金枪鱼之夜
MegEngine架构设计

MEGVII 旷视

主讲人介绍

MEGVII 旷视

许欣然

旷视研究院 - Engine 负责人

2015年正式加入旷视

- **现负责 MegEngine 开源项目**
- **曾负责 Brain++ 集群建设、安防 Core 系统**
- **建立旷视内部 DPFlow、Nori 等系统**

主讲人介绍

MEGVII 旷视

许欣然

旷视研究院 - Engine 负责人

- 清华计算机系 0 字班，Learn Helper 创造者

主讲人介绍

MEGVII 旷视

许欣然

旷视研究院 - Engine 负责人

- **清华计算机系 0 字班，Learn Helper 创造者**
- **研究生退学...**

How to quit school scientifically（DIY version）

由 许欣然创建, 最终由 未知用户 (weiming)修改于 一月 21, 2016

1. 填写研究生退学申请表（可到info下方的“表格下载”：“研究生院”下载），需要填写个人退
2. ~~每年3/6/9/12月各有一天~~（现在随时均可）会办理退学（具体时间在表格上有写），在这个
3. 接到教务电话后，拿到“关于同意XXX退学的通知”，完成“研究生退学离校手续单”（此时宿
   - ”交还公费医疗证“请去校医院，先去校医院北楼三层最里侧的财务室盖章，然后到挂
   - “归还向系借用的钱款”请到主楼三层 东区财务室办理，找其中的一位老爷爷（吕老师

MegEngine架构设计

MEGVII 旷视

大纲：

- 背景介绍
- 如何写出一个深度学习框架？（超简化版）
- 一个陈年静态图框架是怎么变成动态图框架的？（蛋疼的渐进式演进）
- 对未来的展望
- 相关资源

救救孩子吧！

献出你的爱心，帮我们点个 Star

github.com/MegEngine/MegEngine

MEGVII 旷视

1 背景介绍

2 如何写出一个深度学习框架？

3 一个陈年静态图框架是怎么变成动态图框架的？

4 对未来的展望

5 相关资源

深度学习框架是干啥的？

MEGVII 旷视

训练：Python 代码（动态 / 静态）

```python
class LeNet(M.Module):
    def forward(self):
        x = self.pool1(self.relu1(self.conv1(x)))
        x = self.pool2(self.relu2(self.conv2(x)))
        x = F.flatten(x, 1)
        x = self.relu3(self.fc1(x))
        x = self.relu4(self.fc2(x))
        x = self.classifier(x)

lenet = LeNet()

optimizer = optim.SGD(le_net.parameters(), lr=1.05)
logits = le_net(data)
loss = F.cross_entropy_with_softmax(logits, label)
optimizer.backward(loss)
```

推理：C++ 代码（载入 / 搭建）

```cpp
auto loader = serialization::GraphLoader::make(std::move(inp_file));
serialization::GraphLoader::LoadResult network =
 loader->load(config, false);

HostTensorND predict;

std::unique_ptr<cg::AsyncExecutable> func =
      network.graph->compile({make_callback_copy(
         network.output_var_map.begin()->second, predict)});

func->execute();
func->wait();
```

框架

第三方 kernel：cuDNN / MKL

```cpp
cudnn_check(cudnnBatchNormalizationForwardTraining(
        handle, m_tensor_desc.bn_mode,
        &alpha, &beta,
        m_tensor_desc.xy_desc.desc,     // xDesc
        src.raw_ptr,                     // x
        m_tensor_desc.xy_desc.desc,     // yDesc
        dst.raw_ptr,                     // y
        m_tensor_desc.param_desc.desc,  // bnScaleBiasMeanVarDesc
        bn_scale.raw_ptr, bn_bias.raw_ptr, m_param.avg_factor,
        mean.raw_ptr, variance.raw_ptr, m_param.epsilon,
        batch_mean.raw_ptr, batch_inv_variance.raw_ptr));
```

手写 kernel：CUDA ARM neon x86 AVX

```cpp
        for (uint32_t fw = 0; fw < FW; ++fw)
        {
            uint32_t iw;
            if (is_xcorr) iw = ow*SW + fw - PW; else iw = ow*SW + (FW-fw-1) - PW;
            if (iw < IW)
            for (uint32_t ico = 0; ico < ICb; ++ico) {
                uint32_t fid = op*IC*FH*FW*OC + (ic+ico)*FH*FW*OC +
                    fh*FW*OC + fw*OC + oc;
                float fval = filter[fid];
                float src_reg[Ns];
#pragma unroll
                for (uint32_t no = 0; no < Ns; ++no) {
                    src_reg[no] = src_cache[no*ICs*IH*IW + ico*IH*IW + ih*IW + iw];
                }
#pragma unroll
                for (uint32_t no = 0; no < Ns; ++no) {
                    dst_reg[no] += src_reg[no]*fval;
                }
            }
        }
        }
        __syncthreads();
    }
```

天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

深度学习框架是干啥的？

MEGVII 旷视

训练

Python 代码

（动态 / 静态）

框架

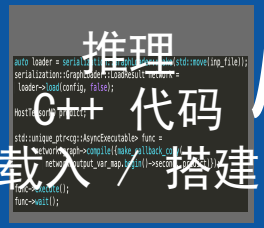
Kernel

```
cudnn_check(cudnnBatchNormalizationForwardTraining(
        handle, m_tensor_desc.bn_mode,
        &alpha, &beta,
        m_tensor_desc.xy_desc.desc,     // xDesc
        src.raw_ptr,                    // x
        m_tensor_desc.xy_desc.desc,     // yDesc
        dst.raw_ptr,                    // y
        m_tensor_desc.param_desc.desc,  // bnScaleBiasMeanVarDesc
        bn_scale.raw_ptr, bn_bias.raw_ptr, m_param.avg_factor,
        mean.raw_ptr, variance.raw_ptr, m_param.epsilon,
        batch_mean.raw_ptr, batch_inv_variance.raw_ptr));
```

推理

C++ 代码

（载入 / 搭建）



天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

深度学习框架是干啥的？

MEGVII 旷视

动态训练

```
        x = F.flatten(x, 1)
        x = self.relu1(self.fc1(x))
        x = self.relu2(self.fc2(x))
        x = self.classifier(x)

lenet = LeNet()

optimizer = optim.SGD(le_net.parameters(), lr=1.05)
logits = le_net(data)
loss = F.cross_entropy_with_softmax(logits, label)
optimizer.backward(loss)
```

通用组件　Optimizer、Loss、Data、数值稳定的算子

Python 接口

自动求导　Shape 推导

显存池

计算图

推理

静态训练

```
auto loader = serialization::GraphLoader::make(std::move(inp_file));
serialization::GraphLoader::LoadResult network =
 loader->load(config, false);
Host
std::unique_ptr<cg::AsyncExecutable> func =
        network.graph->compile({make_callback_copy(
            network.output_var_map.begin()->second, predict)});

func->execute();
func->wait();
```

Kernel

```
cudnn_check(cudnnBatchNormalizationForwardTraining(
        handle, m_tensor_desc.bn_mode,
        &alpha, &beta,
        m_tensor_desc.xy_desc.desc,     // xDesc
        src.raw_ptr,                    // x
        m_tensor_desc.xy_desc.desc,     // yDesc
        dst.raw_ptr,                    // y
        m_tensor_desc.param_desc.desc,  // bnScaleBiasMeanVarDesc
        bn_scale.raw_ptr, bn_bias.raw_ptr, m_param.avg_factor,
        mean.raw_ptr, variance.raw_ptr, m_param.epsilon,
        batch_mean.raw_ptr, batch_inv_variance.raw_ptr));
```

天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

深度学习框架是干啥的？

MEGVII 旷视

通用组件　Optimizer、Loss、Data、数值稳定的算子

Python 接口

异步执行器

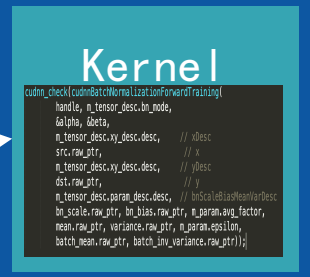
自动求导　Shape 推导

显存池

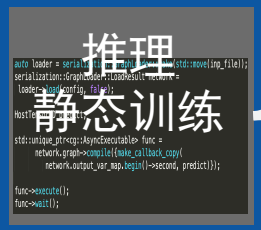
Kernel 选择

Kernel 执行引擎

Remote IO



计算图



天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

深度学习框架是干啥的？

MEGVII 旷视

通用组件　Optimizer、Loss、Data、数值稳定的算子

Python 接口

异步执行器

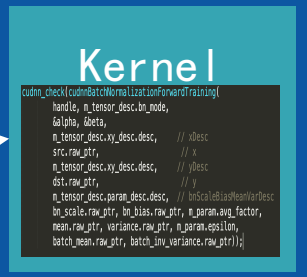
自动求导　Shape 推导

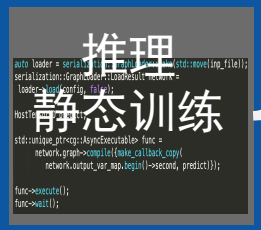
显存池

Kernel 选择

Kernel
执行引擎

Remote IO





计算图

图优化

生成计算序列
显存复用
生成显存方案

JIT 引擎

天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

深度学习框架是干啥的？

MEGVII 旷视

解决以下问题

- **提供高层原语：算子、求导、优化器**
- **隐藏设备细节**
- **各类自动优化，提供最优性能**

天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

道理我都懂，为什么又搞一个深度学习框架？

MEGVII 旷视

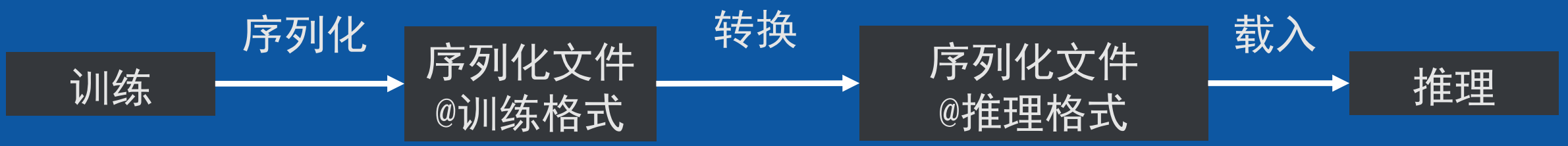
- 历史原因
  - 2014~2015：Theano 和 Caffe 都不能打
  - 2015~2016：TF 性能不太行
  - 2016~至今：为了吃到训推一体的优势

- 你们为啥不用 PyTorch / TensorFlow？
  - 要相信创新才能提升社会效率
  - 沉没成本
  - 性能优势
  - 推理流程简单
  - 支持的硬件设备

https://xkcd.com/303/

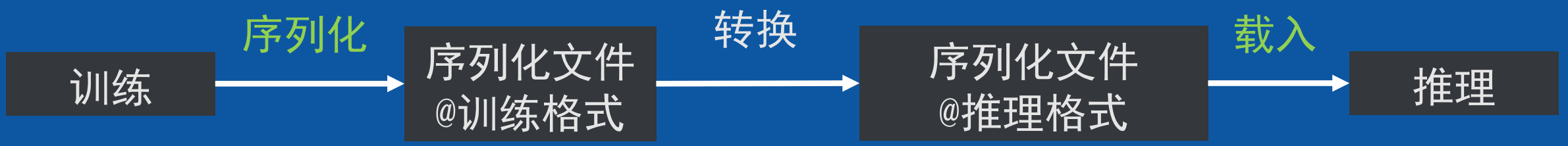
训推一体是个啥玩意？

MEGVII 旷视



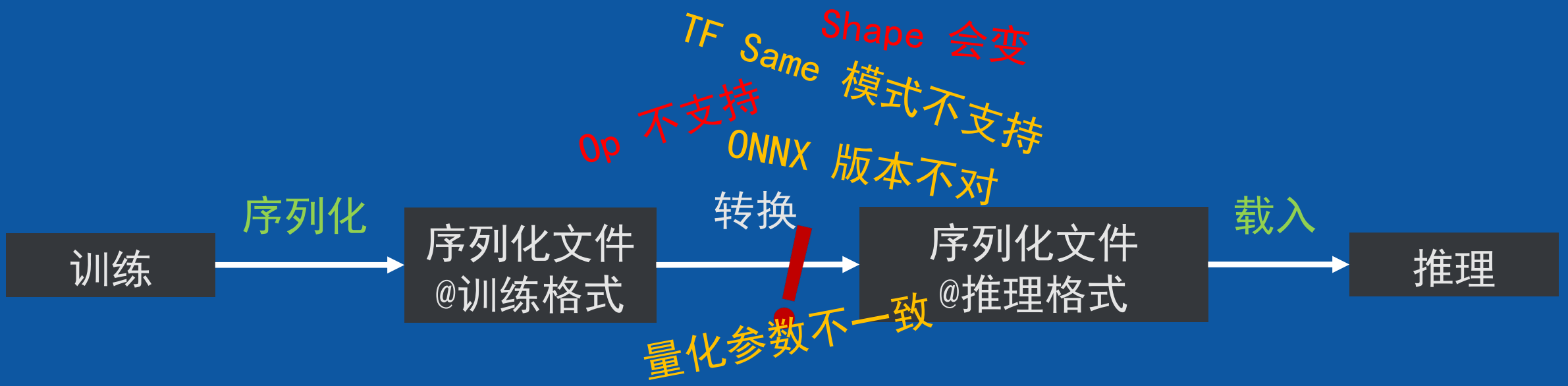
训推一体是个啥玩意？

MEGVII 旷视



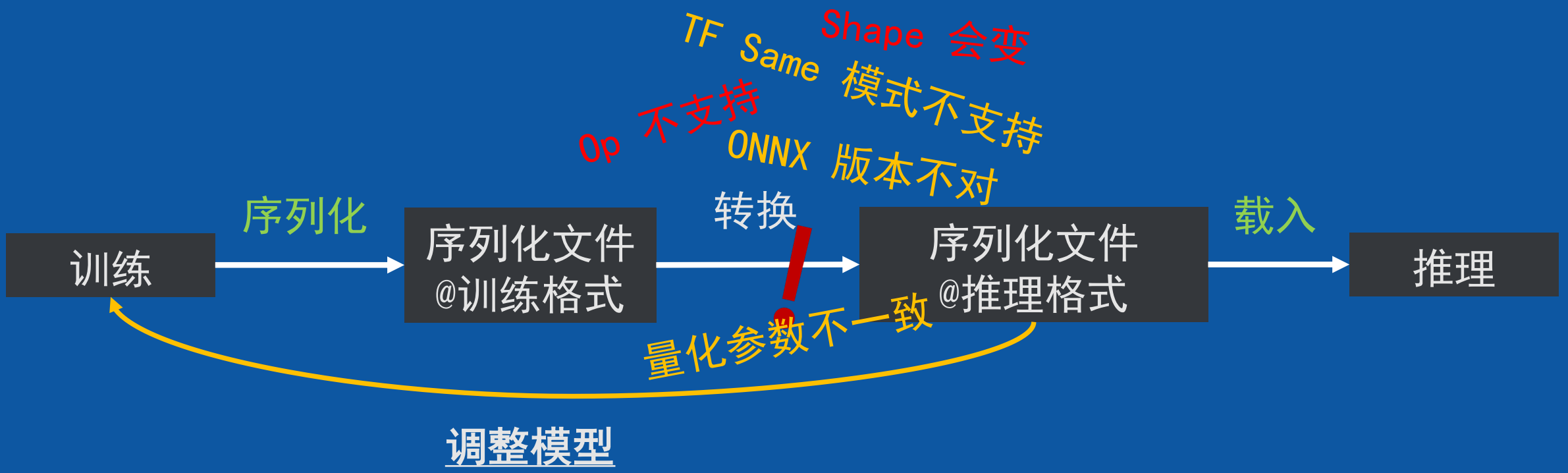
训推一体是个啥玩意？

MEGVII 旷视



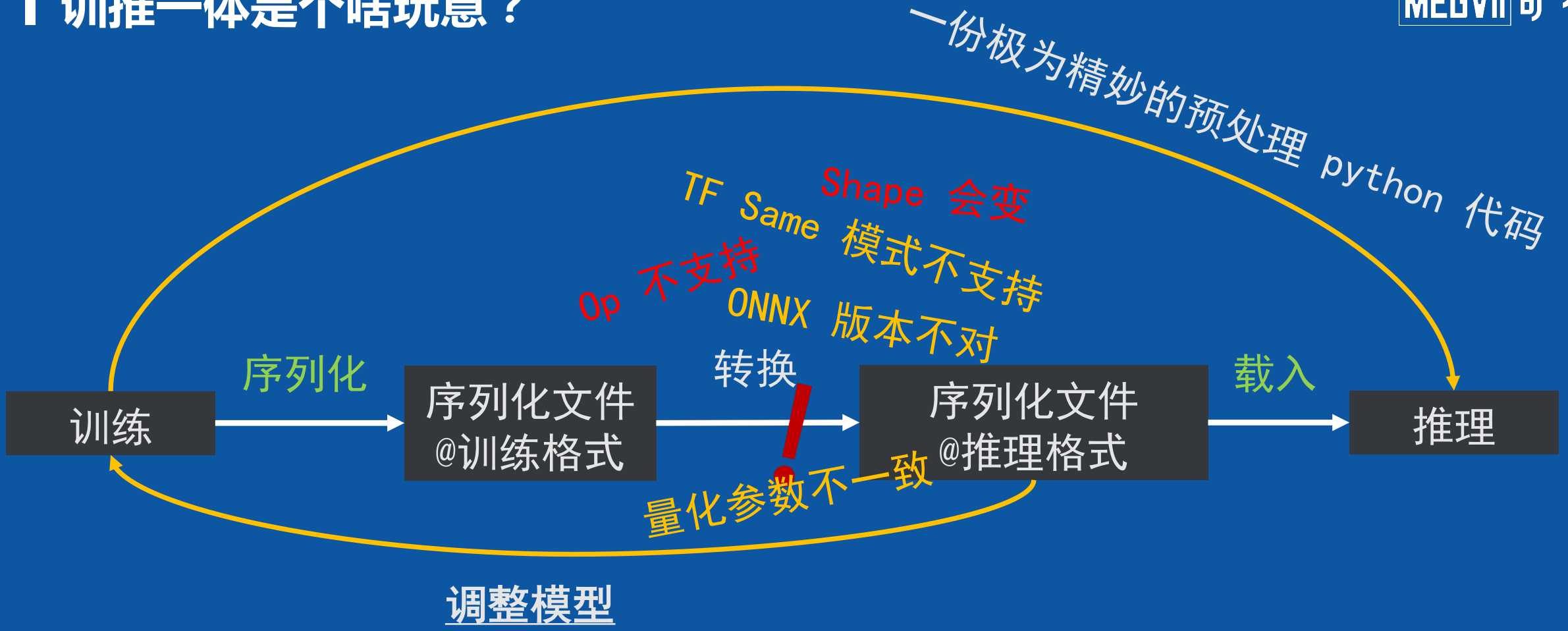
训推一体是个啥玩意？

MEGVII 旷视



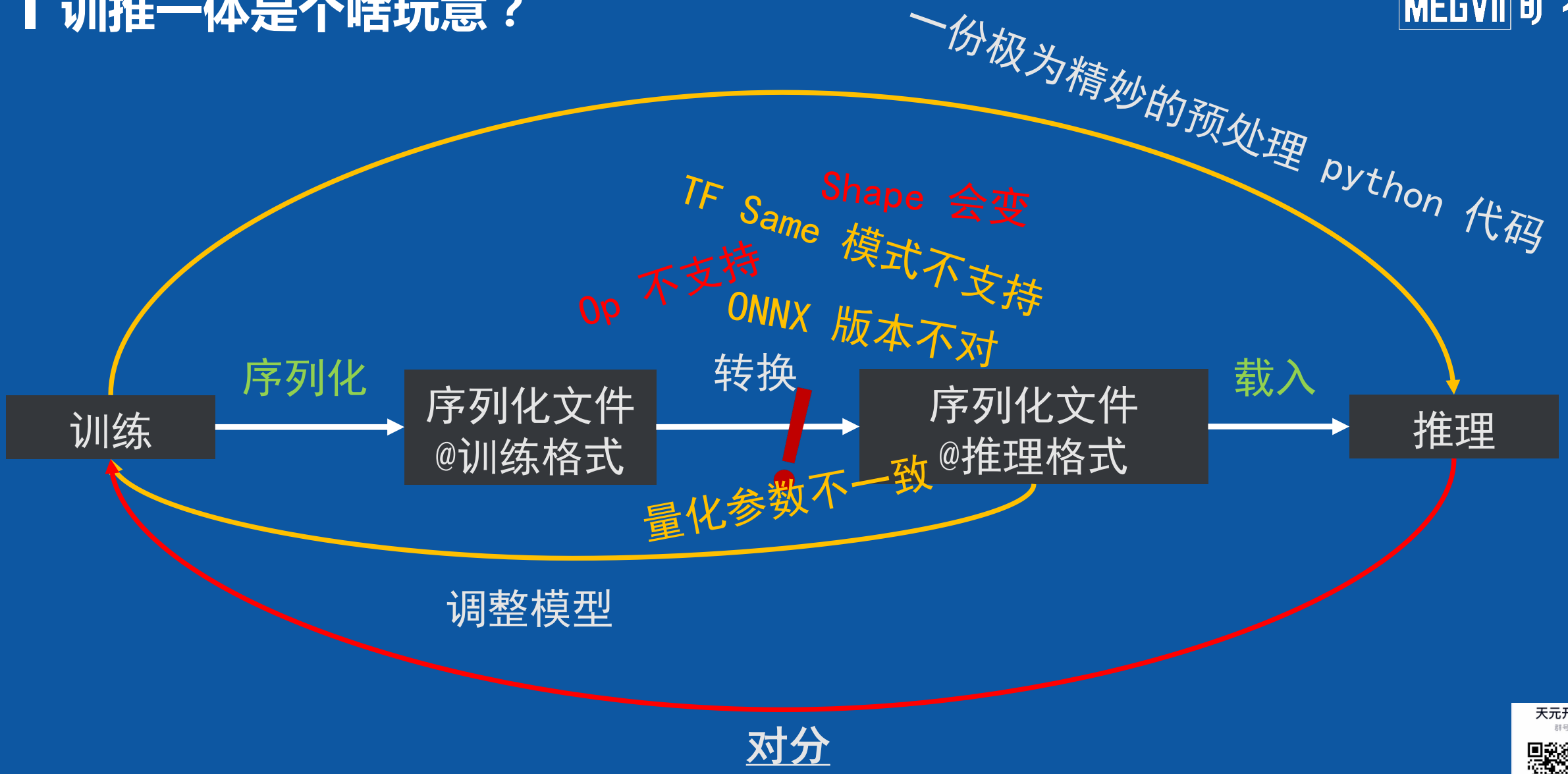
训推一体是个啥玩意？

MEGVII 旷视



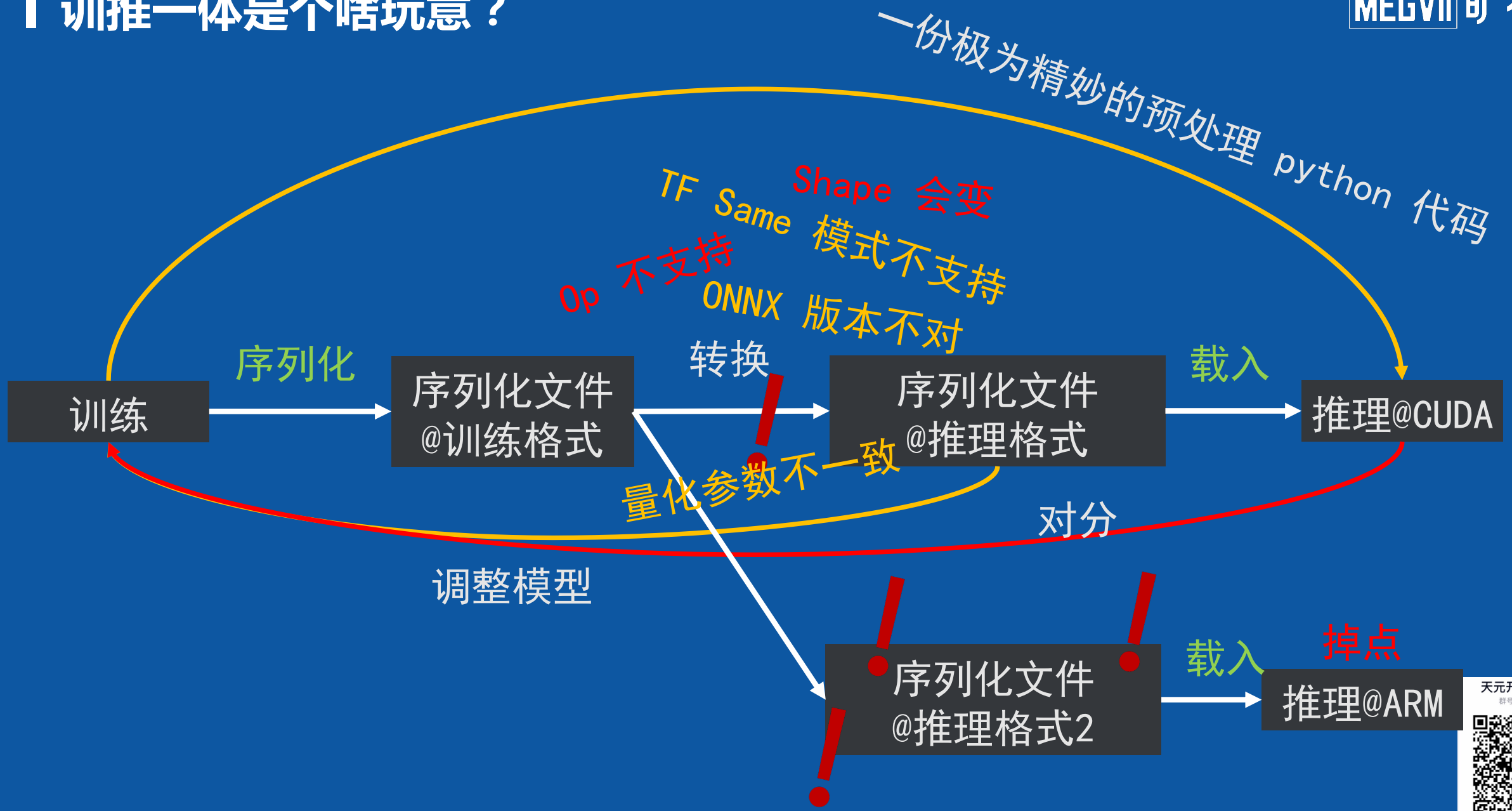
训推一体是个啥玩意？

MEGVII 旷视



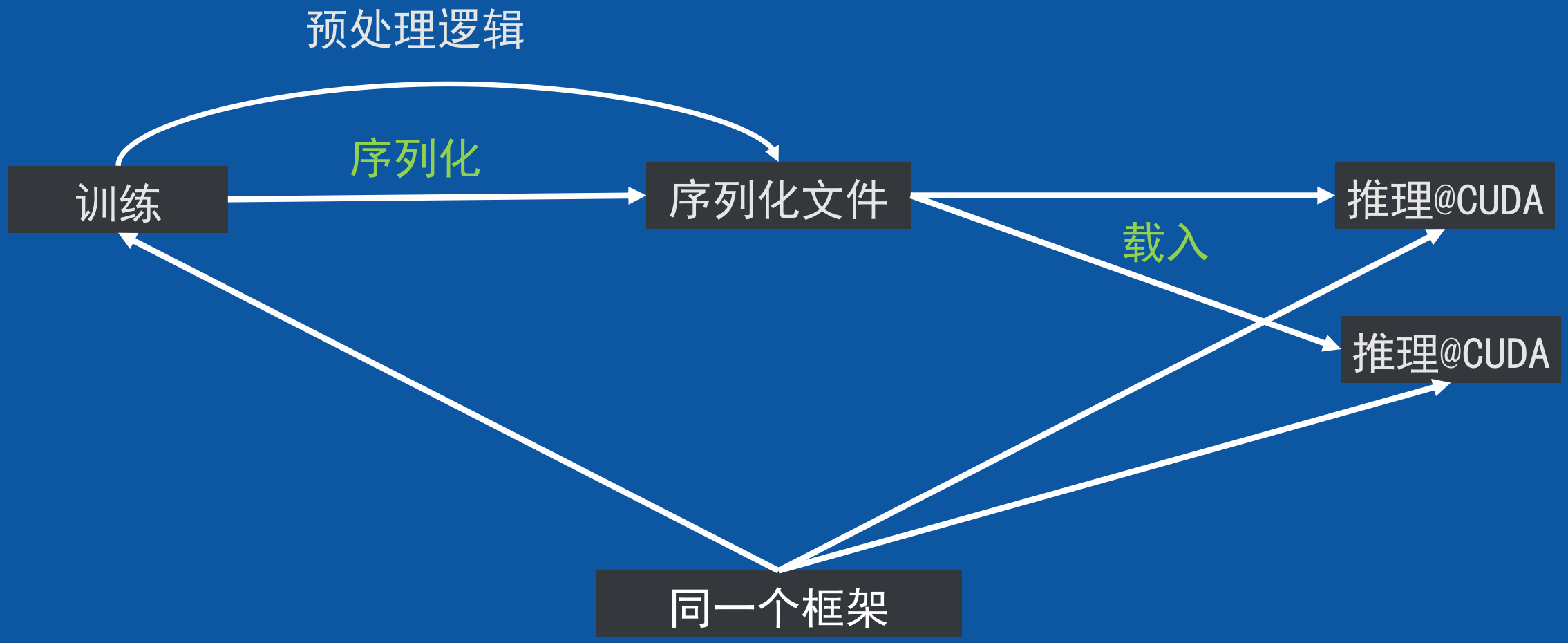
训推一体是个啥玩意？

MEGVII 旷视



训推一体是个啥玩意？

MEGVII 旷视



训推一体是个啥玩意？这不是很容易？

MEGVII 旷视

- 加一个算子要所有平台对齐
  - 不同平台的精度问题
  - 某些平台指令有限制
  - 开发成本高
  - 反复的讨论
- 如何让一个几十万行的代码库，能跨平台跨系统编译
  - linux / windows / macos / TEE / ios / Android
  - X86 / CUDA / ARM / MIPS / RISC-V
- 代码裁剪控制代码体积
- 如何同时在算力极大、极小两个计算的计算场景下都表现良好

MEGVII 旷视

1 背景介绍

2 如何写出一个深度学习框架？

3 一个陈年静态图框架是怎么变成动态图框架的？

4 对未来的展望

5 相关资源

如何写出一个深度学习框架？

MEGVII 旷视

此处应有一张 xkcd 风格的图

天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

动态 or 静态

MEGVII 旷视

动态：

- 用户调试方便，学习成本低
- 框架管的事情少
- 框架性能调优难，显存优化难
- 显存复用、多机防死锁需要老师傅

静态：

- 学习成本高，约等于学一门新语言，报错难懂
- 框架管的事情多，要造世界：**debugger、profiler、visualizer…**
- 性能调优容易，自动显存优化效果好
- 图机制方便自动发现坑，显存复用、多机防死锁可以框架解决
- 难以抽象，难以形成库

天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

静态图报错

MEGVII 旷视

```
megengine._internal.exc.MegBrainError: MegBrain core throws exception: mgb::MegBrainError
16 async errors recorded; first msg: invalid IndexingOneHot: offset=96 idx0=96 indexer=-1 idx2=0
+ bt:/home/megvii/.local/lib/python3.6/site-packages/megengine/_internal/_mgb.cpython-36m-x86_64-linux-gnu.so{1da38b9,1df823b,1d
| Associated operator: id=3173161 name=indexing_one_hot(:cls_loss_fct:Sub2253)[3173161] type=mgb::opr::IndexingOneHot
|   input variables:
|     0: {id:2976507, layout:{128(129388),129388(1)}, Float32, owner:SUB(:cls_loss_fct:Sub2097,:cls_loss_fct:cls_loss_fct:ReduceM
0, s, 4, 8}
|     1: {id:3173093, shape:{128}, Int32, owner:axis_add_rm(argmax[3173089])[3173092]{AxisAddRemove}, name::Argmax2361, slot:0, g
|   output variables:
|     0: {id:3173162, shape:{128,1}, Float32, owner:indexing_one_hot(:cls_loss_fct:Sub2253)[3173161]{IndexingOneHot}, name:indexi
|     1: {id:3173163, layout:{0(1)}, Byte, owner:indexing_one_hot(:cls_loss_fct:Sub2253)[3173161]{IndexingOneHot}, name:indexing_
|
| Unoptimized equivalent of associated operator: id=81671 name=indexing_one_hot(:cls_loss_fct:Sub2253)[81671] type=mgb::opr::Inde
|   input variables:
|     0: {id:77861, shape:{128,129388}, Float32, owner:SUB(:cls_loss_fct:Sub2097,:cls_loss_fct:cls_loss_fct:ReduceMax2244@gpu1:0)
|     1: {id:81670, shape:{128}, Int32, owner:axis_add_rm(argmax[81666])[81669]{AxisAddRemove}, name::Argmax2361, slot:0, gpu1:0,
||  output variables:
|     0: {id:81672, shape:{128,1}, Float32, owner:indexing_one_hot(:cls_loss_fct:Sub2253)[81671]{IndexingOneHot}, name:indexing_o
|     1: {id:81673, shape:{}, Byte, owner:indexing_one_hot(:cls_loss_fct:Sub2253)[81671]{IndexingOneHot}, name:indexing_one_hot(:
```

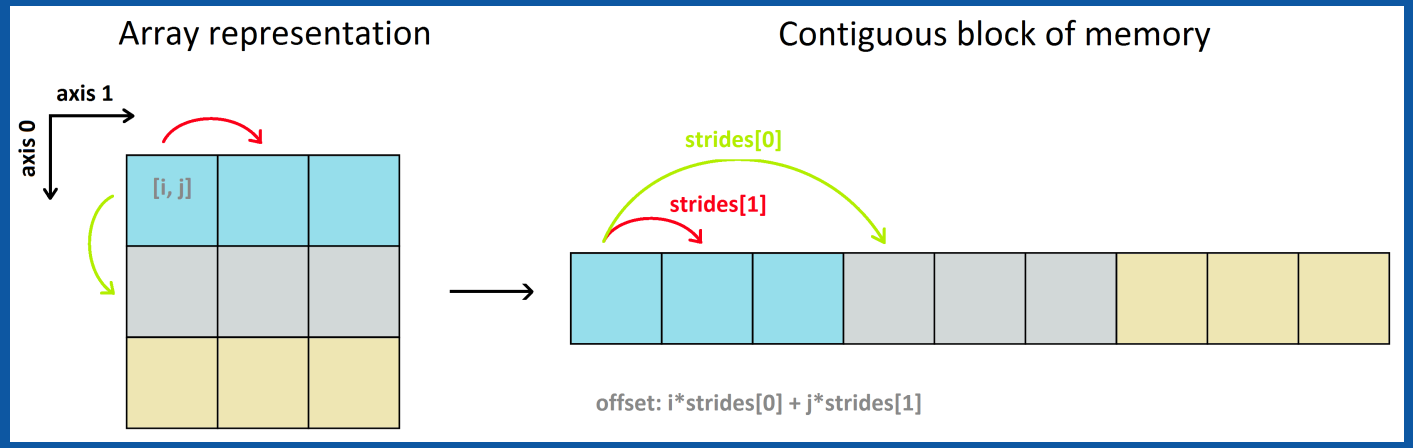
Tensor

MEGVII 旷视

- 承载数据（动态）或 Symbol（静态）

```
class Tensor {
    void*       raw_ptr;
    DType       dtype;
    size_t      shape[MAX_NDIM];
    ptrdiff_t   stride[MAX_NDIM];
    CompNode    node;
}
```

- 实现 lazy 操作
  - boardcast
  - reverse
  - dimshuffle



天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

Kernel

MEGVII 旷视

```
void conv_forward      (Tensor filter, Tensor src,   Tensor* dst, void* workspace, params...)
void conv_backward_data(Tensor filter, Tensor* grad, Tensor diff, void* workspace, params...)
void conv_backward_data(Tensor* grad,  Tensor src,   Tensor diff, void* workspace, params...)
```

动态图训练

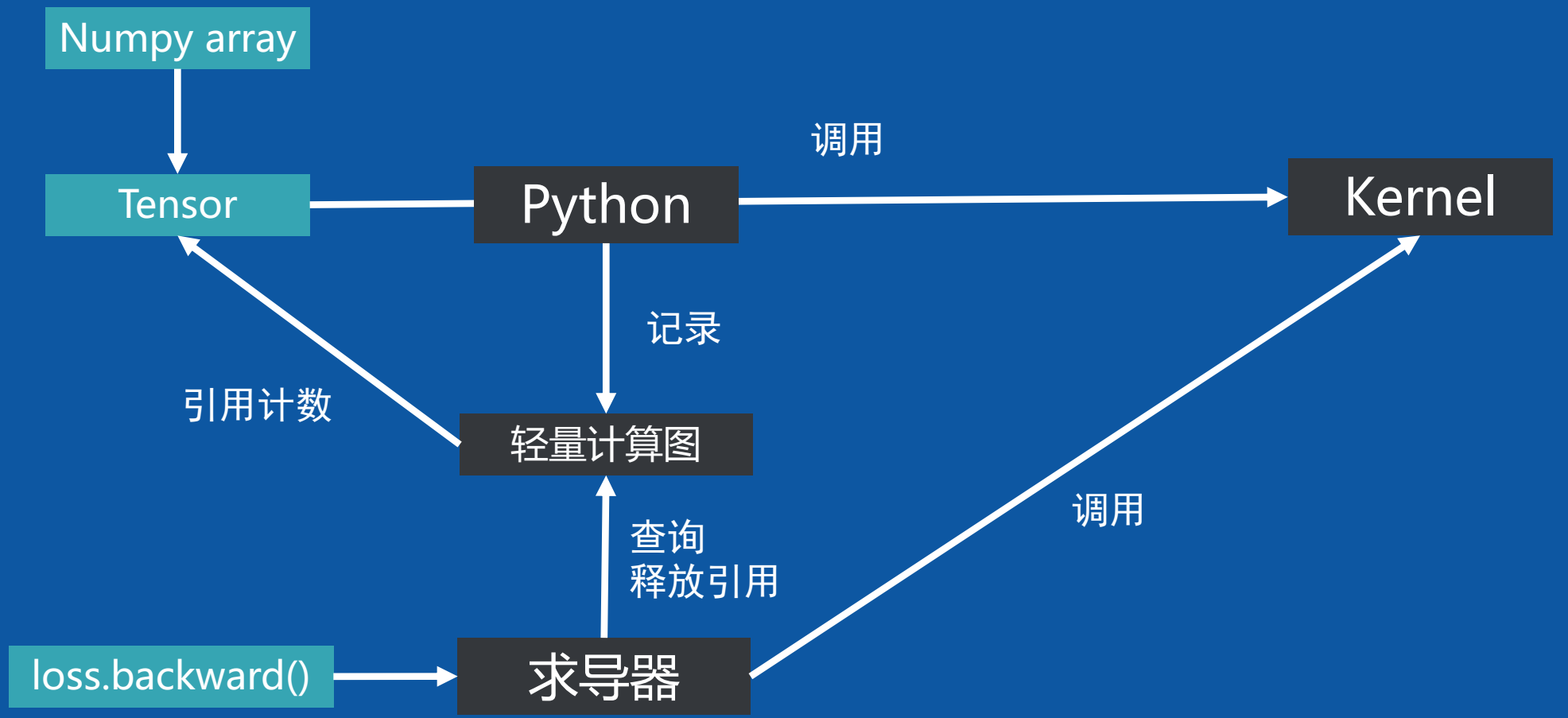
MEGVII 旷视

天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

动态图训练 – 自动求导

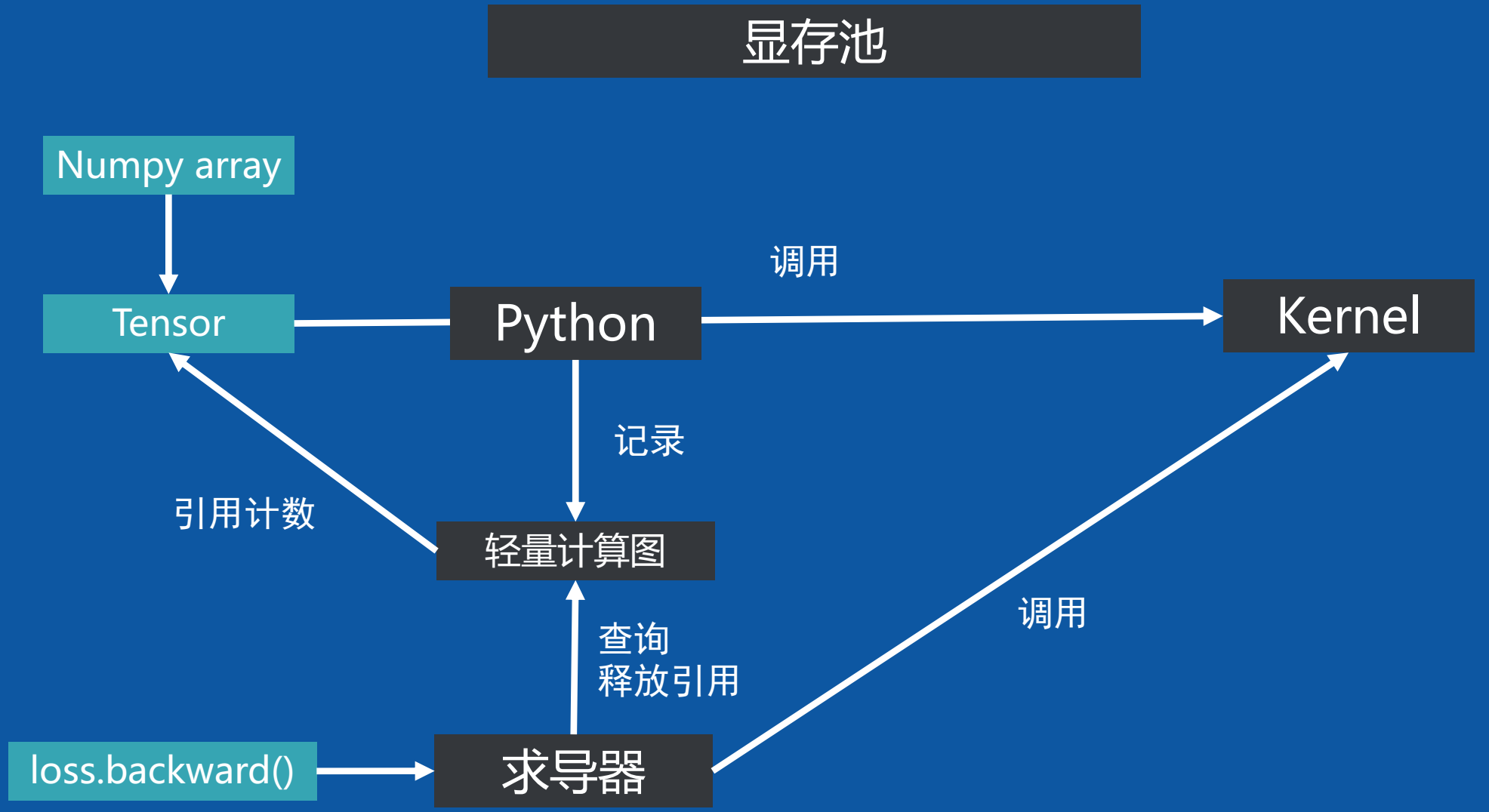
MEGVII 旷视



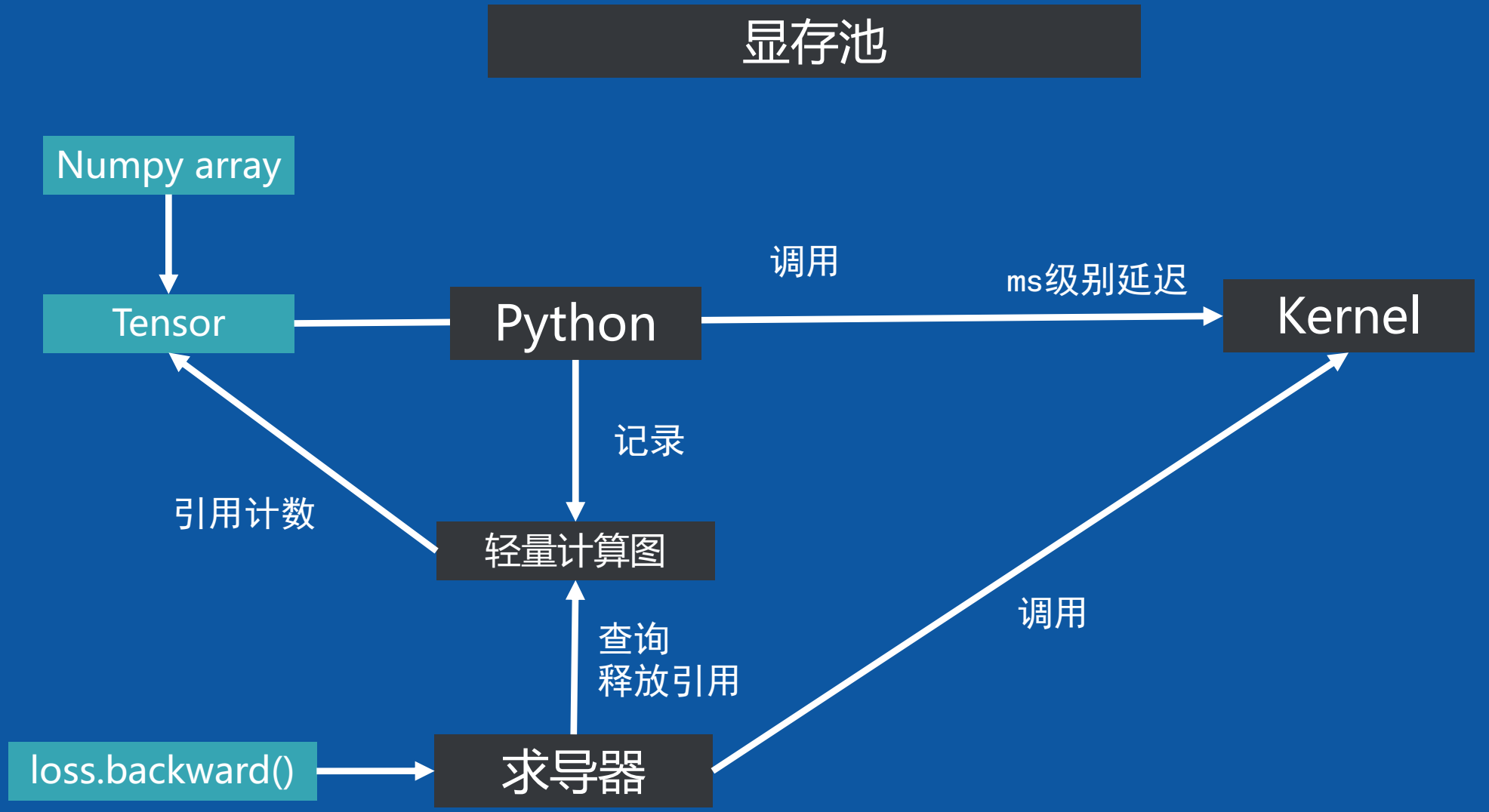
天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

动态图训练 – 提升性能

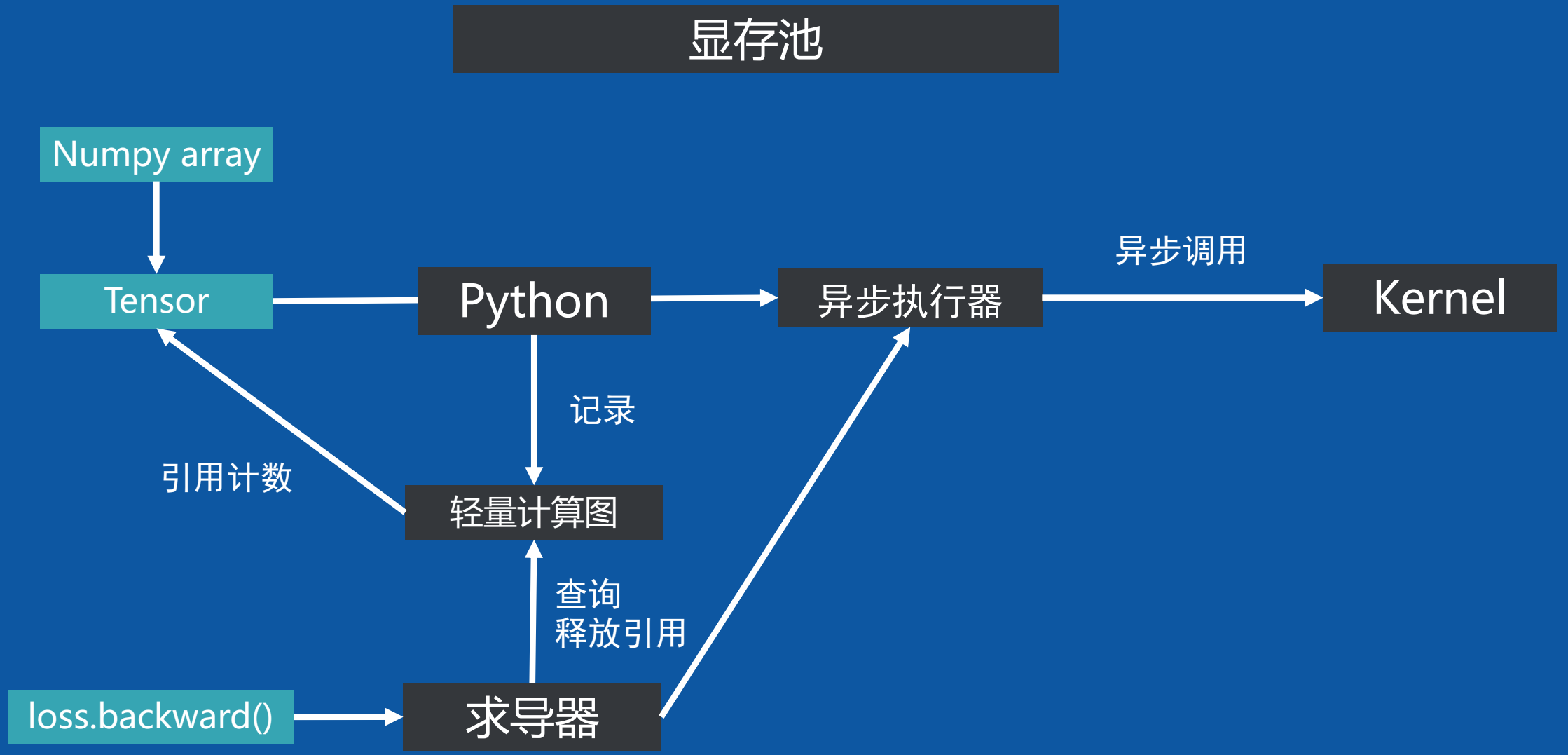


动态图训练 – 提升性能



动态图训练 – 提升性能

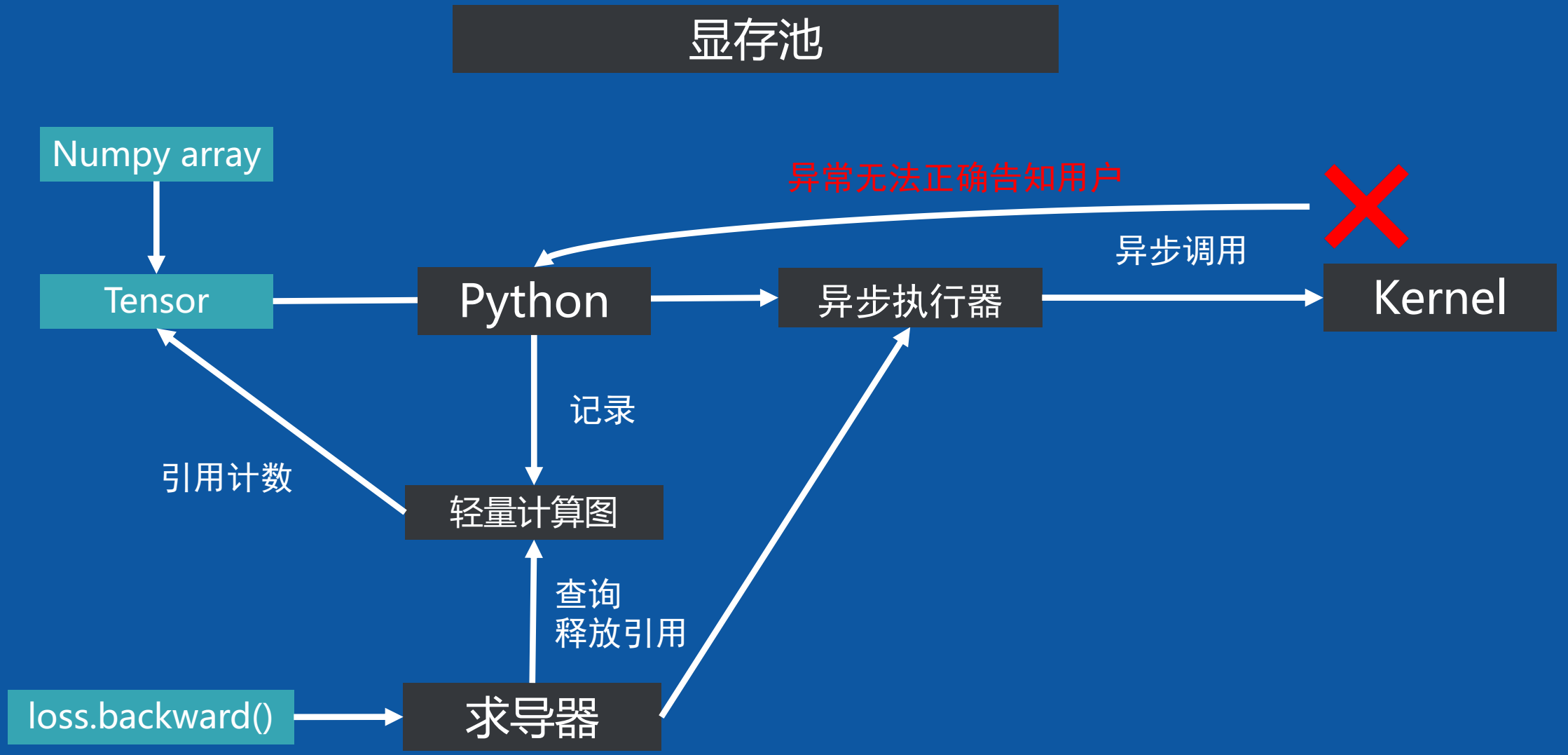
MEGVII 旷视



天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

动态图训练 – 异常



动态图训练 – 异常

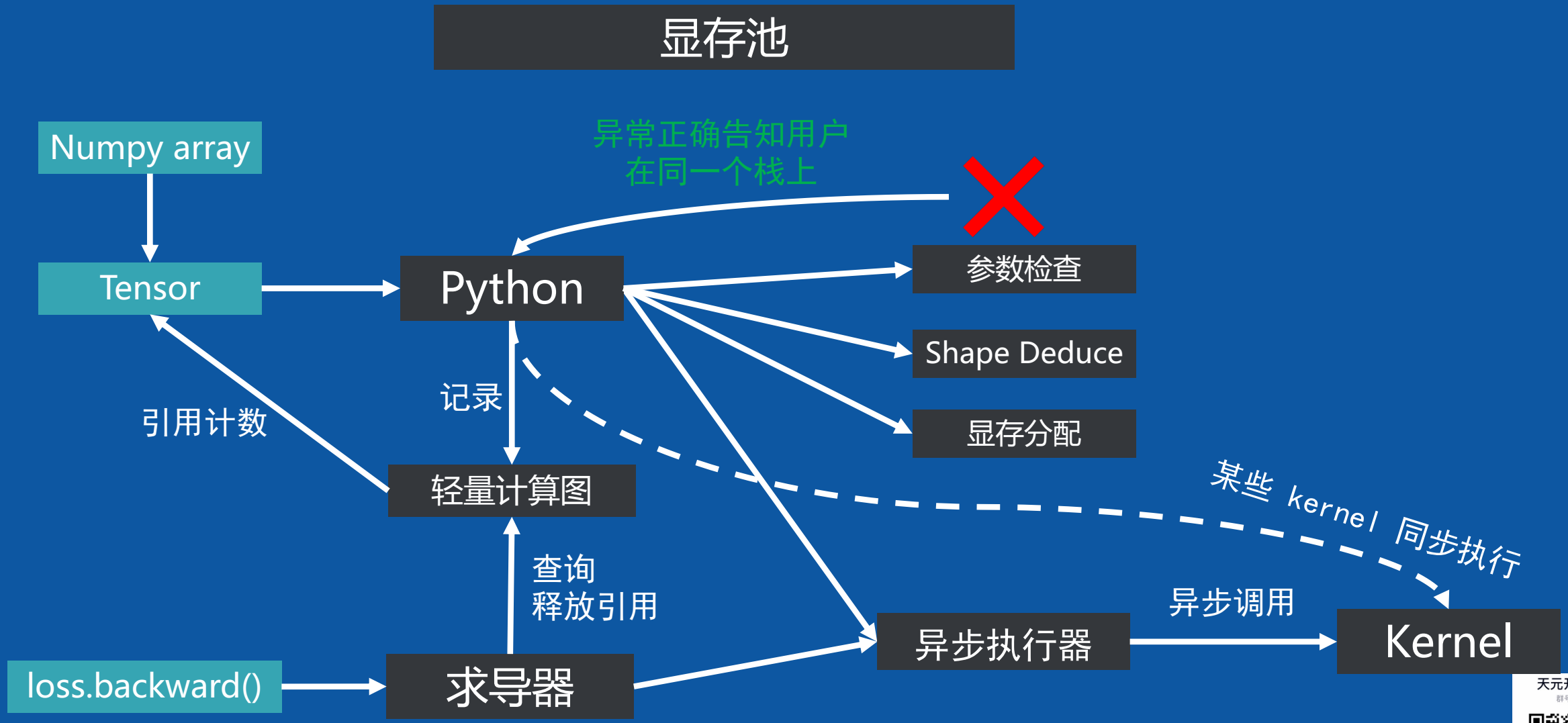
MEGVII 旷视

```
void conv_forward      (Tensor filter, Tensor src,   Tensor* dst, void* workspace, params...)
```

```
TensorLayout deduce_shape(TensorLayout filter, TensorLayout src, params)
void conv_forward      (Tensor filter, Tensor src,   Tensor* dst, void* workspace, params...)
```

动态图训练 – 异常

MEGVII 旷视



天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

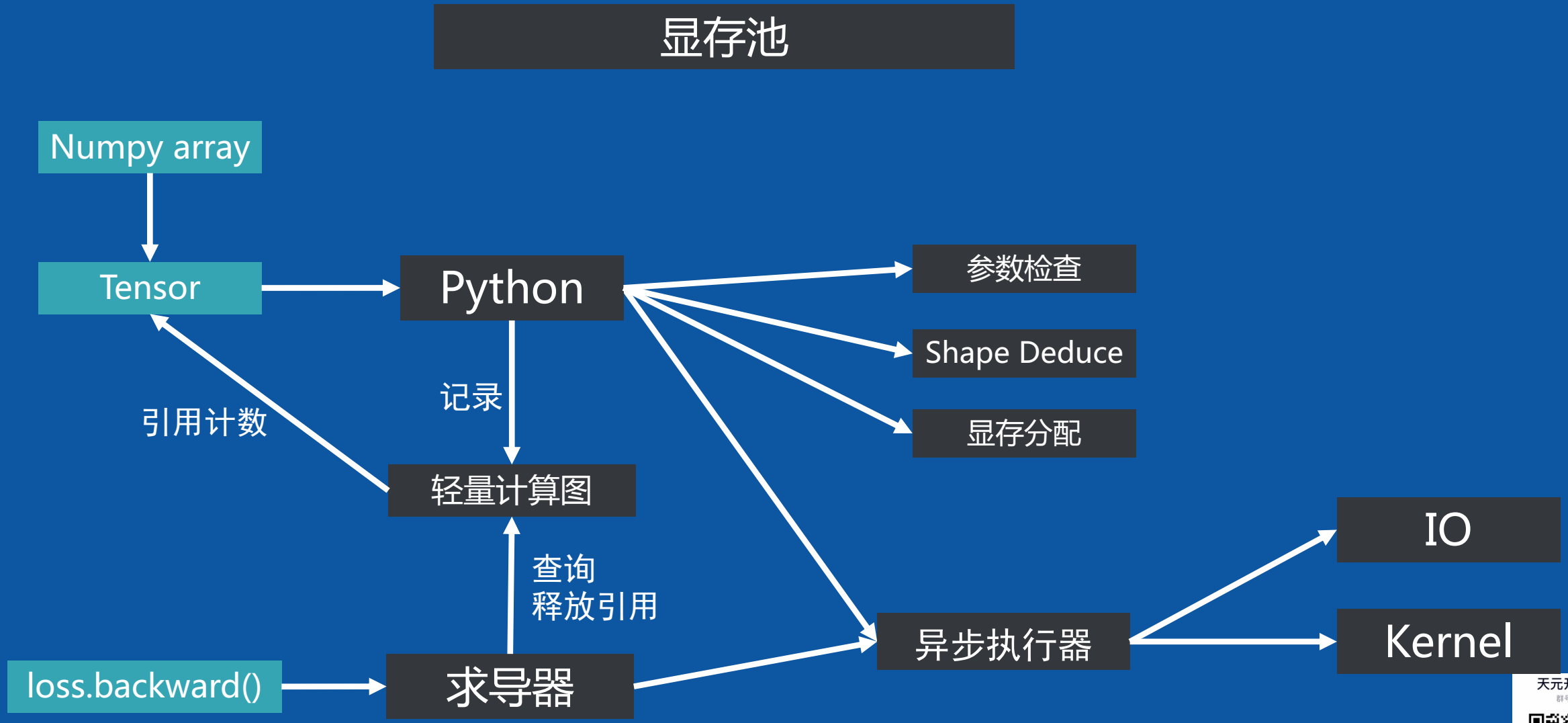
动态图训练 – 多机多卡

MEGVII 旷视

- CompNode：计算设备（例：一个线程、CUDA Stream）
- MemNode：存储单元（例：一块 GPU 上的显存、内存）
- CrossCNCopy

动态图训练 – 多机多卡

MEGVII 旷视



天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

静态图训练 + 推理（粗糙版）

MEGVII 旷视

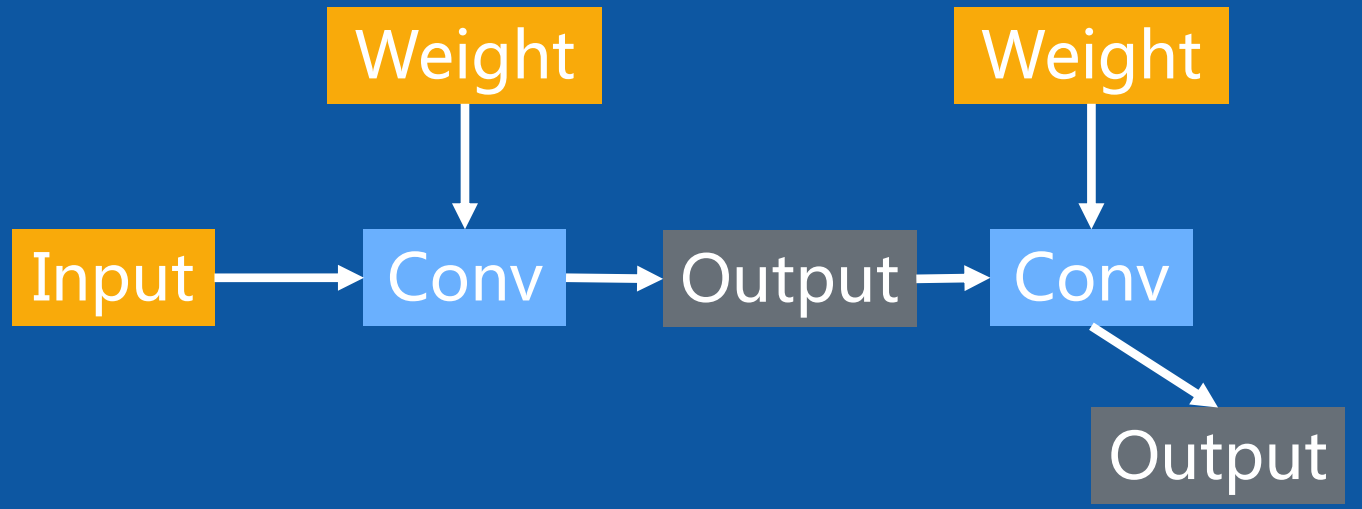
Graph

Op

VarNode

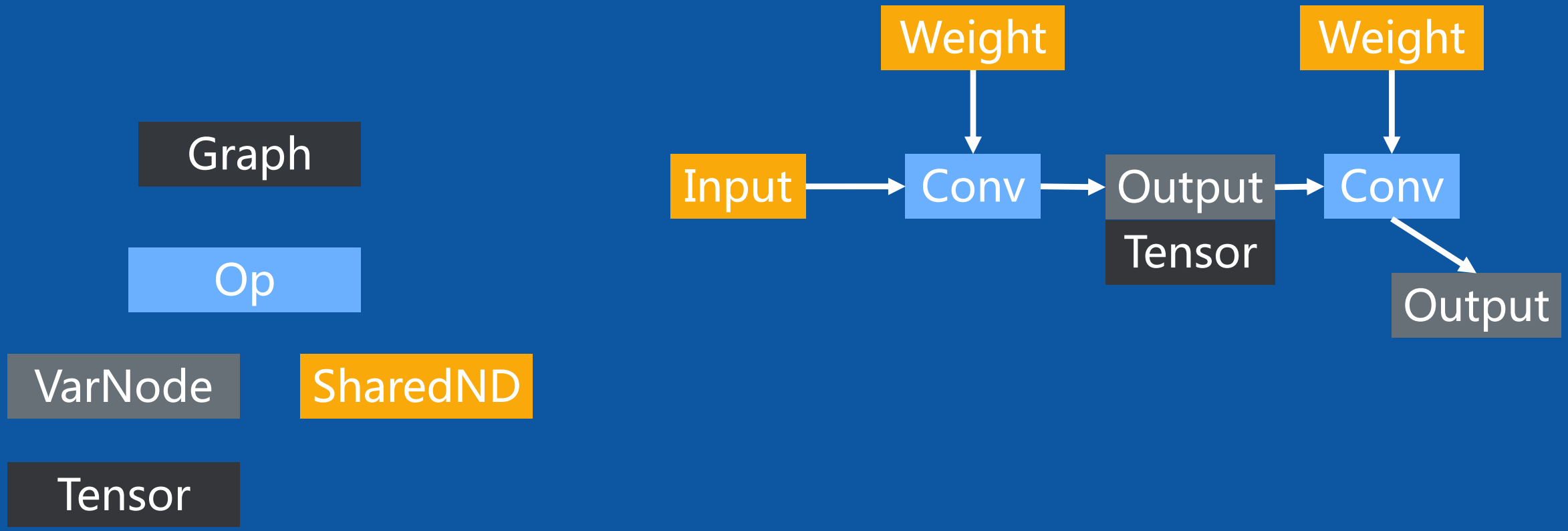
SharedND

Tensor



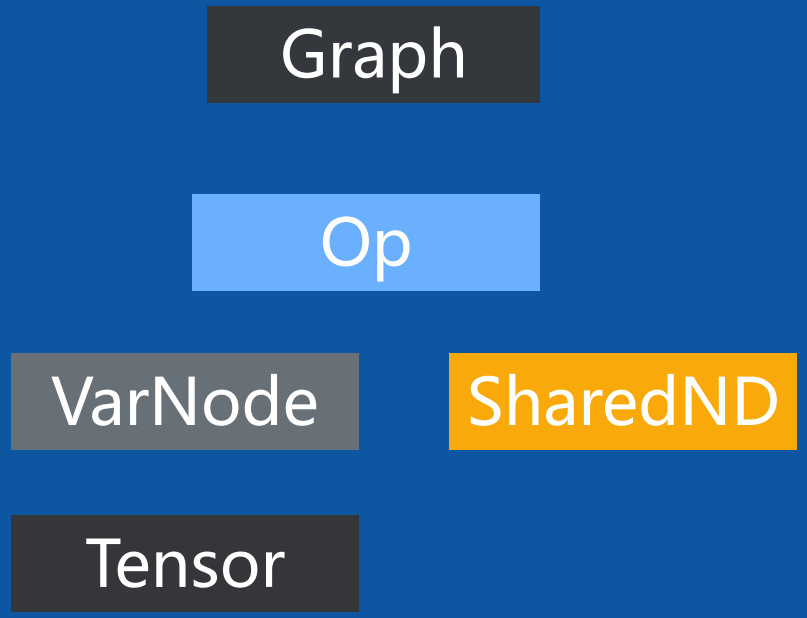
静态图训练 + 推理 - 运行时 Tensor 存储分配

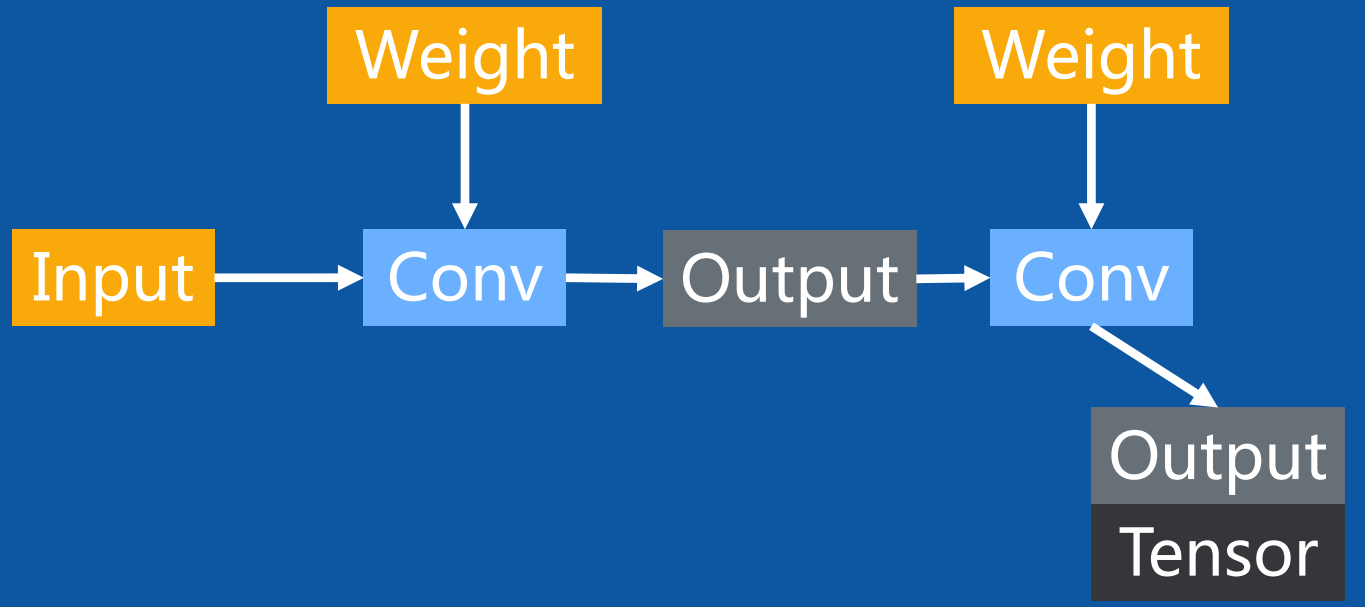
MEGVII 旷视



静态图训练 + 推理 – 运行时 Tensor 存储分配

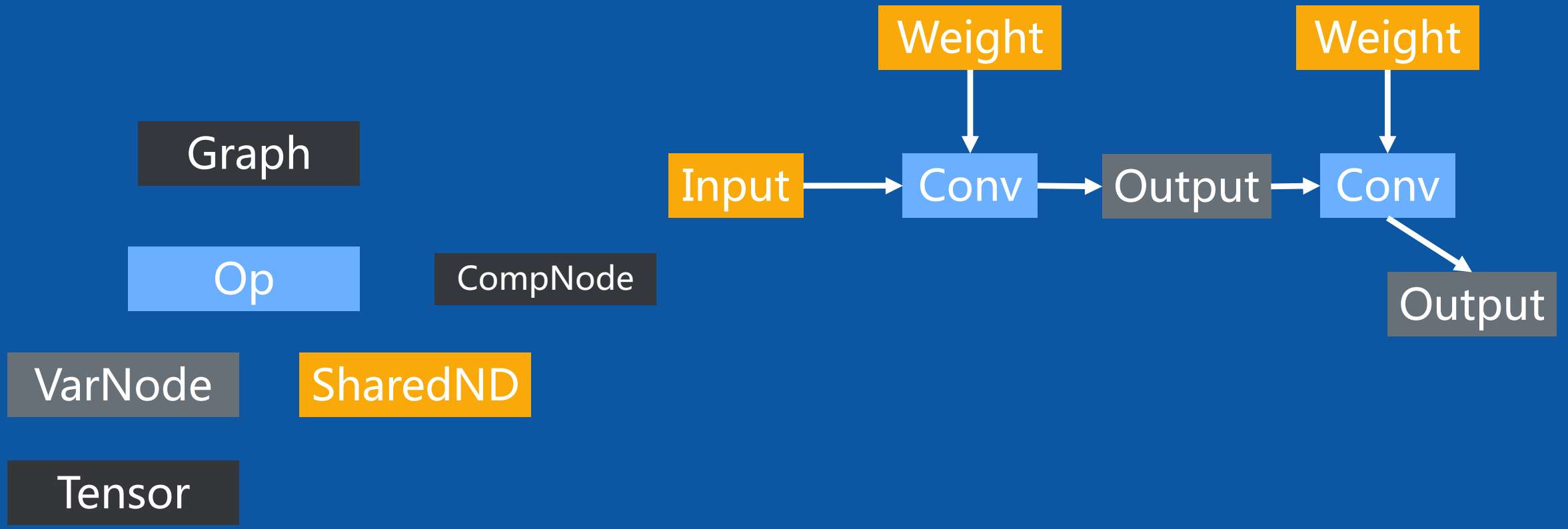
MEGVII 旷视





静态图训练 + 推理 - Cross CompNode

MEGVII 旷视



静态图训练 + 推理

MEGVII 旷视

推理
静态训练

计算图

图优化

JIT 引擎

生成计算序列
显存复用
生成显存方案

Remote IO

Kernel 选择

Kernel
执行引擎

Kernel

天元官网：https://megengine.org.cn/

GitHub（欢迎star）：https://github.com/MegEngine

天元开发者交流群
群号：1029741705
扫一扫二维码，加入群聊。

MEGVII 旷视

1 背景介绍

2 如何写出一个深度学框架？

3 一个陈年静态图框架是怎么变成动态图框架的？

4 对未来的展望

5 相关资源

静态 -> 动态？

MEGVII 旷视

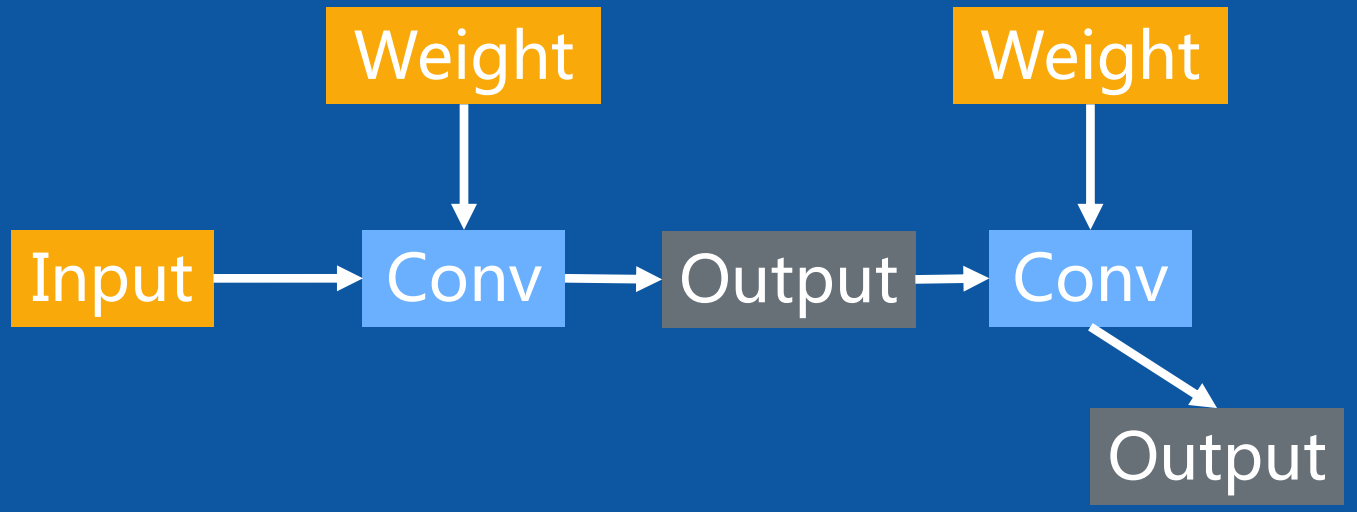
- 已有
  - 5年+静态图引擎
  - 无数采坑经验
- 目标
  - 快速搞出一套动态引擎
  - 推理部分要兼容原有格式和代码
  - 训练和推理精度可以对齐
  - 训练 -> 推理转换代价要低

Eager Graph

MEGVII 旷视

- 永不结束构造的 Graph
- 放弃优化策略，插入即执行

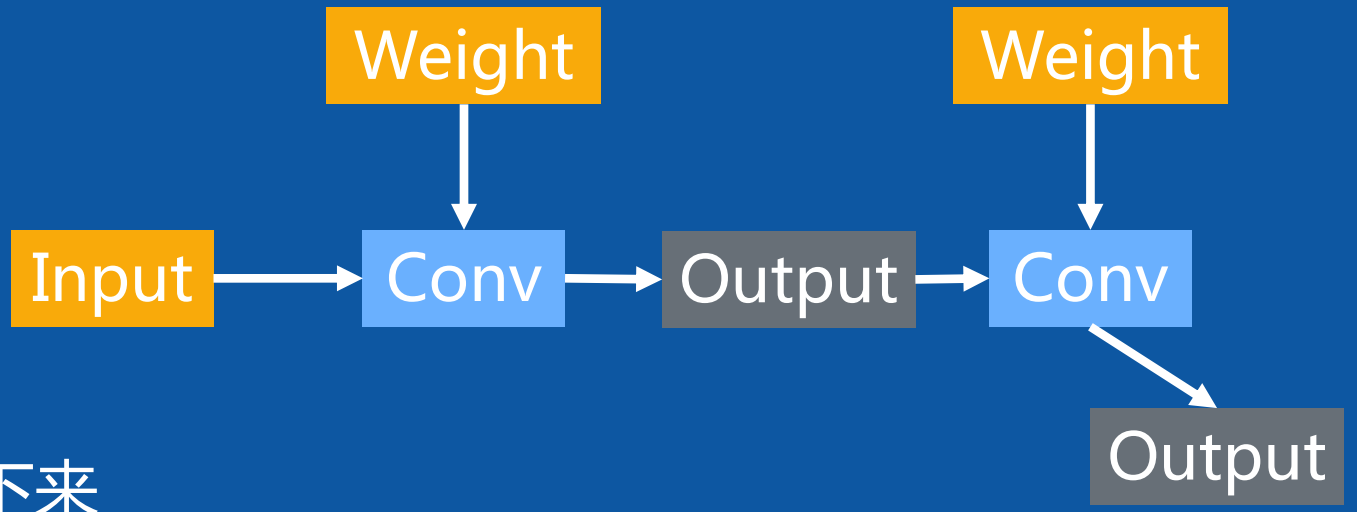
- 动态 -> 静态



Eager Graph

MEGVII 旷视

- 永不结束构造的 Graph
- 放弃优化策略，插入即执行

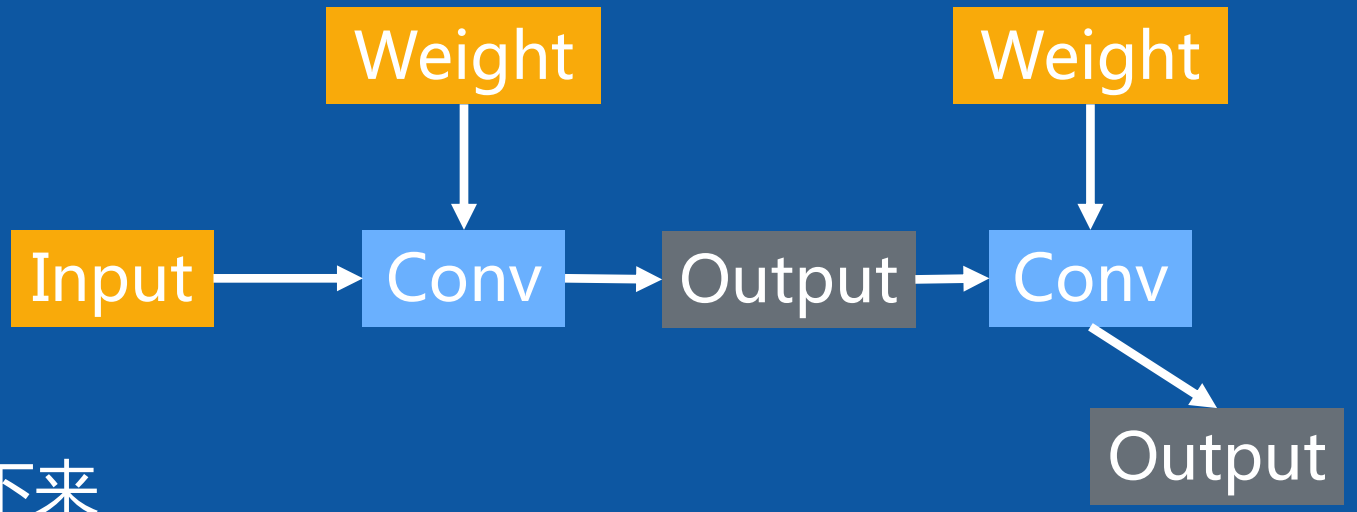


- 动态 -> 静态：把整张计算图存下来
- 复用原有求导机制

Eager Graph

MEGVII 旷视

- 永不结束构造的 Graph
- 放弃优化策略，插入即执行



- 动态 -> 静态：把整张计算图存下来
- 复用原有求导机制

- 计算图过重，基于 immutable 的设计无法删除，计算图爆炸
- 去重机制

Eager Runtime + Proxy Graph

MEGVII 旷视

User Level API

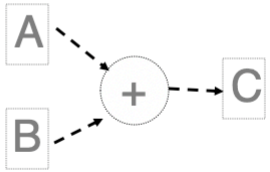
Apply OpDef On, e.g. C = apply(Elemwise('add'), A, B)

immediately calculate:
Tensor(C) = Tensor(a) + Tensor(B)

Proxy calculation task to
ComputingGraph
Binding Tensor to VarNode

Fallback Implementation
Only apply_on_var_node
needed

ProxyGraph



versus ComputingGraphImpl:
No resource management or graph runtime
Just be used as a graph container and op
methods table

Imperative
Runtime

Abstraction of Computing
Resource (CompNode)

Unified Kernel Interface
on different platforms
(MegDNN)

Utils(e.g. RTTI
system, Hash System,
Async worker, Memory
Pool)

一些趣事

MEGVII 旷视

多机 Optimizer 的默认行为应该是 sum 还是 avg

超距作用与对点

一个 Dev 冲着一个 R 含：你这里明明用 WrapAffine 就可以了，为什么你要用 Wrap Perspective！？？

R 反过来喊：都可以用 Wrap Perspective 了，为什么要用 WrapAffine！？

https://xkcd.com/1077/

MEGVII 旷视

1 背景介绍

2 如何写出一个深度学框架？

3 一个陈年静态图框架是怎么变成动态图框架的？

4 对未来的展望

5 相关资源

动态性能提升

MEGVII 旷视

- 性能
- 自动显存复用
- 修 BUG！！！
- 模型复现

各种芯片模组的对接，挑战训推一体的理念

MEGVII 旷视

- **硬件芯片自带推理框架将不可避免**
  - 对接更多芯片，成为新时代的 C 语言

- **训推一体将向更为广义的概念发展**
  - 核心：降低整个算法落地流程的代价
  - Software 2.0 & 深度学习的抽象泄露
  - 懂推理的训练框架，从从一开始就考虑推理
    - 含设备细节的 QAT 训练，点更高、速度更快
    - 坚持落地流程的简单化

MLIR 等技术的兴起

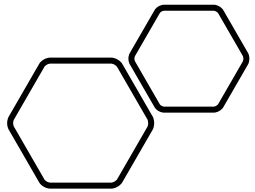
MEGVII 旷视

- 更多的编译与代码生成
- 更多的借鉴编译器的内容

如何做到真 JIT

MEGVII 旷视

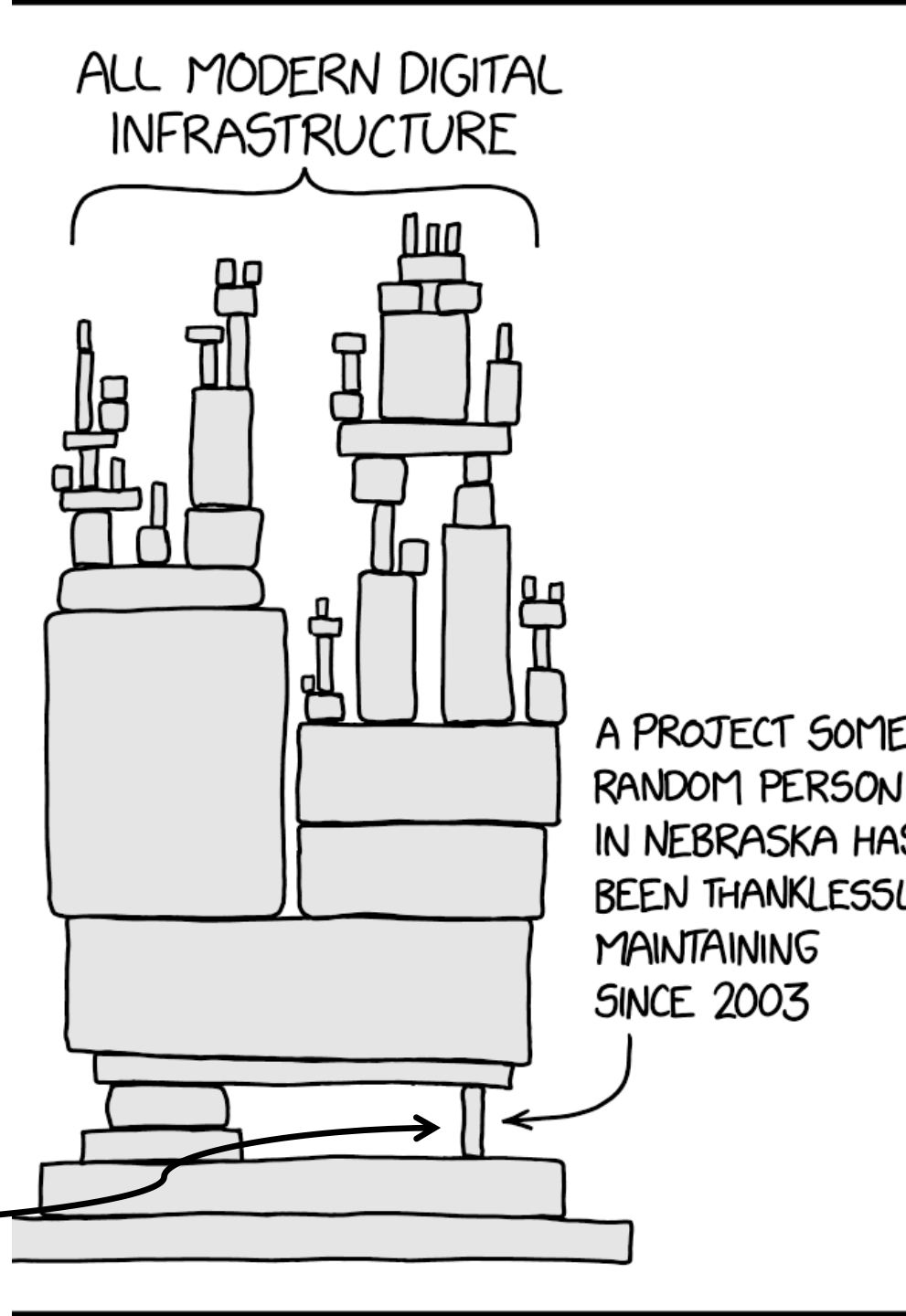
- Lazy 与预测执行



如果你觉得分享好
帮我们点个 Star

让我们能更好的做开源

github.com/MegEngine/MegEngine



MEGVII 旷视

1 背景介绍

2 如何写出一个深度学框架？

3 一个陈年静态图框架是怎么变成动态图框架的？

4 对未来的展望

5 相关资源

相关资源

MEGVII 旷视

北大公开:课深度学习实践

本课程为旷视研究院联合北大数科院机器学习实验室面向北京大学在校学生开设的深度学习基础课程，由旷视研究院院长孙剑及资深研究员们授课，适合深度学习技术的初学者

- **“旷视研究院”公众号：**
  - 工程之道，MegEngine 推理性能极致优化之综述篇
  - 深度解析 MegEngine 亚线显存性优化
  - …

天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

如何对社区进行贡献

MEGVII 旷视

- 关注 MegEngine repo，点 star
- 贡献代码
- 提高文档质量
- 回答技术问题
- 为 Model Hub 贡献模型
- 使用 MegStudio 尝试新想法
- 发现 Bug 和 Issue
- 引用 MegEngine
- 推荐 MegEngine

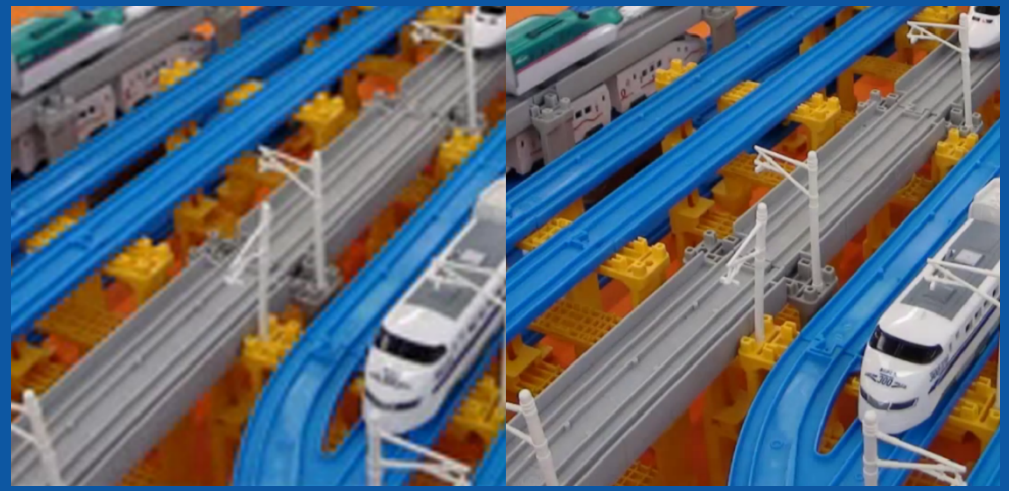
天元官网：https://megengine.org.cn/　　GitHub（欢迎star）：https://github.com/MegEngine

旷视AI开源大赛

MEGVII 旷视

比赛任务：视频超分辨率

参赛团队需要通过训练深度学习模型，针对给定的被降分辨率并压缩后的视频，尽可能保真的恢复压缩前的视频，将降级的低质量视频复原成高质量版本



大赛入口：
https://studio.brainpp.com/competition

大赛奖项设置：

- 第一名：团队奖金人民币**5万**
- 第二名：团队奖金人民币**2万**
- 第三名：团队奖金人民币**1万**
- 第四到十名：团队奖金人民币**1千**
- 参与奖：所有参加初赛，并且提交有效结果的团队，纪念T-Shirt衫

（所有奖金额为税前奖金额）

赛事安排：

- **报名阶段：** 2020年8月10日（UTC+8）到2020年8月31日23:59（UTC+8）
- **初赛阶段：** 2020年9月1日00：01(UTC+8)到2020年9月14日23：59(UTC+8)
- **决赛阶段：** 2020年9月16日00：01(UTC+8)到2020年9月24日23：59(UTC+8)

大赛前三名团队的成员还可获得旷视校招面试直通卡，直接进入面试环节

天元开发者交流群
群号：1029741705
扫一扫二维码，加入群聊。

行正则致远

AI向善，行胜于言。

MEGVII 旷视