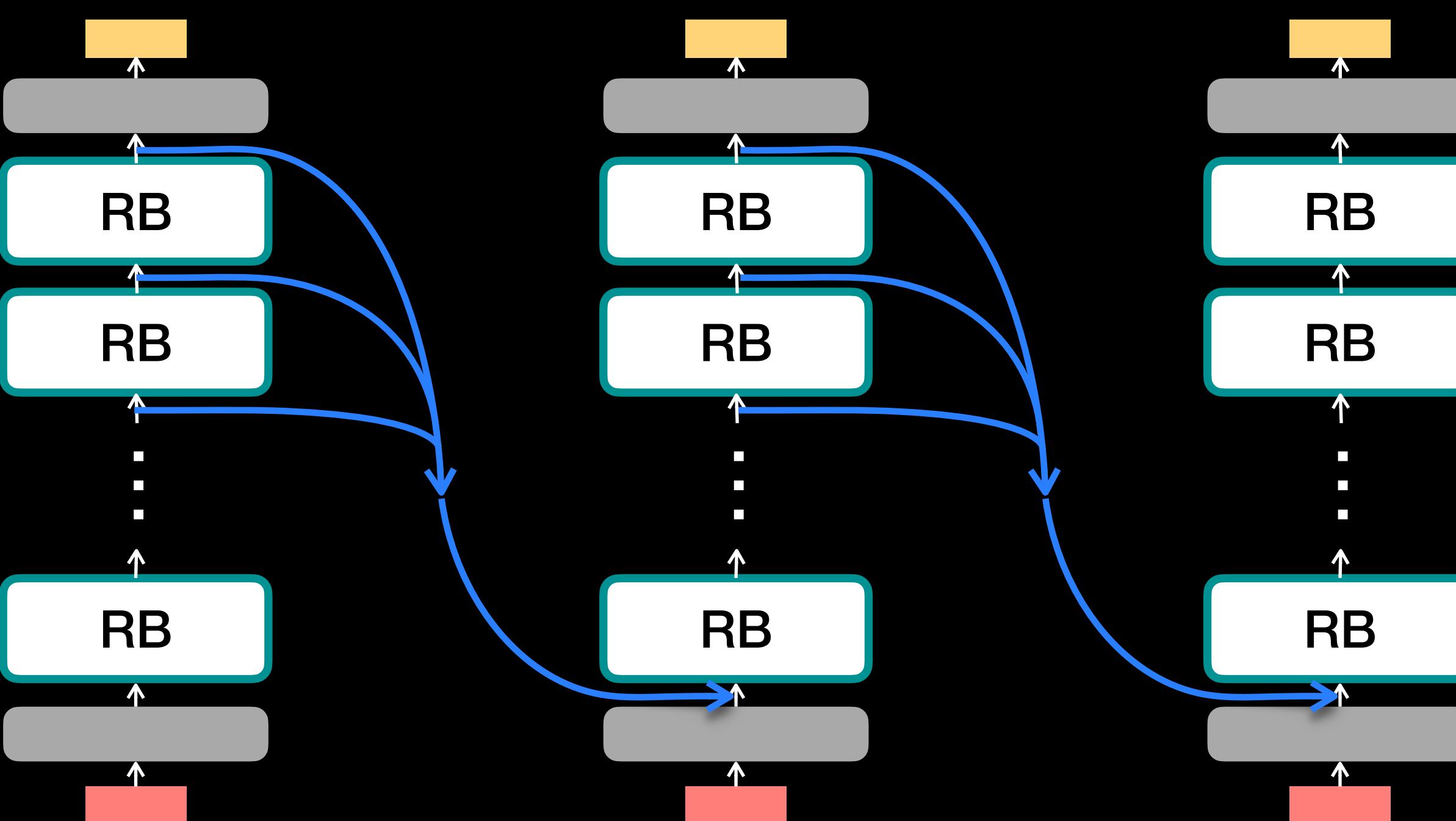
*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
无循环结构	91.912K	8.9709
用最顶层的一组特征反馈一次	95.060K	8.9749
用最顶层的两组特征反馈一次	97.108K	8.9806
用最顶层的三组特征反馈一次	99.156K	8.9846

0.4%
1.0%
1.4%

实验结果

Feedback相关实验



RB: Restormer Block

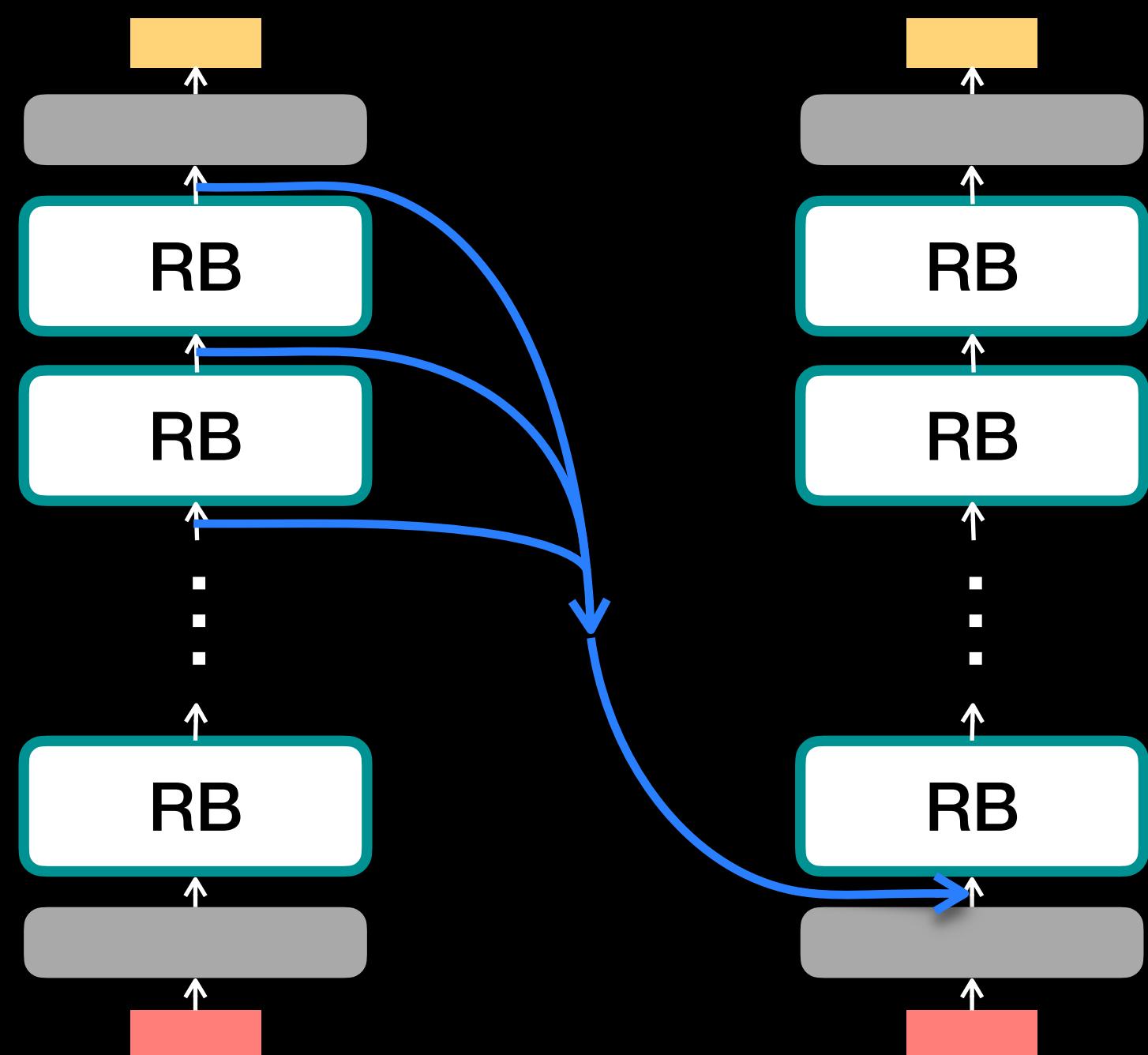
*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
无循环结构	91.912K	8.9709
用最顶层的一组特征反馈一次	95.060K	8.9749
用最顶层的两组特征反馈一次	97.108K	8.9806
用最顶层的三组特征反馈一次	99.156K	8.9846
用最顶层的三组特征反馈两次	99.156K	8.9865

0.4%
1.0%
1.4%
1.6%

实验结果

Feedback相关实验



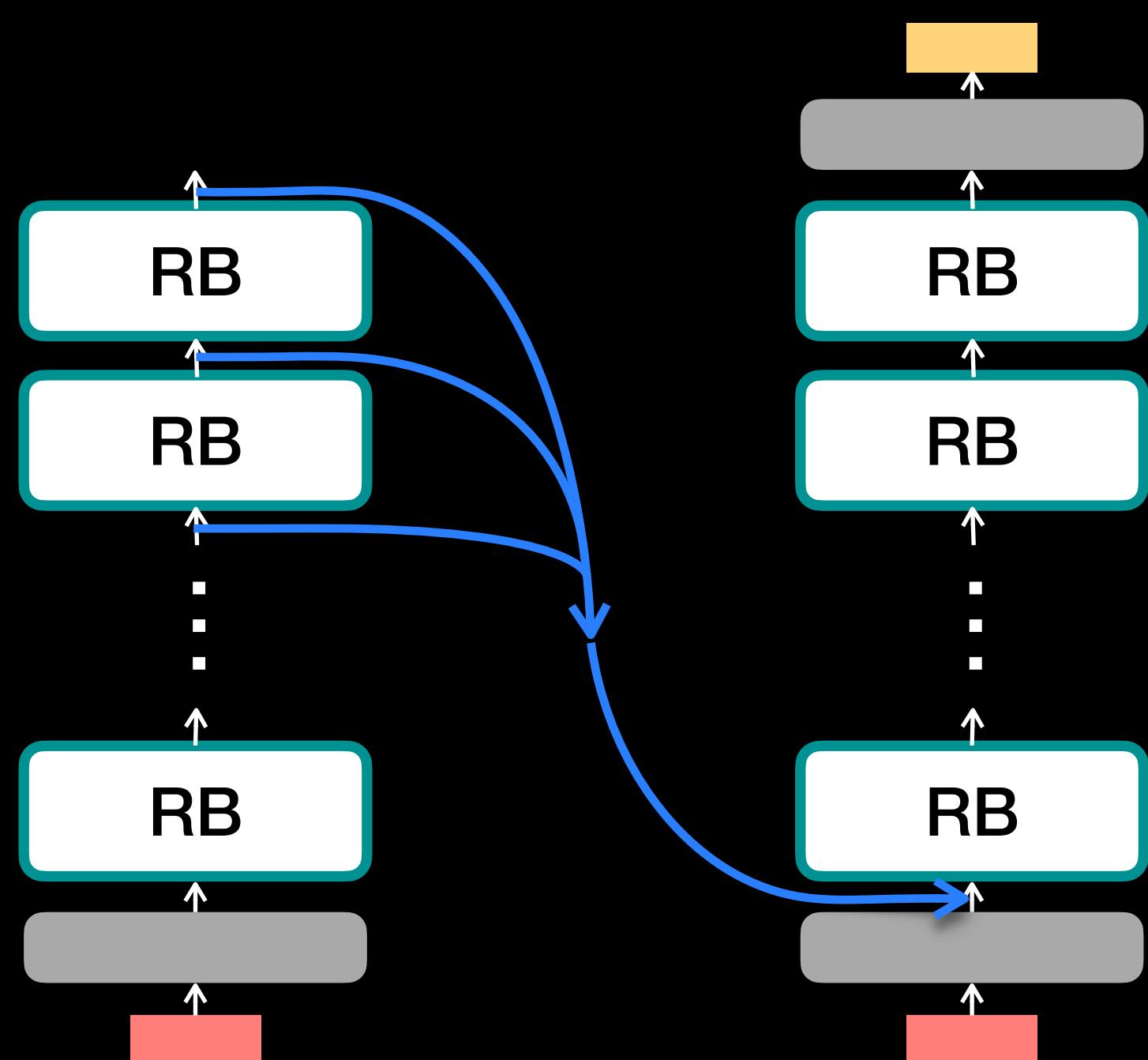
RB: Restormer Block

*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
用最顶层的三组特征反馈一次	99.156K	8.9846

实验结果

Feedback相关实验



RB: Restormer Block

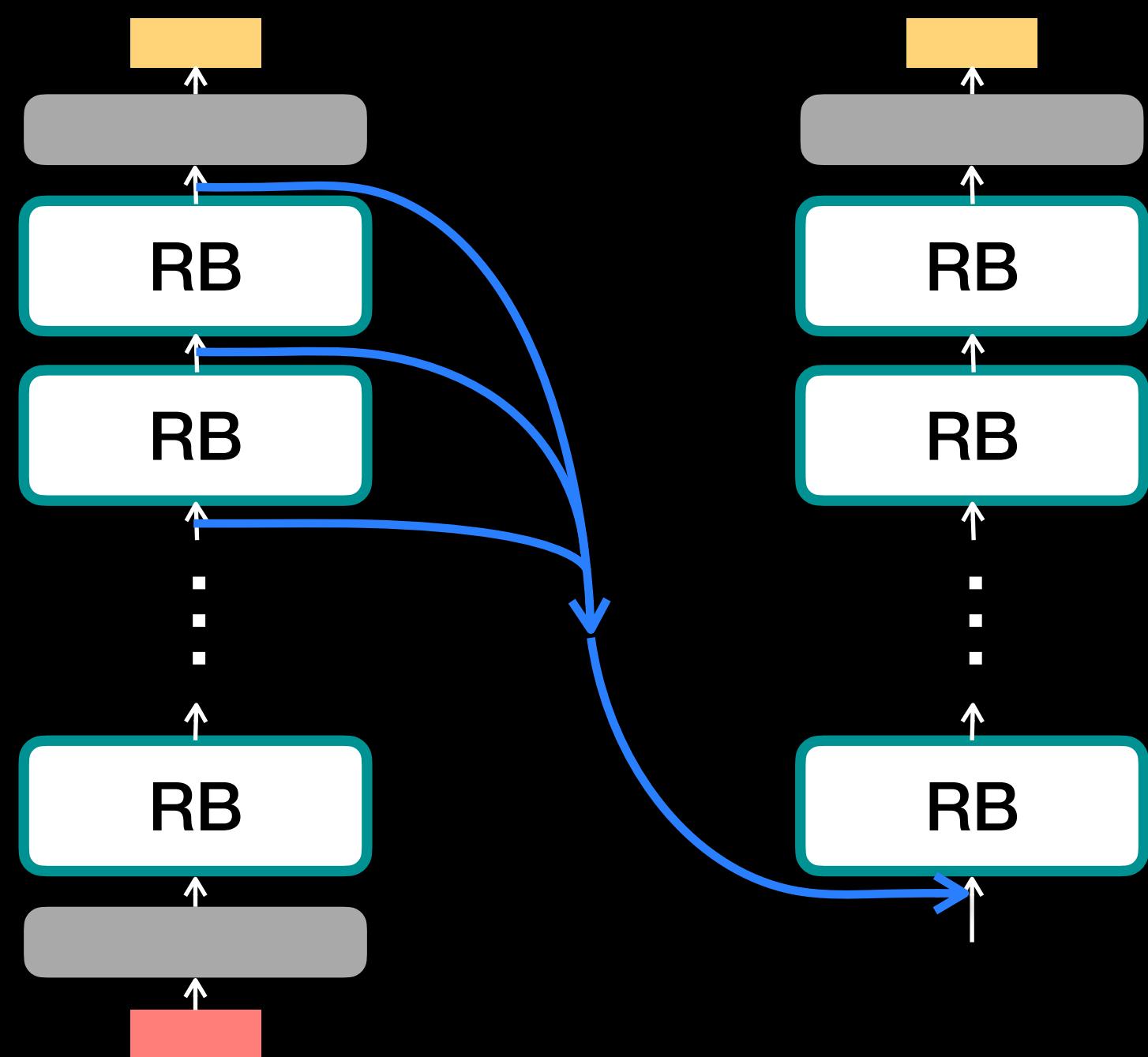
*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
用最顶层的三组特征反馈一次	99.156K	8.9846
只保留最后时间步的loss	-	8.9725

1.2%

实验结果

Feedback相关实验



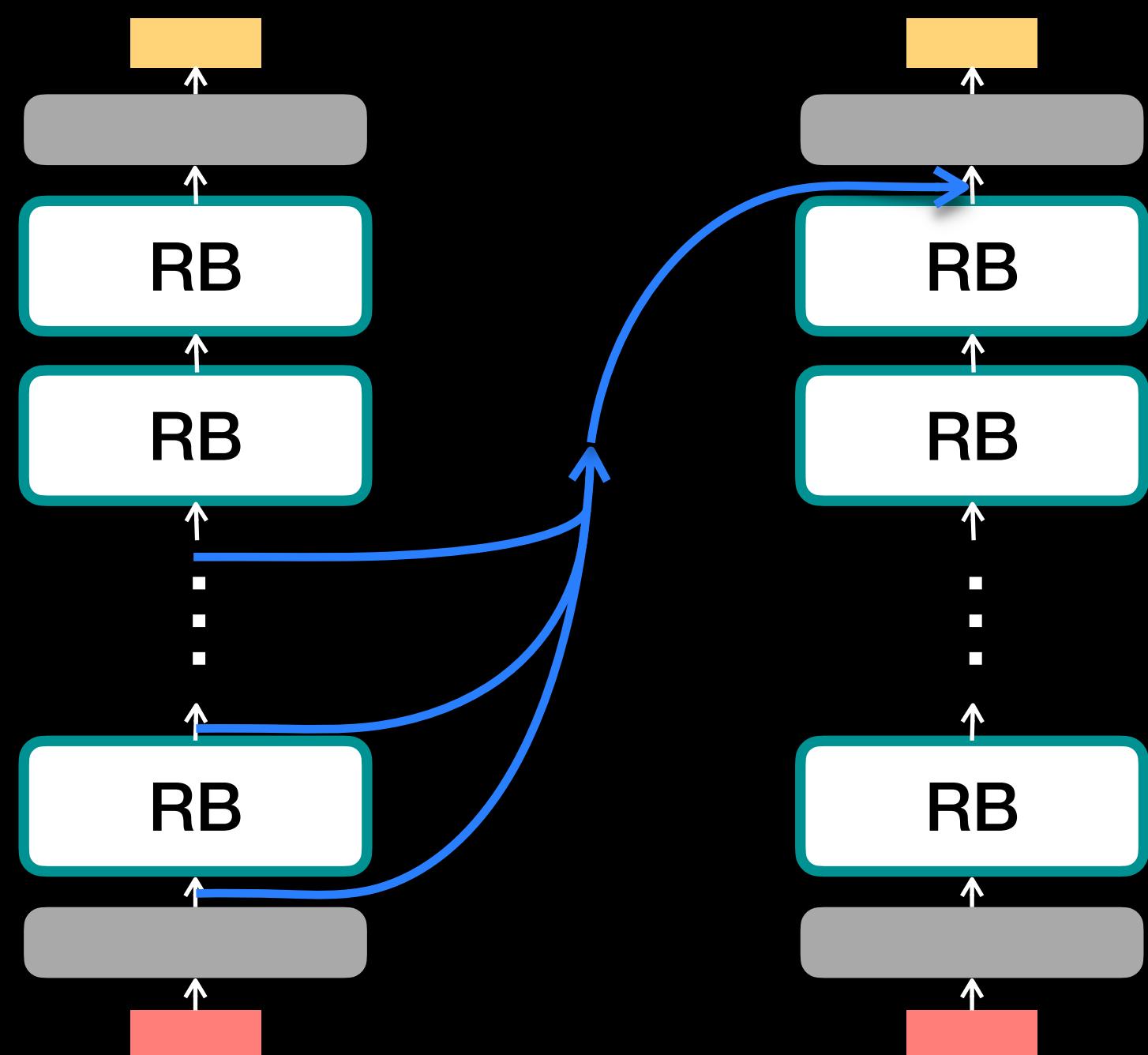
RB: Restormer Block

*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数	
用最顶层的三组特征反馈一次	99.156K	8.9846	↓ 1.2%
只保留最后时间步的loss	-	8.9725	↓ 1.5%
第二个时间步不提供输入特征	-	8.9697	

实验结果

Feedback相关实验



RB: Restormer Block

*验证集按9:1选取，验证分数比提交分数低约0.007

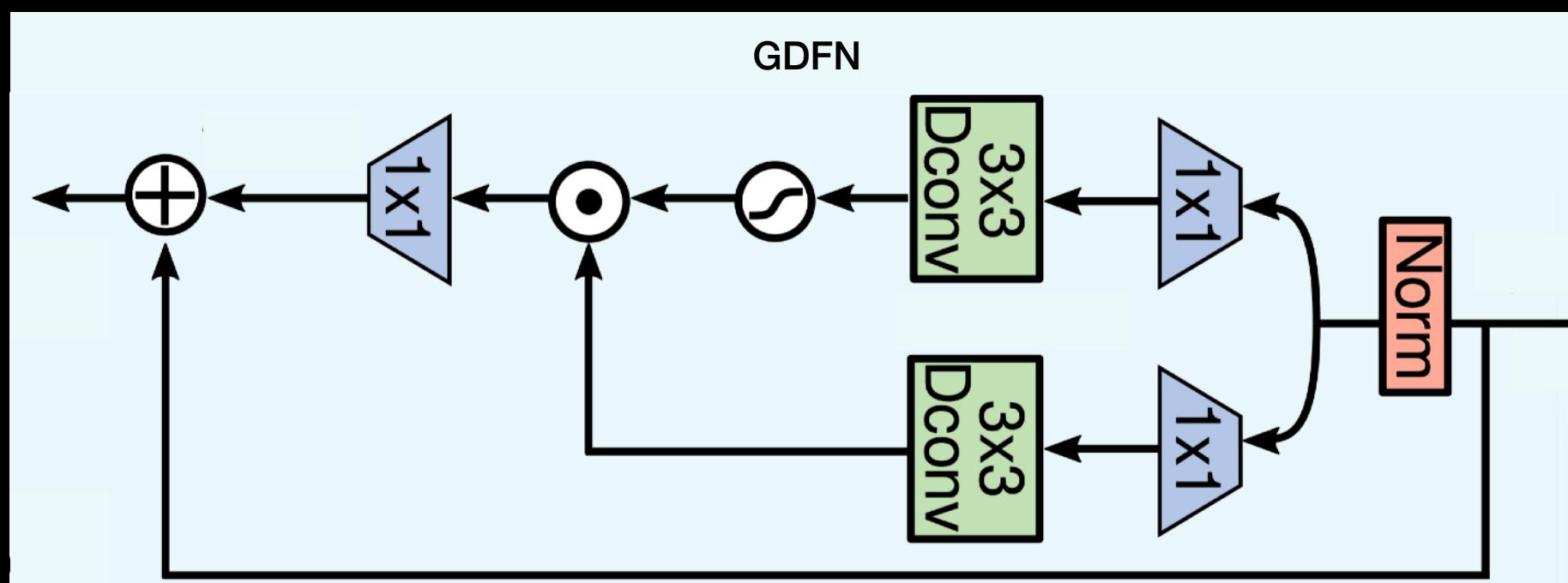
模型	参数量	验证集分数
用最顶层的三组特征反馈一次	99.156K	8.9846
只保留最后时间步的loss	-	8.9725
第二个时间步不提供输入特征	-	8.9697
用最底层的三组特征前传一次	-	8.9375

1.2%
1.5%
4.7%

实验结果

FFN相关实验

*验证集按9:1选取，验证分数比提交分数低约0.007



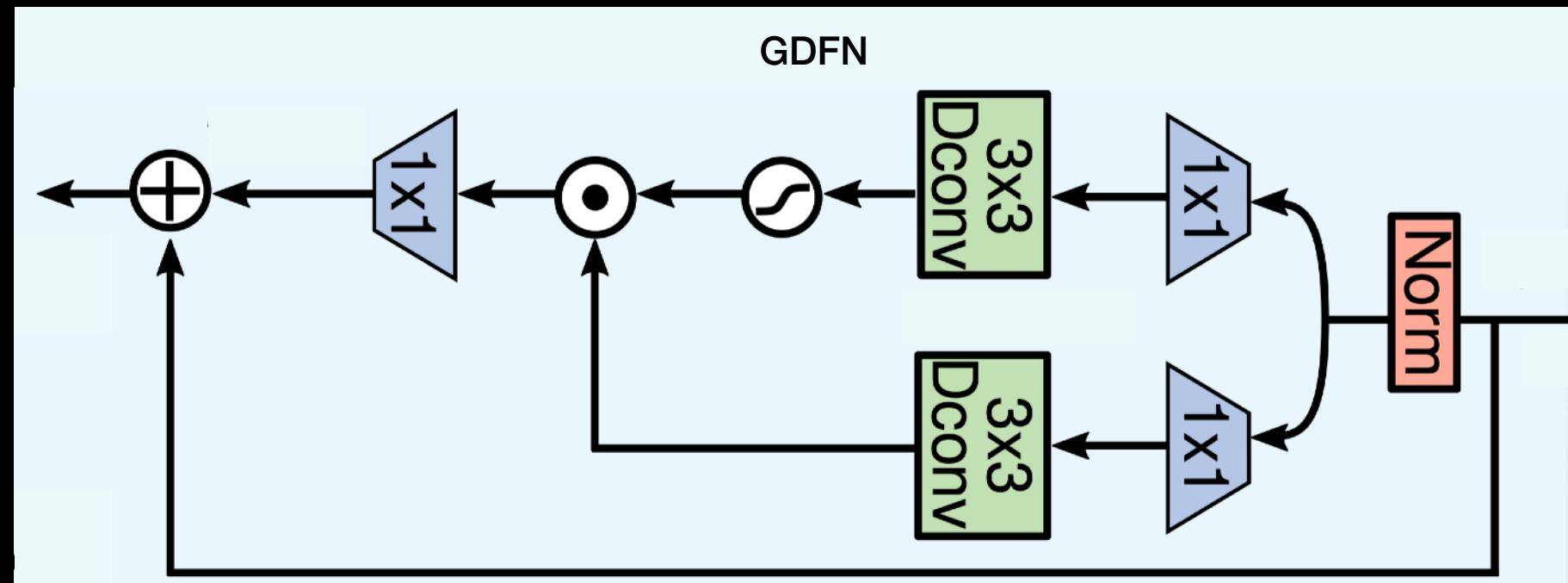
模型	参数量	验证集分数
Large Kernel GDFN	99.156K	8.9846
		0.7% ↓

0.7%

实验结果

FFN相关实验

*验证集按9:1选取，验证分数比提交分数低约0.007



模型	参数量	验证集分数
Large Kernel GDFN	99.156K	8.9846
Restormer原始GDFN	90.644K	8.9771

0.7%

实验结果

Loss设计及相关实验

L1 Loss

$$L_1(x, y) = \|x - y\|_1$$

Weighted L1

$$L_{diff}(x, y) = \textcolor{red}{w} \cdot L_1(x, y)$$

$$w = \sqrt{1/m}, m = \frac{1}{HWC} \sum_i^C \sum_j^H \sum_k^W y_{ijk}$$

*验证集按9:1选取，验证分数比提交分数低约0.007

模型	参数量	验证集分数
L1 Loss	99.156K	8.9830
Weighted L1	-	8.9846

↑ 0.2%

最终训练策略

数据	学习率	迭代次数	学习率衰减	Batch Size	验证分数
90%	0.001	200K	不衰减	8	8.9846

最终训练策略

数据	学习率	迭代次数	学习率衰减	Batch Size	提交分数
90%	0.001	200K	不衰减	8	8.9913

最终训练策略

数据	学习率	迭代次数	学习率衰减	Batch Size	提交分数
90%	0.001	200K	不衰减	8	8.9913
100%	0.001	1000K	每250K衰一半	32	9.0066

↑ 1.5%

+训练数据

+迭代次数

+学习率衰减

+Batch Size

解决方案

核心代码 (Feedback)

```
def forward(self, x):
    shortcut = x
    init_feat = self.conv_in(x)

    x_list = []
    last_feats_list = []

    for _ in range(self.num_steps):
        last_feats_list = self.block(init_feat, last_feats_list)
        # using the highest-level feature to perform reconstruction
        x_list.append(self.conv_out(last_feats_list[-1]) + shortcut)

    return x_list
```

```
def forward(self, input_feat, last_feats_list):

    cur_feats_list = []

    for idx, b in enumerate(self.trans_list):
        # refining the lowest-level features
        if len(last_feats_list) > 0 and idx < self.num_refine_feats:
            input_feat = self.GFM([input_feat] + last_feats_list)

        input_feat = b(input_feat)
        cur_feats_list.append(input_feat)

    # rerouting the highest-level features
    return cur_feats_list[-self.num_reroute_feats:]
```

谢 谢



<https://github.com/Srameo/megcup-feedforward>