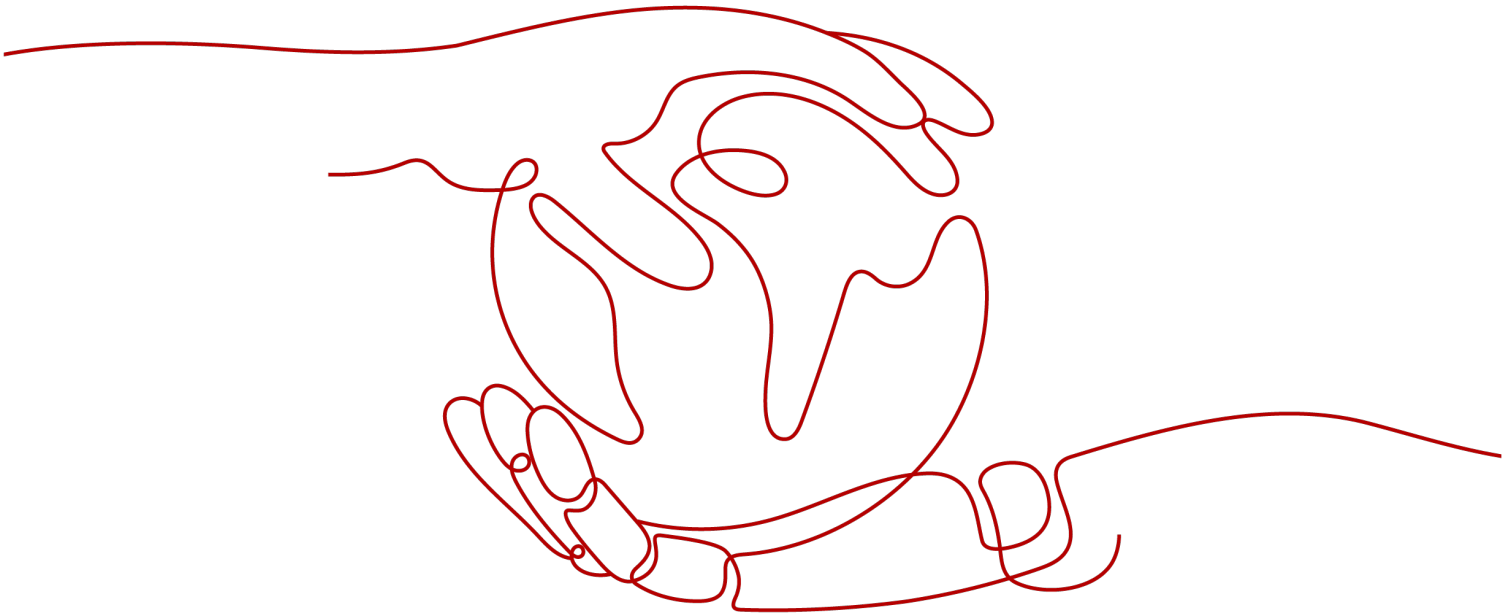
MindX SDK
3.0.RC3

mxTuningKit 微调工具包 用户指南

文档版本	01
发布日期	2022-11-30



华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 简介

微调工具包功能介绍

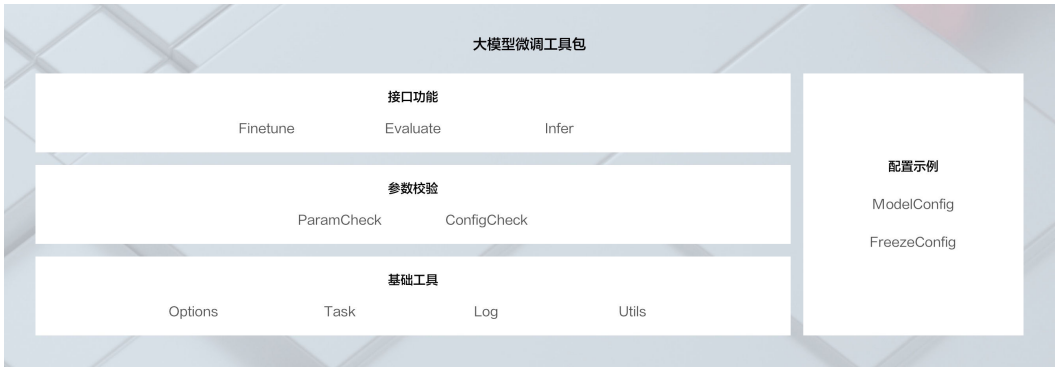
微调工具包作为大模型套件的基础工具，可以帮助用户在本地快速启动微调、评估、推理等任务。通过封装用户的大模型下游任务启动脚本，工具包提供命令行（CLI）与Python SDK两种方式简单快捷的启动任务。

软件架构

如图1-1所示，微调工具包主要由四个模块组成：

- 接口功能模块：实现微调、评估、推理接口，支持命令行（CLI）与Python SDK两种调用方式。
- 参数校验模块：实现接口参数校验、配置参数校验。
- 基础工具模块：实现参数定义、任务流程、日志、工具类。
- 配置示例模块：提供自定义模型参数、冻结网络的配置样例说明。

图 1-1 软件架构



2 软件安装

环境依赖

支持使用Python 3.7至3.9版本。

软件安装

下载微调工具包的安装包（解压获取wheel包），用户在同路径下创建安装/卸载脚本“tk_user.sh”（内容可参考**B 用户安装/卸载脚本**），运行如下命令进行安装：

```
bash tk_user.sh install
```

安装完成后，对应安装日志存放在指定日志路径“$HOME/.cache/Huawei/mxTuningKit/log/”。

软件卸载

用户可创建且调用安装/卸载脚本“tk_user.sh”（内容参考**B 用户安装/卸载脚本**），运行如下命令进行卸载：

```
bash tk_user.sh uninstall
```

卸载完成后，对应卸载日志存放在指定日志路径“$HOME/.cache/Huawei/mxTuningKit/log/”。

安全风险提示

若用户未按照上述方式进行安装与卸载操作，会存在如下风险：

- 程序文件访问权限未最小化，存在文件篡改风险。
- 缺失软件安装卸载日志记录。

3 快速上手

样例说明

本章节以开源项目**紫东.太初（OPT）**适配下游任务为例，主要展示如何使用微调工具包，快速拉起模型微调、评估、推理任务。

准备环境

步骤1 预训练模型代码。

从官网下载紫东.太初（OPT）模型代码：

```
git clone https://gitee.com/mindspore/omni-perception-pretrainer.git -b adapt_tk
```

步骤2 预训练模型权重。

从官网下载紫东.太初（OPT）模型权重至“{*opt模型代码路径*}/pretrained_model/”路径，**下载链接**。

步骤3 下游任务数据集。

参考紫东.太初（OPT）模型，准备下游任务所需的训练、测试和推理数据集，放在“{*opt模型代码路径*}/dataset/”路径。

步骤4 模型配置文件（可选）。

模型配置文件主要用于**自定义超参数**以及**冻结部分网络**，如果配置该文件，对应参数值以该文件为准；否则，使用启动脚本中的默认值。具体配置规则可参考**6.3 模型配置文件适配（可选）**。

----结束

模型微调

使用微调工具包，执行如下命令进行模型微调：

```
tk finetune
        --boot_file_path {opt模型代码路径}/code/src/scripts/train_caption.py # 微调任务启动脚本
        --data_path {opt模型代码路径}/dataset/data/ # 数据集路径根目录
        --output_path {opt模型代码路径}/output/ # 指定结果输出路径，存放需要落盘的文件，如ckpt、自定义输出等
        --pretrained_model_path {opt模型代码路径}/pretrained_model/ # 可选参数，预训练模型
        --model_config_path {opt模型代码路径}/code/model_configs/model_config_finetune.yaml # 可选参数，模型配置文件
```


说明

- 落盘结果可在指定输出路径“{*opt模型代码路径*}/output/TK_UUID/”下查看。
- 任务运行日志可在“$HOME/.cache/Huawei/mxTuningKit/log”下查看。
- 文件路径均为本地绝对路径。
- 文件/文件夹读写权限最大可设置为同组可读可执行，其他用户无访问权限。

模型评估

使用微调工具包，执行如下命令进行模型评估：

```
tk evaluate
        --boot_file_path {opt模型代码路径}/code/src/scripts/test_caption.py # 评估任务启动脚本
        --data_path {opt模型代码路径}/dataset/data/ # 数据集路径根目录
        --ckpt_path {opt模型代码路径}/pretrained_model/ # 评估所需的ckpt文件路径
        --output_path {opt模型代码路径}/output/ # 指定结果输出路径，存放需要落盘的文件，如ckpt、自定义输出等
        --model_config_path {opt模型代码路径}/code/model_configs/model_config_evaluate.yaml # 可选参数，模型配置文件
```

说明

- 文件路径均为本地绝对路径。
- 落盘结果可在指定输出路径“{*opt模型代码路径*}/output/TK_UUID/”下查看。
- 任务运行日志可在“$HOME/.cache/Huawei/mxTuningKit/log”下查看。

模型推理

使用微调工具包，执行如下命令进行模型推理：

```
tk infer
      --boot_file_path {opt模型代码路径}/code/src/scripts/inference_caption.py # 推理任务启动脚本
      --data_path {opt模型代码路径}/dataset/data/ # 数据集路径根目录
      --ckpt_path {opt模型代码路径}/pretrained_model/ # 推理所需的ckpt文件路径
      --output_path {opt模型代码路径}/output/ # 指定结果输出路径，存放需要落盘的文件，如ckpt、自定义输出等
      --model_config_path {opt模型代码路径}/code/model_configs/model_config_infer.yaml # 可选参数，模型配置文件
```


4 使用场景

4.1 命令行场景

4.1.1 模型微调

功能介绍

功能名称	功能说明	备注
finetune	模型微调	根据用户启动脚本，本地创建并拉起模型训练或微调任务。

前提约束

- 运行finetune命令前，按照参数说明准备好训练所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。

命令示例

```
tk finetune --quiet --data_path /home/HwAiUser/datasets/ --output_path /home/HwAiUser/output/ --pretrained_model_path /home/HwAiUser/pretrained_models/ --model_config_path /home/HwAiUser/config/model_config_finetune.yaml --boot_file_path /home/HwAiUser/finetune.py --timeout 1d2h
```

finetune 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号**。
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**。

- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

参数缩写	参数全称	必选	路径粒度	描述
-bp	--boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/finetune.py”。
-dp	--data_path	是	文件夹	待训练数据集本地绝对路径，如“/home/HwAiUser/caption/datasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
-op	--output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请确保用户有该文件夹的写权限；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
-pm	--pretrained_model_path	否	文件夹	预训练模型本地绝对路径，如/“home/HwAiUser/caption/pretrained_models/”。模型代码必须从该参数值对应路径获取预训练模型文件，不配置该参数时默认用户不加载预训练模型。

参数缩写	参数全称	必选	路径粒度	描述
-mc	--model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/model_config_finetune.yaml”；用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。
-q	--quiet	否	不涉及	使用该参数即可开启安静模式，具体使用方式可参考**5.2 启用安静模式（--quiet）**。该参数生效方式为**设置为首个参数**。在该模式下，命令行接口不再对除“--boot_file_path”外的参数进行提权风险交互式提醒，只通过warning日志来提示风险，不影响后续业务的执行。**不建议用户开启安静模式**。
-t	--timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/”下查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成随机串，用户可在其中查看模型训练结果与自定义落盘文件。

4.1.2 模型评估

功能介绍

功能名称	功能说明	备注
evaluate	模型评估	根据用户启动脚本，本地拉起评估任务。

前提约束

- 运行evaluate命令前，按照参数说明准备好评估所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。
- 用户模型代码需将模型评估结果以json串格式落盘至指定输出路径（output_path），落盘文件名为“eval_result.json” 且用户具有访问该文件的权限。

命令示例

```
tk evaluate --quiet --data_path /home/HwAiUser/dataset/ --ckpt_path /home/HwAiUser/ckpt/ --
output_path
/home/HwAiUser/output/ --model_config_path /home/HwAiUser/config/model_config_eval.yaml --
boot_file_path
/home/HwAiUser/eval.py --timeout 1d2h
```

evaluate 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号**。
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**。
- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

参数缩写	参数全称	必选	路径粒度	描述
-bp	--boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/eval.py”。

参数缩写	参数全称	必选	路径粒度	描述
-dp	--data_path	是	文件夹	待验证数据集（包含ground truth）本地绝对路径，如“/home/HwAiUser/caption/verifydatasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
-op	--output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请**确保用户有该文件夹的写权限**；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
-cp	--ckpt_path	是	文件夹	待评估模型文件本地绝对路径，如“/home/HwAiUser/caption/ckpt/”。
-mc	--model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/model_config_evaluate.yaml”。用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。

参数缩写	参数全称	必选	路径粒度	描述
-q	--quiet	否	不涉及	使用该参数即可开启安静模式，具体使用方式可参考**5.2 启用安静模式（--quiet）**。该参数生效方式为**设置为首个参数**。在该模式下，命令行接口不再对除“--boot_file_path”外的参数进行提权风险交互式提醒，只通过warning日志来提示风险，不影响后续业务的执行。**不建议用户开启安静模式**。
-t	--timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/”下查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成随机串，用户可在其中查看模型训练结果与自定义落盘文件。

4.1.3 模型推理

功能介绍

功能名称	功能说明	备注
infer	模型推理	根据用户启动脚本，本地拉起推理任务。

前提约束

- 运行infer命令前，按照参数说明准备好训练所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。
- 用户模型代码需将模型推理结果以json串格式落盘至指定输出路径（output_path），落盘文件名为“infer_result.json”且用户具有访问该文件的权限。

命令示例

```
tk infer --quiet --data_path /home/HwAiUser/dataset/ --ckpt_path /home/HwAiUser/ckpt/ --output_path
/home/HwAiUser/output/ --model_config_path /home/HwAiUser/config/model_config_infer.yaml --
boot_file_path
/home/HwAiUser/infer.py --timeout 1d2h
```

infer 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号。**
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**。
- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

参数缩写	参数全称	必选	路径粒度	描述
-bp	--boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/infer.py”。
-dp	--data_path	是	文件夹	待推理数据集本地绝对路径，如“/home/HwAiUser/caption/inferdatasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
-op	--output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请**确保用户有该文件夹的写权限**；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
-cp	--ckpt_path	是	文件夹	推理模型文件本地绝对路径，如“/home/HwAiUser/caption/ckpt/”。

参数缩写	参数全称	必选	路径粒度	描述
-mc	--model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/model_config_infer.yaml”。用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。
-q	--quiet	否	不涉及	使用该参数即可开启安静模式，具体使用方式可参考**5.2 启用安静模式（--quiet）**。该参数生效方式为**设置为首个参数**。在该模式下，命令行接口不再对除“--boot_file_path”外的参数进行提权风险交互式提醒，只通过warning日志来提示风险，不影响后续业务的执行。**不建议用户开启安静模式**。
-t	--timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/”下查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成随机串，用户可在其中查看模型训练结果与自定义落盘文件。

4.2 SDK 场景

4.2.1 模型微调

```
import tk.tk_sdk as tk # 导入微调工具包
tk.finetune(data_path, output_path, boot_file_path, pretrained_model_path=None,
model_config_path=None,timeout=None)
```


功能介绍

根据给定参数值创建并拉起模型训练或微调任务。

前提约束

- 运行finetune命令前，按照参数说明准备好训练所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。

tk_sdk.finetune 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号**；
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**；
- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置为**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

表 4-1 入参说明

参数全称	必选	路径粒度	描述
boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/finetune.py”。
data_path	是	文件夹	待训练数据集本地绝对路径，如“/home/HwAiUser/caption/datasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请**确保用户有该文件夹的写权限**；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
pretrained_model_path	否	文件夹	预训练模型本地绝对路径，如“/home/HwAiUser/caption/pretrained_models/”。模型代码必须从该参数值对应路径获取预训练模型文件，不配置该参数时默认用户无预训练需求。

参数全称	必选	路径粒度	描述
model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/model_config_finetune.yaml”；用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。
timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

表 4-2 返回值说明

返回值	说明
True	任务成功。
False	任务失败。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/下”查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成，用户可在其中查看模型训练结果与自定义落盘文件。

4.2.2 模型评估

```
import tk.tk_sdk as tk # 导入微调工具包
tk.evaluate(data_path, ckpt_path, output_path, boot_file_path, model_config_path=None, timeout=None)
```

功能介绍

根据给定参数值创建并拉起评估任务，任务结束后读取用户模型评估结果文件并返回。

前提约束

- 运行evaluate命令前，按照参数说明准备好训练所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。
- 用户模型代码需将模型评估结果以json串格式落盘至指定输出路径（output_path），落盘文件名为“eval_result.json”，且用户具有访问该文件的权限。

tk_sdk.evaluate 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号**。
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**。
- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置为**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

表 4-3 入参说明

参数全称	必选	路径粒度	描述
boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/eval.py”。
data_path	是	文件夹	待验证数据集（包含ground truth）本地绝对路径，如“/home/HwAiUser/caption/verifydatasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请**确保用户有该文件夹的写权限**；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
ckpt_path	是	文件夹	待评估模型文件本地绝对路径，如“/home/HwAiUser/caption/ckpt/”。
model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/eval_config_caption.yaml”。用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。
timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

表 4-4 返回值说明

返回值	说明
eval_result	任务成功，返回json串形式的评估结果。
空字符串	任务失败。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/”下查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成，用户可在其中查看模型训练结果与自定义落盘文件。

4.2.3 模型推理

```
import tk.tk_sdk as tk # 导入微调工具包
tk.infer(data_path, ckpt_path, output_path, boot_file_path, model_config_path=None, timeout=None)
```

功能介绍

根据给定参数值创建并拉起推理任务，任务结束后读取用户模型推理结果文件并返回。

前提约束

- 运行infer命令前，按照参数说明准备好训练所需文件，做好对应权限控制。
- 模型代码必须与微调工具包适配，具体适配方式参考**6 模型适配**。
- 用户模型代码需将模型推理结果以json串格式落盘至指定输出路径（output_path），落盘文件名为“infer_result.json”且用户具有访问该文件的权限。

tk_sdk.infer 参数说明

参数规则：

- 文件命名规则为：允许**字母大小写、数字、减号、下划线、空格、英文句号**。
- 文件夹命名规则为：允许**字母大小写、数字、减号、下划线、空格**。
- 在拉起服务前对输入输出的文件与路径做合理权限控制；为避免越权攻击，建议读写权限设置为**同组用户不具有写权限，其他用户不具有任何访问权限，参考A 文件/文件夹权限规则说明**。
- 超时中断机制参数设置的天数需设置为自然数，小时数需设置为小于等于23的自然数。

表 4-5 入参说明

参数全称	必选	路径粒度	描述
boot_file_path	是	文件	模型启动脚本本地绝对路径，需要为**.py文件**，如“/home/HwAiUser/caption/infer.py”。
data_path	是	文件夹	待推理数据集本地绝对路径，如“/home/HwAiUser/caption/infer-tasets/”。仅当数据通过sfs存储时允许模型代码中指定数据读取路径，其他场景下，模型代码必须从该参数值对应路径下获取数据集。
output_path	是	文件夹	输出文件本地绝对路径，如“/home/HwAiUser/caption/output/”。模型代码必须将需要落盘的文件写入该路径。请**确保用户有该文件夹的写权限**；保证该路径有足够存储空间（大于1GB），否则会进行交互式风险提醒，由用户选择是否接受风险继续业务；且需保证输出路径不与其他输入路径相同或为其他输入路径的子路径。
ckpt_path	是	文件夹	推理模型文件本地绝对路径，如“/home/HwAiUser/caption/ckpt/”。
model_config_path	否	文件	“model_config”配置文件本地绝对路径，如“/home/HwAiUser/caption/config/infer_config_caption.yaml”。用户可参考**6 模型适配**适配该文件。通过该文件修改模型代码所需超参，也可以不配置该参数，直接使用源码预定义超参。
timeout	否	不涉及	使用该参数即可开启任务超时中断机制，启动的任务运行时间如超过设定的超时时间，任务将被强制停止。超时时间允许天数（d/D）/小时数（h/H）的组合，例如设置3d12h、5d、18h都是合法的。

表 4-6 返回值说明

返回值	说明
infer_result	任务成功，返回json串形式的推理结果。
空字符串	任务失败。

日志文件与结果输出

日志文件可在“$HOME/.cache/Huawei/mxTuningKit/log/”下查看，具体存放规则参考**D 日志存放规则**。

对于落盘结果文件，微调工具包将会在指定输出路径新建一个读写权限为“750”，名称为“TK_UUID”文件夹，其中*UUID*为动态生成，用户可在其中查看模型训练结果与自定义落盘文件。

5 接口风险警示

5.1 风险交互提示（默认模式）

默认模式下，用户在命令行发起模型微调/评估命令时，如果输入参数路径指向文件/文件夹访问权限设置过于宽松，不合理（权限设置规则可参考**A 文件/文件夹权限规则说明**）时，微调工具包会给出如下风险提示，需要用户确认：

接口	参数项	风险操作	CLI	Python接口
finetune	pretrained_model_path	输入参数路径指向文件/文件夹访问权限设置过于宽松，有提权风险。	交互式风险提醒，由用户选择是否接受风险继续业务。	warning日志提醒。
evaluate/infer	ckpt_path			
finetune/evaluate/infer	data_path			
	output_path			
	model_config_path			
	boot_file_path			返回报错。

即CLI接口会给予用户风险交互提示。Python接口除高风险参数“boot_file_path”权限提升校验不通过时直接报错停止任务外，其余参数会在日志中给出风险提示，但不终止任务。

5.2 启用安静模式（--quiet）

用户调用CLI接口时，如无需接收风险提示，可在命令首个参数中添加“--quiet”参数，以启用安静模式。开启该模式后，默认接受除“boot_file_path”外其他潜在的风险，即可跳过CLI的交互式提示。

命令示例

```
tk finetune --quiet --data_path /home/HwAiUser/datasets/ --output_path /home/HwAiUser/output/ --pretrained_model_path /home/HwAiUser/pretrained_models/ --model_config_path /home/HwAiUser/config/model_config_finetune.yaml --boot_file_path /home/HwAiUser/finetune.py
```

说明

“--quiet”参数必须放在首位。

启用安静模式之后，存在权限风险时，提示机制变为：

接口	参数项	风险操作	CLI	CLI安静模式	Python接口
finetune	pretrained_model_path	输入参数路径指向文件/文件夹访问权限设置过于宽松，有提权风险。	交互式风险提醒，由用户选择是否接受风险继续业务。	**warning日志提醒。**	warning日志提醒。
evaluate/infer	ckpt_path				
finetune/evaluate/infer	data_path				
	output_path				
	model_config_path				
	boot_file_path			**返回报错**。	返回报错。

6 模型适配

6.1 简介

本章将详细介绍将模型适配微调工具包需要的步骤及对应要求，该过程是使用微调工具包进行微调/评估/推理任务的**必要前提**。

6.2 模型启动脚本适配（必选）

模型适配过程，主要针对模型启动脚本进行适配。根据任务类型（微调/评估/推理）的不同，在模型启动脚本中定义所需参数。

由于模型启动脚本要求为python文件，推荐使用argparse，click等模块实现。

模型微调

微调启动脚本	参数名	说明
必选适配参数	data_path	微调所需数据集根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确加载数据，导致任务失败。
	output_path	模型微调结果输出路径根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确获得输出路径，导致输出文件保存失败。

微调启动脚本	参数名	说明
可选适配参数	pretrained_model_path	预训练模型文件根目录。当用户需要在微调启动前加载预训练模型时，需要定义并使用该参数，并在模型代码内部实现具体预训练模型加载逻辑（具体模型文件可通过模型自定义超参数配置，详见“模型自定义超参数”）。如用户无加载预训练模型诉求时，可不定义该参数。

如果模型代码中存在相对路径参数，请将**tk**运行目录切换到模型代码的根目录，模型微调启动脚本适配样例可参考**E 模型适配样例**。

模型评估

评估启动脚本	参数名	备注
必选适配参数	data_path	评估所需数据集根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确加载数据，导致任务失败。
	output_path	模型评估结果输出路径根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确获得输出路径，导致输出文件保存失败。此外，用户需要将评估结果**以json串的形式存储在该路径下**，文件命名为“eval_result.json”，否则可能因无法获取评估结果判定评估任务失败。
	ckpt_path	待评估模型文件根目录。用户必须在模型启动脚本中定义并使用该参数，并在模型代码内部实现具体模型加载逻辑（具体模型文件名可通过配置文件中的params部分配置，即下述“模型自定义超参数”），否则可能因无法正确获取模型文件，导致评估结果异常。

如果模型代码中存在相对路径参数，请将**tk**运行目录切换到模型代码的根目录，模型评估启动脚本适配样例可参考**E 模型适配样例**。

模型推理

推理启动脚本	参数名	备注
必选适配参数	data_path	推理所需数据集根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确加载数据，导致任务失败。

推理启动脚本	参数名	备注
	output_path	模型评估结果输出路径根目录。用户必须在模型启动脚本中定义并使用该参数，否则可能因无法正确获得输出路径，导致输出文件保存失败。此外，用户需要将推理结果以json串的形式存储在该路径下，文件命名为“infer_result.json”，否则可能因无法获取评估结果判定推理任务失败。
	ckpt_path	推理所需模型文件根目录。用户必须在模型启动脚本中定义并使用该参数，并在模型代码内部实现具体模型加载逻辑（具体模型文件名可通过配置文件中的params部分配置，即下述“模型自定义超参数”），否则可能因无法正确加载模型文件，导致推理结果异常。

如果模型代码中存在相对路径参数，请将**tk**运行目录切换到模型代码的根目录，模型评估启动脚本适配样例可参考**E 模型适配样例**。

6.3 模型配置文件适配（可选）

文件作用

模型配置文件为用户提供配置自定义超参数与网络冻结的功能。

如果用户不配置该文件，则模型超参数使用模型启动脚本定义的默认值，另外可根据需求自行实现网络冻结功能。

如果用户想自定义模型超参数，或训练时需要冻结网络部分参数，可以配置该文件，并且在调用工具包功能接口（微调/评估/推理）时，传入参数“model_config_path”，其参数值为模型配置文件的本地绝对路径。

配置规则

模型配置文件必须是yaml 格式，内部包括**params**与**freeze**两部分，配置规则如下：

- params

  将模型超参数，按照yaml 缩进配置在params 模块下，仅支持单层级键值对形式。微调工具包会将这些配置参数，传递给模型启动脚本中预定义的同名超参数。

  说明

  此处配置的键值不允许与微调工具包预定义的参数名称或缩写名称重名，包括： dp，data_path，op，output_path，bp，boot_file_path，mc，model_config_path，pm，pretrained_model_path，cp，ckpt_path，q，quiet，t，timeout，advanced_config。键值对应的value不支持配置列表、元组、字典、集合等python对应的数据结构。

- freeze
  - 将需要冻结的网络层名称，按照yaml缩进配置在freeze下。当微调工具包检测到yaml文件中freeze部分存在配置时，会自动生成超参数“--

advanced_config”并传递到模型启动脚本，其参数值为模型配置文件的本地绝对路径。

- 模型代码内部需要适配网络冻结代码，实现网络冻结，可参考**C 网络冻结样例代码**。

模型配置文件样例参见如下，模型启动脚本配置示例可参考**E 模型适配样例**。

```
params:
  sample_key: sample_value
freeze:
    net1:
      block1:
        layer1
        layer2
        ...
      block2:
        ...
    net2:
      block1:
        layer1
        ...
```


7 分布式训练场景

7.1 单机多卡训练场景

本章节以**OPT**中caption任务的finetune功能为例，配置请参考**MindSpore官方教程 单机多卡训练章节**。

修改模型启动脚本，将Python启动方式替换为微调工具包启动方式，修改后完整脚本参考**链接**。

- 原启动命令。

```
python -u src/scripts/train_caption.py \
        --config=config/caption/$task_config_file \
        --pretrained_model_path=$pretrained_model_path \
        --output_path=$output_path/$task_name \
        --use_parallel=True \
        --data_path=$data_path &
```

- 修改后启动命令。

```
tk finetune \
       --quiet \
       --model_config_path $model_config_path \
       --boot_file_path $boot_file_path \
       --data_path $data_path \
       --output_path $output_path \
       --pretrained_model_path $pretrained_model_path &
```

  - “boot_file_path”为train脚本的**绝对路径**。
  - “data_path”为数据集**绝对路径**。
  - “model_config_path”为模型配置文件**绝对路径**，配置文件请参见**7.3 模型配置文件**。
  - “output_path”为输出**绝对路径**。

7.2 多机多卡训练场景

本章节以**OPT**中caption任务的finetune功能为例，配置请参考**MindSpore官方教程 多机多卡训练章节**。

模型启动脚本的“SERVER_ID”需要根据机器进行修改，其他修改与**7.1 单机多卡训练场景**相同，修改后完整脚本参考**链接**。

```
export SERVER_ID=0 # 第一台机器设置为0，第二台机器设置为1
```

- RANK_TABLE_FILE的生成方式参考**F RANK_TABLE_FILE生成方法**。
- 每台机器分别设置SERVER_ID，如第一台机器为0，第二台机器为1。
- 为保证运行日志正确落盘，建议合理设置SERVER_ID，DEVICE_ID，取值范围[0, 8)。
- 每台机器对应的启动脚本、模型代码以及“rank_table_file.json”的存放路径需保持一致。

7.3 模型配置文件

示例对应配置文件“model_config_finetune.yaml”内容参见如下。

```
params:
  start_learning_rate: 1e-5
  end_learning_rate: 1e-7
  epochs: 10
  decay_epochs: 10
  use_parallel: True
```

8 安全风险提示

根据**CVE-2021-28861**，python3.x至python3.10版本的“lib/http/server.py”模块存在一个开放重定向问题，URI路径的开头没有做多个“/”的检查，可能会导致一些重定向攻击以及信息泄露风险。此为开源软件漏洞，工具包使用中也不涉及该场景。如需修复，用户可自行前往官网下载补丁。

9 FAQ

9.1 日志告警 risk of disk exhaustion

问题描述

日志告警： free space of disk where param [output_path] is located is no greater than 1 GB, there is risk of disk exhaustion.'

原因分析

用户指定输出路径“output_path”中剩余空间不足1GB。

解决方案

需确保输出路径所在磁盘的大小足够，避免因空间不足导致文件落盘异常。

9.2 日志告警/报错 risk of rights escalation

问题描述

日志告警/报错： current login user has accepted the risk of rights escalation.

原因分析

文件/文件夹属主权限异常。

解决方案

参考**A 文件/文件夹权限规则说明**进行处理。

9.3 日志报错 invalid character(s) in param

问题描述

日志报错： invalid character(s) in param: [***], check settings.

原因分析

文件命名异常。

解决方案

文件命名规则为：允许字母大小写、数字、减号、下划线、空格；文件夹命名规则为：允许字母大小写、数字、减号、下划线、空格、英文句号。

9.4 日志报错 maximum recursion depth exceeded while calling a Python object

问题描述

日志报错： maximum recursion depth exceeded while calling a Python object

原因分析

“model_config*.yaml”文件中freeze模块层级过深，Python最大递归深度错误。

解决方案

请用户检查模型配置文件的层级深度，建议深度不超过100。

A 文件/文件夹权限规则说明

用户入参路径中的文件/文件夹权限最大范围为“750”（rwxr-x---），微调工具包会对用户输入参数做属主一致性校验和路径权限范围校验，所有入参路径要求属主与当前用户保持一致。

- CLI使用场景，如发现违背上述规则，会在命令行给出提示，需要用户二次确认。
- SDK使用场景，参数“boot_file_path”对应文件权限范围必须控制在“750”之内，即发现违背上述规则时任务请求将被直接终止。其他参数对应的文件与文件夹权限不做强制要求，当权限范围超出“750”时，微调工具包会以警告方式提示用户该文件属主不一致，但不会终止任务。

B 用户安装/卸载脚本

提供参考安装/卸载脚本如下，用户可拷贝，本地保存为sh脚本（如tk_user.sh）：

```
#!/bin/bash

script=$(readlink -f "$0")
route=$(dirname "$script")

tk_package_name="Ascend_mindxsdk_mxTuningKit"
python_version="python3"
cmd_para="$1"

get_python_version()
{
  $python_version --version &>/dev/null
  if test $? -ne 0; then
    echo "Can not find $python_version, wheel package will not be installed."
    exit
  fi
}

tk_get_install_path()
{
  space_char=' '

  # get tk install path
  tk_install_info=$($python_version -m pip show $tk_package_name | grep 'Location:')
  tk_install_path=$(echo $tk_install_info |sed 's/^Location: //g')

  # get tk bin file path
  tk_bin_info=$(whereis tk)
  tk_bin_path=${tk_bin_info#*$space_char}
  if [[ $tk_bin_path == *$space_char* ]];then
    tk_bin_path=${tk_bin_path%%$space_char*}
  fi
}


tk_chmod_whl()
{
  chd_value=$1
  if [[ -d "$tk_install_path/tk" ]];then
    chmod $chd_value -R $tk_install_path/tk
  fi
  if [[ -f "$tk_bin_path" ]];then
    chmod $chd_value  $tk_bin_path
  fi
}

tk_chmod_log()
```



```
{
  chd_value=$1
  log_file=$2
  if [[ -f "$log_file" ]];then
    chmod $chd_value $log_file
  fi
}

tk_check_os_inject()
{
  whl_name=$(basename "$1")
  if [[ ! "$whl_name" =~ ^([a-zA-Z0-9_\-\.]+)$ ]];then
    echo "Invalid Characters in whl package's name, please check it."
    exit 1
  fi
}


tk_install_whl()
{
  get_python_version

  whl_file_name=$(find $route -maxdepth 1 -type f -name "$tk_package_name*.whl")
  if [[ -f "$whl_file_name" ]];then
    tk_check_os_inject "$whl_file_name"
    echo "Begin to install mxTuningKit wheel package ($whl_file_name)."
    log_file="$HOME/.cache/Huawei/mxTuningKit/log/install-log.log"
    $python_version -m pip install "$whl_file_name" --log-file $log_file
    if test $? -ne 0; then
      echo "Install mxTuningKit wheel package failed."
    else
      tk_get_install_path
      tk_chmod_whl 550
      echo "Install mxTuningKit wheel package successfully."
    fi
  elif [[ "$whl_file_name" == *\n* ]];then
    echo "Only support a single version of whl package, please check it."
  else
    echo "There is no mxTuningKit wheel package to install."
  fi
  tk_chmod_log 640 $log_file
}

tk_uninstall_whl()
{
  tk_get_install_path
  if [ -n "$tk_install_path" ];then
    tk_chmod_whl 750
    log_file="$HOME/.cache/Huawei/mxTuningKit/log/uninstall-log.log"
    $python_version -m pip uninstall $tk_package_name --log-file $log_file
    if test $? -ne 0; then
      echo "Uninstall mxTuningKit wheel package failed."
    fi
  else
    echo "There is no mxTuningKit wheel package to uninstall. Please check if it has been installed."
  fi
  tk_chmod_log 640 $log_file
}

print_error()
{
  echo "command arguments are wrong! Valid arguments are necessary as follows: "
  echo "----1 install cmd:'bash tk_user.sh install'"
  echo "----2 uninstall cmd:'bash tk_user.sh uninstall'"
}

main()
{
  umask 027
```



```
  case $cmd_para in
    install)
      tk_install_whl
      ;;
    uninstall)
      tk_uninstall_whl
      ;;
    *)
      print_error
  esac
}

main
```


C 网络冻结样例代码

如果用户需要使用网络冻结能力，请参考以下步骤适配：

步骤1 在模型配置文件中的freeze关键词下，配置需要冻结的网络层名称。下方示例表示冻结bottleneck.block1.layer1、bottleneck.block1.layer2、bottleneck.fc.weight三层。

```
freeze:
  bottleneck:
    block1:
      layer1
      layer2
    fc:
      weight
```

步骤2 参考下方代码，将对应逻辑适配到模型代码中，实现网络冻结。

```
import os
import stat
import logging
import yaml
import argparse

"""
需要使用网络冻结能力时, 请参考如下实现

使用方式:
1、在模型启动脚本(boot_file_path)中, 通过argparse等参数定义工具定义入参'--advanced_config'
parser = argparse.ArgumentParser()
parser.add_argument('--advanced_config', type=str)
args = parser.parse_args()

2、获取模型启动脚本(boot_file_path)接收到的'--advanced_config'值
advanced_config_path = args.advanced_config

3、使用mindspore定义需要冻结/部分冻结的模型
model = mindspore.nn.Cell(...)

4、从advanced_config_path中解析网络冻结配置
freeze_layers = get_freeze_layers(advanced_config_path)

5、冻结网络
freeze_model(model, freeze_layers)
"""

CONN_WITH = '.'
FREEZE_KEY = 'freeze'

logging.getLogger().setLevel(logging.INFO)


def freeze_model(model, freeze_layers):
```



```
    """
    冻结网络指定部分
    :param model:网络模型
    :param freeze_layers:冻结部分, 值是一个字符串列表
    """
    if not freeze_layers:
        logging.info('freeze_layers is empty, no layers in model will be frozen.')
        return

    if not isinstance(freeze_layers, list):
        freeze_layers = list(freeze_layers)

    layer_list = []
    for layer in freeze_layers:
        layer_list.append({'layer': layer, 'exist': False})

    logging.info('freeze model start.')

    for name, param in model.parameters_and_names():
        for value in layer_list:
            if not isinstance(value.get('layer'), str):
                raise ValueError('freeze layer is not str, freeze layer: %s' % freeze_layers)
            if name.startswith(value.get('layer')):
                param.requires_grad = False
                value['exist'] = True
    for value in layer_list:
        if not value['exist']:
            logging.warning('layer: %s is not exist.', value.get('layer'))

    logging.info("freeze model finish.")


def get_freeze_layers(model_config_path):
    """
    从model config配置文件中, 解析出mindspore能够识别的需要冻结的网络层
    :param model_config_path: model config配置文件本地绝对路径
    :return: 需要冻结的网络层名称集合
    """
    if model_config_path is None or not str(model_config_path):
        logging.warning('param model_config_path is None or empty.')
        return []

    model_config_path = str(model_config_path)

    # 获取绝对路径
    model_config_path = os.path.abspath(model_config_path)

    # 软链接校验
    if os.path.islink(model_config_path):
        logging.warning('detect link path, stop parsing freeze configs from model config file.')
        return []

    # 路径真实性校验
    if not os.path.exists(model_config_path):
        logging.error('model config file path does not exist.')
        return []

    try:
        content = read_file(model_config_path)
    except Exception as ex:
        logging.error('exception occurred when reading model config file, detail error message: %s', ex)
        raise ex

    if FREEZE_KEY not in content.keys():
        logging.error('no [freeze] config found in model config file, no layers will be frozen.')
        return []

    freeze_info = content.get(FREEZE_KEY)
    if freeze_info is None:
```



```
        logging.error('[freeze] attribute is empty in model config file, check model config file.')
        return []

    if isinstance(freeze_info, str):
        return [freeze_info]

    expanded_dict = expand_dict(freeze_info)
    res = split_vals_with_same_key(expanded_dict)
    return res


def read_file(model_config_path):
    """
    读取配置文件
    """
    flags = os.O_RDWR | os.O_CREAT  # 允许读写, 文件不存在时新建
    modes = stat.S_IWUSR | stat.S_IRUSR  # 所有者读写

    with os.fdopen(os.open(model_config_path, flags, modes), 'rb') as file:
        content = yaml.safe_load(file)

    return content


def expand_dict(dict_info):
    """
    将网络冻结配置解析出的字典平铺化
    :param dict_info: 平铺前的配置字典
    :return: 平铺后的配置字典
    """
    common_prefix_dict = dict()

    for key_item, val_item in dict_info.items():
        if key_item is None:
            logging.error('find [none] key from [freeze] config in model config file, '
                          'config is ignored, check model config file.')
            continue

        if val_item is None:
            logging.error('attribute of key: [%s] is none, config is ignored, check model config file.',
str(key_item))
            continue

        if isinstance(val_item, dict):
            val_item = expand_dict(val_item)
            common_prefix_dict.update(get_prefix_dict(dict_info=val_item, prefix_str=str(key_item)))
        else:
            if str(key_item) in common_prefix_dict:
                logging.warning('find duplicate key from [freeze] part in model config file, check settings.')
            else:
                common_prefix_dict.update({str(key_item): [val_item for val_item in str(val_item).split(' ')]})

    return common_prefix_dict


def split_vals_with_same_key(expanded_dict):
    """
    对同一前缀下的多个子名称进行拆分
    :param expanded_dict: 平铺后的字典
    :return: 拆分后的完整名称列表
    """
    res = []
    for key_item, val_item in expanded_dict.items():
        for val in val_item:
            res.append(f'{str(key_item)}{CONN_WITH}{str(val)}')
    return res


def get_prefix_dict(dict_info, prefix_str):
```

```
    """
    获取包含前缀字典(多层嵌套使用)
    :param dict_info: 配置字典
    :param prefix_str: 前缀信息
    :return: 拼接前缀后的字典
    """
    return {prefix_str + CONN_WITH + str(k): v for k, v in dict_info.items()}
```

----结束

D 日志存放规则

日志存放路径为“$HOME/.cache/Huawei/mxTuningKit/log/”，对应目录结构参考如下。

```
$HOME/.cache/Huawei/mxTuningKit/
└────log
  ├──── node_0                    #节点号
    ├──── device_0          # 卡号
      ├──── service.log       # 业务日志
      ├──── operation.log           # 操作日志
    ├──── device_1
  ├──── node_1
 ...
```

- 其中，**卡号、节点号**会根据环境变量中的“DEVICE_ID”和“SERVER_ID”确定。
- 每条日志会按照如下格式记录：
  [*日志等级*] *时间 进程号* : *日志内容*

  如：**[INFO] 2022-09-08 15:49:51,408 [83390] [CLI] : xxxxx**
- 组件会捕捉用户模型运行时的标准输出，如print、logging等标准输出，会按照如下格式记录：
  时间 进程号 : *日志内容*

  如：**2022-09-08 15:49:51,408 [83390] [Model] : xxxxx**

E 模型适配样例

E.1 模型微调启动脚本适配样例（argparse）

```
import argparse
import logging

# 模拟模型源码逻辑
def model_finetune(args):
    logging.info('%s', args.data_path)    # 模型代码中可接收到对应参数
    logging.info('%s', args.output_path)
    logging.info('%s', args.advanced_config)
    logging.info('%s', args.learning_rate)


if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    # 必选适配项
    parser.add_argument('-dp', '--data_path', type=str, required=True)    # 数据集路径
    parser.add_argument('-op', '--output_path', type=str, required=True)    # 输出路径

    # 可选适配项
    parser.add_argument('-pm', '--pretrained_model_path', type=str, required=False)    # 预训练模型路径
    parser.add_argument('-ac', '--advanced_config', type=str, required=False)    # 模型配置文件路径, 仅当模型
需要网络冻结等高阶能力场景时需要

    # 模型配置文件中包含的params配置项
    parser.add_argument('-lr', '--learning_rate', type=float, required=False)
    parser.add_argument('-bs', '--batch_size', type=int, required=False)

    # 模型脚本接收的参数集合
    args = parser.parse_args()
    model_finetune(args)
```

对应模型配置文件的内容：

```
params:
 learning_rate: 1e-6
 batch_size: 32
```

E.2 模型评估启动脚本适配样例（argparse）

```
import argparse
import logging
import json
import stat
```



```
import os

DEFAULT_FLAGS = os.O_RDWR | os.O_CREAT
DEFAULT_MODES = stat.S_IWUSR | stat.S_IRUSR

# 模拟模型源码逻辑
def model_eval(args):
    logging.info('%s', args.data_path)    # 模型代码中可接收到对应参数
    logging.info('%s', args.output_path)
    logging.info('%s', args.model_config)
    logging.info('%s', args.learning_rate)
    eval_result = json.dumps({'评估结果': '待输出评估结果'})
    eval_result_path = os.path.join(args.output_path, 'eval_result.json')    # 此处必须以eval_result.json文件存储至指定输出路径,否则评估任务无结果返回，视为失败
    with os.fdopen(os.open(eval_result_path, DEFAULT_FLAGS, DEFAULT_MODES), 'w') as file:
        json.dump(eval_result, file)

# 用于argparse参数类型转换。
def str2bool(param):
    return param.lower() == 'true'

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    # 必选适配项
    parser.add_argument('-dp', '--data_path', type=str, required=True)    # 数据集路径
    parser.add_argument('-op', '--output_path', type=str, required=True)    # 输出路径
    parser.add_argument('-cp', '--ckpt_path', type=str, required=True)    # 模型结果路径

    # 模型配置文件中包含的配置项
    parser.add_argument('-ep', '--eval_param', type=float, required=False)
    parser.add_argument('-bp', '--bool_param', type=str2bool, required=False)  # argparse不支持传入bool类型参数，需要在type中转换参数类型

    # 模型脚本接收的参数集合
    args = parser.parse_args()
    model_eval(args)
```

对应模型配置文件的内容：

```
params:
 eval_param: 'your settings'
 bool_param: false
```

E.3 模型推理启动脚本适配样例

推理启动脚本与评估脚本适配方法相同，只需将落盘结果文件名需变更为“infer_result.json”。

F RANK_TABLE_FILE 生成方法

生成用于Ascend芯片分布式通信的芯片资源信息配置文件（ RANK_TABLE_FILE ）。

Ascend HCCL RANK_TABLE_FILE 文件提供Ascend分布式训练作业的集群信息，构建该文件需**下载hccl_tools.py脚本**， 详细使用介绍参考**README**。

```
# 如生成8卡的rank_table_file
python hccl_tools.py --device_num "[0,8)"
```