深度学习技术及应用
国家工程研究中心 | 飞桨

灵活高效、多场景统一的大模型分布式checkpoint系统研发

答辩人：郑天宇 / zty-king

指导人：陈锐彪 / From00

飞桨护航计划集训营

自我介绍

Tianyu Zheng

zty-king

姓名 郑天宇/zty-king

学校 电子科技大学

参与活动 第九期护航计划、Wave Summit lighting talk

研究方向 **大模型推理优化、分布式、CUDA**

PART1 ：开发任务简介与个人拆解

任务1：自动并行核心模块优化

- 新动半pp框架的相关逻辑修复与功能增强
- 新动半pp对齐模式下，一些场景的功能适配与bug修复

任务2：API算子的修复与适配

- 修复fused_rotary_position_embedding算子反向逻辑及其单测逻辑错误
- 针对paddle的5个window函数与17个Loss函数，做torch对齐，并修复文档

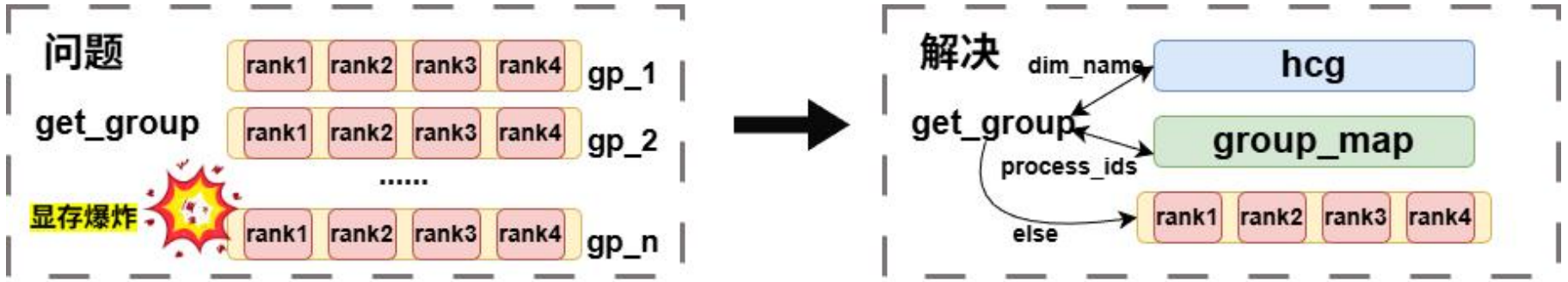
任务3：Flexcheckpoint系统研发

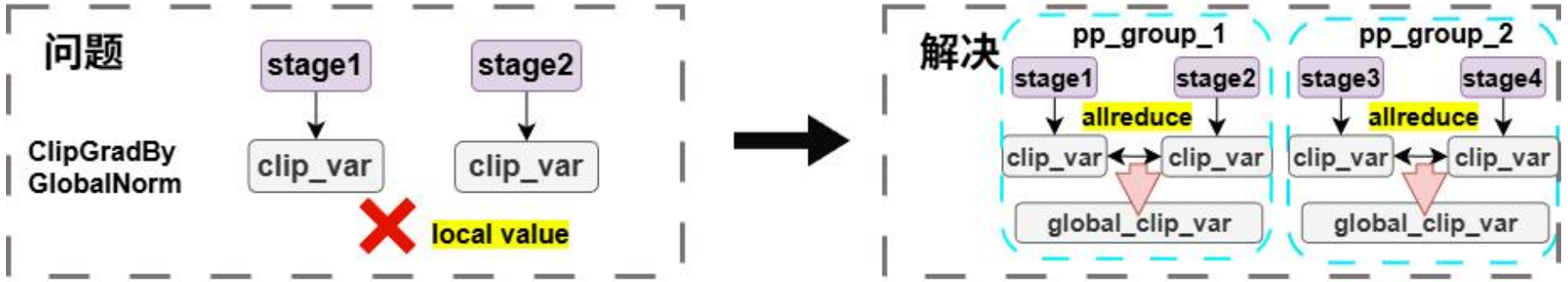
- 在Paddleformers与PaddleNLP适配FC
- 在llama2、ernie4.5上对fc框架做了大量验证工作，并基于此修复FC存在的一些问题
- 完善与优化FC在实际使用过程中存在的问题，对FC进行功能增强。

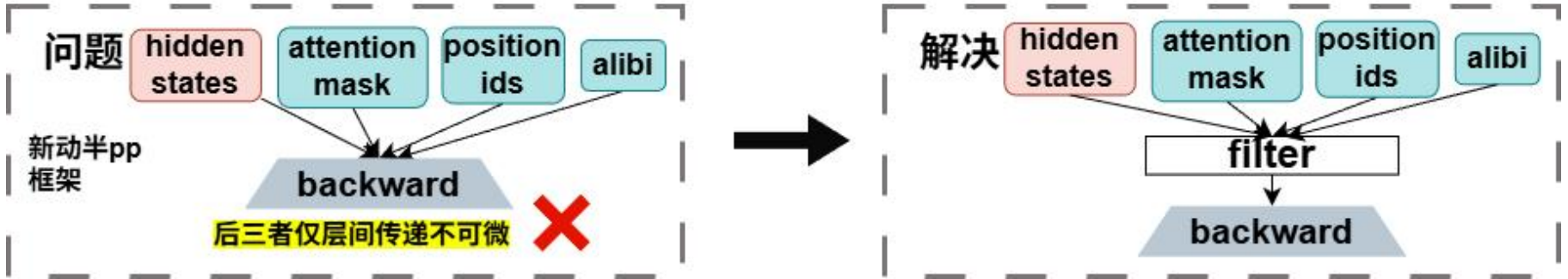
PART2：实际开发工作介绍-自动并行核心模块优化

>>>>>>>>>

➢功能修复



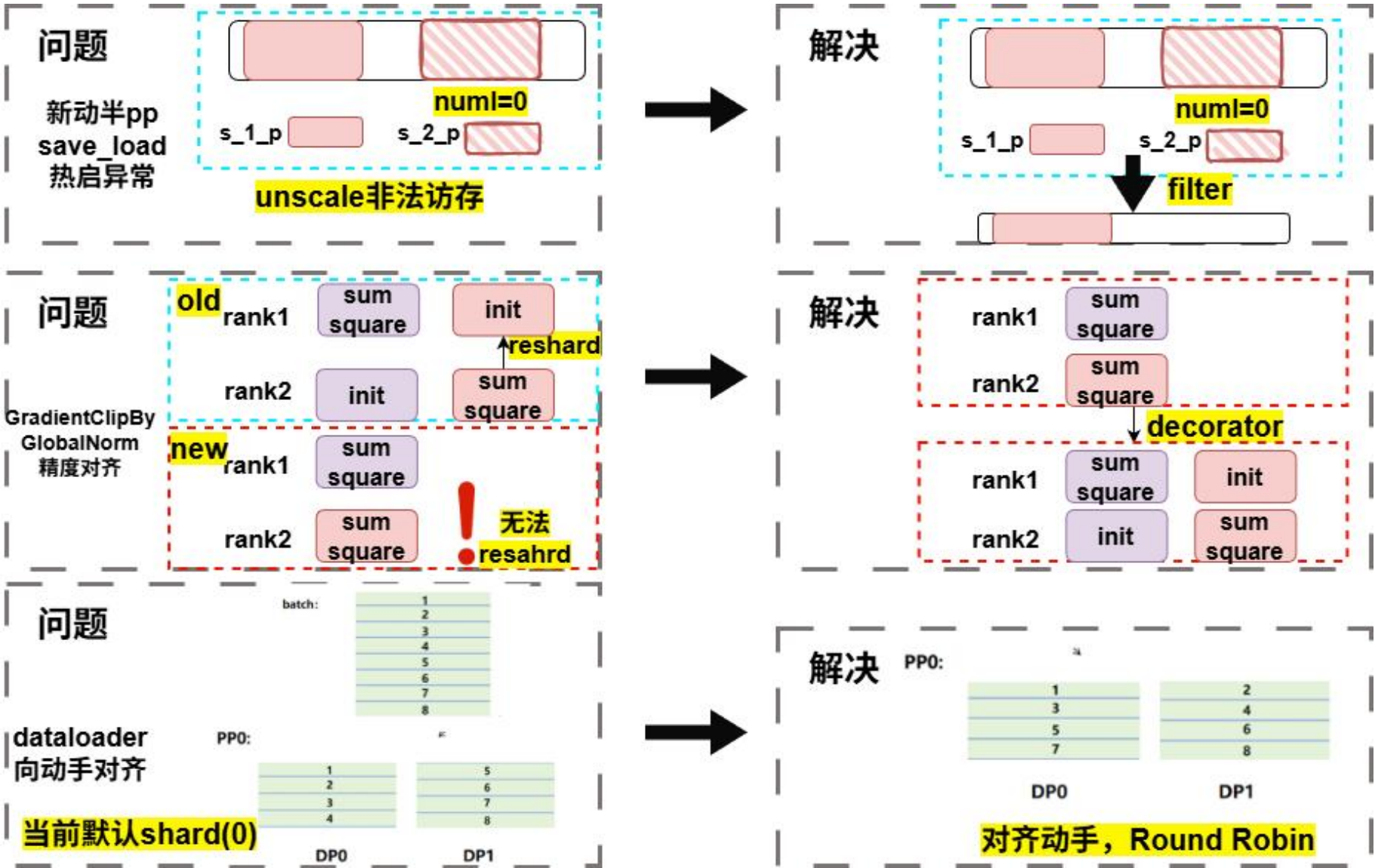




PART2：实际开发工作介绍-自动并行核心模块优化

>>>>>>>>>

➢精度对齐支持



PART2：实际开发工作介绍-API算子的修复与适配

>>>>>>>>>

fused_rotary_position_embedding kernel反向逻辑错误

$$R = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

R为一个逆时针旋转90°矩阵

当前问题：

grad_x = g * cos - rotate_half(g) * sin

正确结果：

grad_x = g * cos - rotate_half(g * sin)

推导说明(简)：

正确的反向公式：设上游梯度 g，R=rotate_half，则:

y = x⊙cos + R(x)⊙sin

dL/dx = g⊙cos − R(g⊙sin)

展开到每对二维（x0,x1）：

正确：dx0 = g0c0 + g1s1；dx1 = g1c1 − g0s0

当前：dx0 = g0c0 + g1s0；dx1 = g1c1 − g0s1

当且仅当s0=s1时，两者相等，而原来的单测就是把sin的前半部分和后半部分数据设为一致，才导致未检测出来

PART2：实际开发工作介绍-API算子的修复与适配

>>>>>>>>

loss函数的size_average和reduce组合映射到reduciton

关键问题：

paddle

```
class CrossEntropyLoss(Layer):

    weight: Tensor | None
    ignore_index: int
    reduction: _ReduceMode
    soft_label: bool
    axis: int
    use_softmax: bool
    label_smoothing: float
    name: str | None
```

pytorch

```
class CrossEntropyLoss(_WeightedLoss):

    def __init__(
        self,
        weight: Optional[Tensor] = None,
        size_average=None,
        ignore_index: int = -100,
        reduce=None,
        reduction: str = "mean",
        label_smoothing: float = 0.0,
```

reduce和size_average可以以位置参数传入

解决思路：

```
_SA0_RD1 = {'size_average': 0, 'reduce': 1}
_SA1_RD2 = {'size_average': 1, 'reduce': 2}
_SA1_RD3 = {'size_average': 1, 'reduce': 3}
_SA3_RD4 = {'size_average': 3, 'reduce': 4}
_SA4_RD5 = {'size_average': 4, 'reduce': 5}
_SA2_RD4 = {'size_average': 2, 'reduce': 4}
```

构建位置index映射并用bool判断是否在以torch格式使用，且特殊情况特殊处理，如CrossEntropyLoss

对wendow函数封装，使其对齐torch

关键问题：

```
@window_function_register.register()
def _hamming(M: int, sym: bool = True, dtype: str = 'float64') -> Tensor:
    """Compute a Hamming window.
    The Hamming window is a taper formed by using a raised cosine with
    non-zero endpoints, optimized to minimize the nearest side lobe.
    """
    return _general_hamming(M, 0.54, sym, dtype=dtype)
```

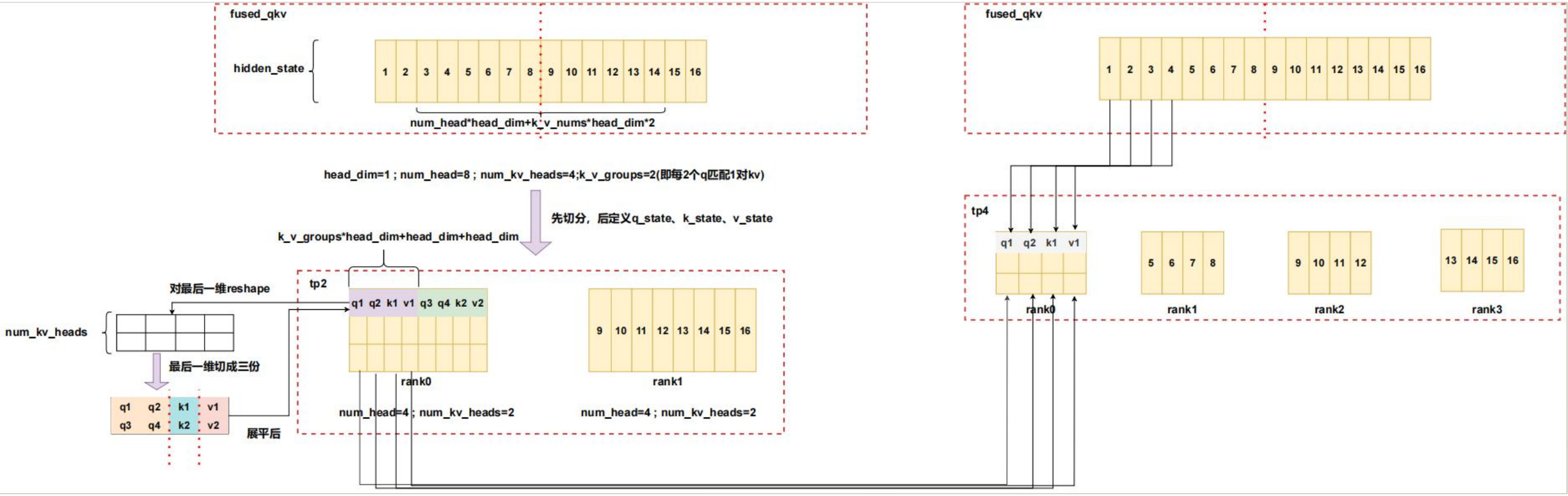
以hamming window为例，get window默认给定了α和β，但需要实现用户自定义且必须调用get_window实现并且不能改变get_window接口

解决思路：

```
w0 = get_window('hamming', w
alpha0, beta0 = 0.54, 0.46
B = beta / beta0
A = alpha - B * alpha0
w = A + B * w0
```

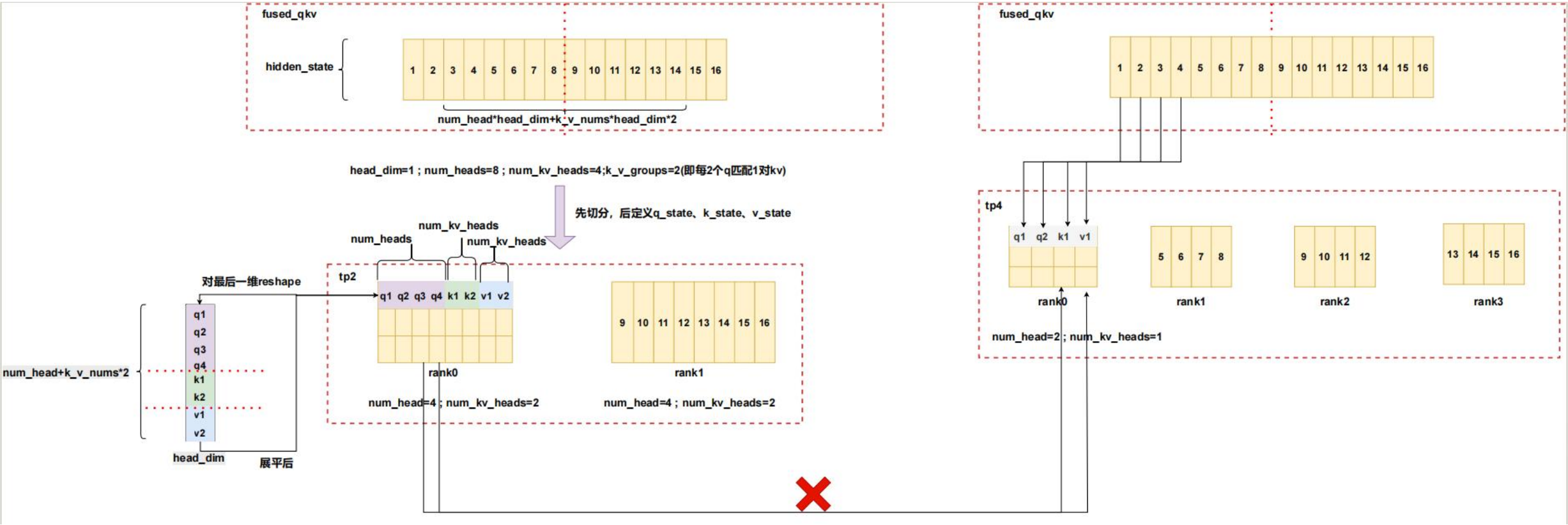
推导逆运算，更新α和β

PART2：实际开发工作介绍-FC背景介绍



fused_qkv(tp2->tp4)　llama　DCP功能介绍

PART2：实际开发工作介绍-FC背景介绍



fused_qkv(tp2->tp4)　ernie4.5　DCP+AOA功能介绍

PART2：实际开发工作介绍-Flexcheckpoint系统研发

FC工作概览

| 8月中旬-9月底

FC验证工作

- **在paddlenlp与paddleformers上适配FC**
- **llama2上验证FC的并行策略互转，并修复存在的问题**
- **ernie4.5上验证FC的并行策略互转，并修复存在的问题**
- **ernie4.5上验证FC与UC的精度对齐，并修复存在的问题**

| 10月初-12月初

FC框架优化工作

- **AOA组件的bug修复**
- **AOA组件的功能拓展**
- **FC框架相关逻辑的适配与优化**

PART2：实际开发工作介绍-Flexcheckpoint系统研发

>>>>>>>>

llama2验证fc(部分重要工作)：

1.验证md5对齐、loss收敛。并均写了复现脚本。

2.发现tp下两套不同的fused_qkv逻辑，推进相关macro的研发

3.解决纯dp会hang住的问题

4.修复使用aoa_config时的问题：macros优先级、关键字过滤、fused_ffn参数长度未对齐

5.撰写llama2上验证Flexcheckpoint框架的运行文档

LLama2

验证方式：

1. ckpt1变换到ckpt2，再变换回ckpt1，md5与原始ckpt对齐，且均可正常训练，loss收敛或逐位对齐
2. ckpt1变换到ckpt2，再分别合成开源格式权重，两份开源权重md5对齐，且均可正常推理，loss逐位对齐

目前先验第一点即可：

loss 收敛趋势图需要用 ckpt1 训 50 个 step， load 成 ckpt 2 继续训 200 个 step，loss 图用不同颜色区分。

- 1e-5 表示：MD5 校验通过，续训的 loss diff 精度误差在 1e-5 以内
- [✅] 表示：MD5 校验通过，续训的 loss 逐位对齐

	dp4	sd4(v1)	sd4(v2)	tp4	pp4	vpp4	sd2+tp2	tp2+pp2
dp2	1E-5	1E-5	1E-4	1E-5	1E-5	1E-5	1E-5	1E-5
sd2 (v1)	1E-5	1E-5	1E-4	1E-5	1E-5	1E-5	1E-5	1E-5
sd2 (v2)	1E-4	1E-4	1E-4	1E-4	1E-4	1E-4	1E-4	1E-4
tp2	1E-5	1E-5	1E-4	1E-4	1E-4	1E-5	1E-4	✅
pp2	1E-5	1E-5	1E-4	1E-4	✅	✅	1E-4	1E-4
vpp4	1E-5	1E-4	1E-5	1E-5	✅	✅	1E-4	1E-4
sd2+tp2	1E-5	1E-5	1E-5	1E-4	1E-5	1E-4	✅	1E-4
tp2+pp2	1E-5	1E-5	1E-5	1E-5	1E-5	1E-5	1E-4	✅

PART2：实际开发工作介绍-Flexcheckpoint系统研发

>>>>>>>>

ernie4.5验证fc-1：

1.验证md5对齐、loss收敛。并均写了复现脚本。

2.验证tp2(ep2)->tp4的gate参数未对齐，非fc框架的问题，而是gate绕过opt更新

3.tp2(ep2)->pp4场景下，发现未适配共享参数的场景，做了适配

4.修复MPScale.apply未考虑bias为None的问题

5.DP2、ShardingV1、V2转TP2(EP2)_PP2等出现src_var和dst_var的global_size对不齐的现象，经验证，主要是Ernie下的SequenceParallelLayer未适配sharded_state_dict，做了适配。

Ernie4.5 纯文

验证方式：

1. ckpt1变换到ckpt2，再变换回ckpt1，md5与原始ckpt对齐，且均可正常训练，loss收敛或逐位对齐
2. ckpt1变换到ckpt2，再分别合成开源格式权重，两份开源权重md5对齐，且均可正常推理，同时验证与UC的权重md5对齐。

Method_1

- Method_1对应验证方式第1点
- 1E-5 表示：MD5 校验通过，续训的 loss diff 精度误差小于 1E-5
- [✅] 表示：MD5 校验通过，续训的 loss 逐位对齐

	dp4	sd4(v1)	sd4(v2)	tp4	pp2	sd2(ep2)	sd2+tp2(ep2)	tp2(ep2) +pp2
dp2	1E-3	1E-3	1E-3	1E-2	1E-2	1E-6	1E-4	1E-2
sd2 (v1)	1E-3	1E-3	1E-3	1E-2	1E-2	1E-6	1E-5	1E-2
sd2 (v2)	1E-3	1E-3	1E-3	1E-2	1E-2	1E-5	1E-5	1E-2
tp2 (ep2)	1E-2	1E-2	1E-2	1E-5	1E-2	1E-2	1E-2	1E-2
pp2	1E-2	1E-2	1E-2	1E-2	✅	1E-2	1E-2	1E-2
sd2(ep2)	1E-3	1E-3	1E-3	1E-2	1E-2	✅	✅	1E-2
sd2+tp2(ep2)	1E-2	1E-2	1E-2	1E-2	1E-2	1E-5	✅	1E-2
tp2(ep2) +pp2	1E-2	1E-2	1E-2	1E-2	1E-2	1E-2	1E-2	1E-2

PART2：实际开发工作介绍-Flexcheckpoint系统研发

>>>>>>>>

ernie4.5验证fc-2：

1.验证FC与UC的精度对齐，并均写了复现脚本。

2.sd2(ep2)的md5无法对齐，因为UC未对expert_id做偏移，处理后md5对齐

3.TP相关均无法对齐：ernie_moe的tp_mappings未对mtp_block层做映射，load_state_dict和_handle_aoa未考虑到多卡转单机，已做适配。

Method_2

- Method_2对应验证方式第2点，合参与UC精度对齐
- [✅] 表示：MD5 校验通过
- [❌] 表示：MD5 校验失败

	dp4	sd4(v1)	sd4(v2)	tp4	pp2	sd2(ep2)	sd2+tp2(ep2)	tp2(ep2) +pp2
dp2	✅	✅	✅	✅	✅	✅	✅	✅
sd2 (v1)	✅	✅	✅	✅	✅	✅	✅	✅
sd2 (v2)	✅	✅	✅	✅	✅	✅	✅	✅
tp2 (ep2)	✅	✅	✅	✅	✅	✅	✅	✅
pp2	✅	✅	✅	✅	✅	✅	✅	✅
sd2(ep2)	✅	✅	✅	✅	✅	✅	✅	✅
sd2+tp2(ep2)	✅	✅	✅	✅	✅	✅	✅	✅
tp2(ep2) +pp2	✅	✅	✅	✅	✅	✅	✅	✅

PART2：实际开发工作介绍-Flexcheckpoint系统研发

>>>>>>>>

FC框架优化：

1.问题修复：

get_num_hidden_layer、star_macro字段匹配错误；add、remove、rename一些边界功能未完善；full param的cast异常；build_input_vars中dtype覆盖问题。

2.功能扩展：

添加专家ID的macro；优化ID相关macro逻辑；统一使用id_macro；新增get_var_mapping_chain_macro，使得src与dst切分信息可追溯；支持merge_sharded_state_dict的单卡运行；支持自动推导aoa_statemens的逆过程；为Sharding Optimizer Stage3 适配fc。

3.单测补充：

补充整个macro单测并覆盖AOAShardInfoContext类的测试；为aoa添加add、cast以及混合多种原语的单测；为load hf checkpoint添加单测；为full param添加cast单测；为DynamicShardingOptimizerV1,V2，Adawm添加sharded_state_dict单测

PART3：未来工作规划

>>>>>>>>>

FC框架优化

- 当前FC在实际使用时，用户提出了一系列诉求如下，需要优化：

序号	任务描述	是否完成 ✅	对应 PR 链接
0	权重加载时报错信息过少 & 当权重与模型 state_dict 不匹配时，缺少对 missing keys 和 unexpected keys 的报错信息		
1	`save_pretrained` 每个卡把收集来的 tensor 暂存在显存上，改成暂存在 CPU 上		
2	在PaddleFormers中，模型load/save配置了 `flex_checkpoint` 但实际没有 `aoaconfig`，需要拦截	✅	#3055
3	`load` 和 `save` 需要写两份 `aoaconfig`，是否可以简化	✅	#3055
4	当前 `aoaconfig` 写法较为复杂且不同模型写法大多数部分类似，是否后续可以简化		
5	文档中缺少对模型为 tie weight 和 qkv fused（phi4 模型）的补充文档	✅	
6	处理 safetensor 文件缺失或 safetensor 文件更改场景		

PART4：工作总结

>>>>>>>>

1.合入25个Pr。

2.提交136个FC精度对齐测试脚本供复现。并在测试中记录一系列ernie、UC、FC的问题，为其它同事提供适配经验。脚本仓库：https://github.com/PFCCLab/FlexCheckpointVerf

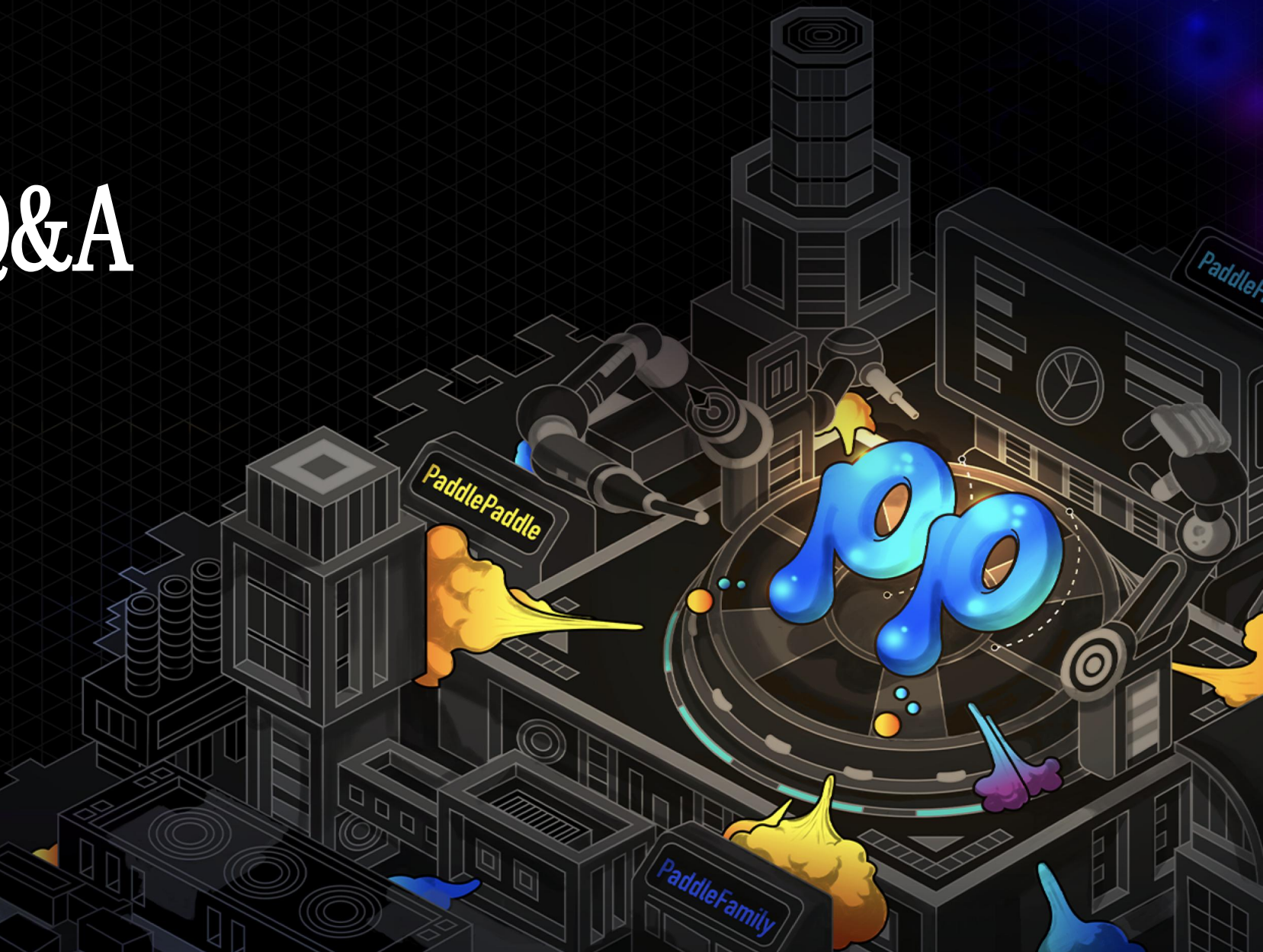
3.总结了一个93面的FlexCheckpoint的工作记录文档

4.总结一份24面DygraphShardingOptimizer的学习分享(暂未分享)

5.参加wave_summit，作为开发者分享

6.担任护航计划助教，负责组织周报提交以及经验分享会

Thanks! Q&A



答辩人：郑天宇 / zty-king
指导人：陈锐彪 / From00
飞桨护航计划集训营