

Module 1---Research Agent Tutorial

一个典型的数据驱动任务Research流程为：

1. 梳理任务SoW (Statement of Work)，包括任务简介、任务Benchmark（训练集和测试集）、测评指标和脚本等；
2. 综述完成任务相关工作，对现有技术进行分类；
3. 实验筛选，将检索到的相关工作在已有任务Benchmark上进行适配和复现，选出效果最佳的工作；
4. 在已有工作上进行优化以进一步提升方案性能。

Research Agent的目标是将上述Research流程尽可能自动化由Agent完成，本文以LLM merge任务为例，以下是一个完整的Research Agent tutorial。

1. 面向Agent的任务SoW

任务简介

动机：

Training high-performing large language models (LLMs) from scratch is a notoriously expensive and difficult task, costing hundreds of millions of dollars in compute alone. These pretrained LLMs, however, can cheaply and easily be adapted to new tasks via fine-tuning, leading to a proliferation of models that suit specific use cases. Recent work has shown that specialized fine-tuned models can be rapidly merged to combine capabilities and generalize to new skills.

任务：

The competition will provide the participants with a list of expert models that have already been trained on a task-specific dataset. All of these models will be publicly available on the Hugging Face Model Hub with licenses that permit their use for research purposes. These models can either be fully fine-tuned models or models obtained by parameter-efficient fine-tuning methods such as LoRA. Models on this list will be required to satisfy the following criteria: (1) model size $\leq 8\text{B}$ parameters, and (2) model with licenses compatible with research use (e.b., [MIT](#), [Apache 2](#) etc).

The goal of this competition is to re-use the provided models to create a generalist model that can perform well on a wide variety of skills like reasoning, coding, maths, chat, and tool use.

This list of models will include popular pre-trained models such as LLaMA-7B, Mistral-7B, and Gemma-7B.

Allowed Models:

本次我们对允许的models做了限定，以减少任务的复杂度：The models to be merged are `meta-llama/Meta-Llama-3-8B-Instruct` and `MaziarPanahi/Llama-3-8B-Instruct-v0.8`.

Benchmark

HumanEval+：

数据地址：<https://huggingface.co/datasets/evalplus/humanevalplus/viewer?views%5B%5D=test>

数据构成：

Datasets: evalplus, humanevalplus		like 13	Follow	EvaPlus 17	Dataset card	Data Studio	Files and versions	Community
Split (1) test · 164 rows								
<input type="checkbox"/> Search this dataset								
task_id string · lengths	prompt string · lengths 	canonical_solution string · lengths 	entry_point string · lengths 	test string · lengths 				
HumanEval/0	from typing import List def has_close_elements(numbers: List[float], threshold: float) -> bool: """ Check if in given...	sorted_numbers = sorted(numbers) for i in range(len(sorted_numbers) - 1): if sorted_numbers[i + 1] - ...	has_close_elements	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/1	from typing import List def separate_paren_groups(paren_string: str) -> List[str]: """ Input to this function is a string. ...	cnt, group, results = 0, "", [] for ch in paren_string: if ch == "(": cnt += 1 if ch == ")": cnt -= 1 if ch != "=": group += ch... return number - int(number)	separate_paren_groups	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/2	def truncate_number(number: float) -> float: """ Given a positive floating point number, it can be decomposed into and...	return account = 0 for operation in operations: account += operation if bool: """ You're given a list of deposit and withdrawal...	truncate_number	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/3	from typing import List def below_zero(operations: List[int]) -> bool: """ You're given a list of deposit and withdrawal...	account = 0 for operation in operations: account += operation if bool: """ You're given a list of deposit and withdrawal...	below_zero	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/4	from typing import List def mean_absolute_deviation(numbers: List[float]) -> float: """ For a given list of input numbers,...	mean = sum(numbers) / len(numbers) return sum(abs(x - mean) for x in numbers) / len(numbers)	mean_absolute_deviation	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/5	from typing import List def intersperse(numbers: List[int], delimiter: int) -> List[int]: """ Insert a number 'delimiter'...	res = [] for i in range(len(numbers)): res.append(numbers[i]) if i != len(numbers) - 1: res.append(delimiter) return res	intersperse	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/6	from typing import List def parse_nested_parens(paren_string: str) -> List[int]: """ Input to this function is a string. ...	def count_depth(s: str) -> int: max_depth, cnt = 0, 0 for ch in s: if ch == "(": cnt += 1 if ch == ")": cnt -= 1 max_depth = ... return list(filter(lambda s: substring in s, strings))	parse_nested_parens	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/7	from typing import List def filter_by_substring(strings: List[str], substring: str) -> List[str]: """ Filter an input...	s, p = 0, 1 for number in numbers: s += number p *= number return s, p	filter_by_substring	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/8	from typing import List, Tuple def sum_product(numbers: List[int]) -> Tuple[int, int]: """ For a given list of integers...	return [max(numbers[:i+1]) for i in range(len(numbers))]	sum_product	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/9	from typing import List, Tuple def rolling_max(numbers: List[int]) -> List[int]: """ From a given list of integers,...	if is_palindrome(string): return string for i in range(len(string)): if is_palindrome(string[i]): return string...	rolling_max	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				
HumanEval/10	def is_palindrome(string: str) -> bool: """ Test if given string is a palindrome """ return string == string[::-1] def...	make_palindrome	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...					
HumanEval/11	from typing import List def string_xor(a: str, b: str) -> str: """ Input are two strings a and b consisting only of 0s and 1s...	return "".join(str(int(a[i]) ^ int(b[i]))) for i in range(len(a))	string_xor	import numpy as np def is_floats(x) -> bool: # check if it is float; List[float]; Tuple[float] if isinstance(x, float): return...				

task_id表示任务的ID， prompt表示题目（通常直接请求大模型获取答案）， entry_point是唯一标记， canonical_solution是参考答案， test是测试单元。

在原始 HumanEval 164 道 Python 题目基础上，对每道题新增约 80 倍测试用例，从而得到 HumanEval+，即用海量、高覆盖率的单元测试严格验证代码功能正确性。

评估指标

指标：Pass @ 1

1) 背景：为什么要用 pass @ k

传统代码评估用 BLEU／Edit Distance 等“文本相似度”指标，但这些**无法保证语义等价**。于是近几年工作 (Kulal 2019; Chen 2021) 转向**功能正确性**：只认“是否通过全部单元测试”。

当一次对同一题采样 k 份代码，只要 **至少 1 份** 通过，就视为“题目被解决”，这一概率就是 pass @ k ([arXiv][1])。

2) 一般公式（无偏估计器）

设

记号	含义
n	对该题实际生成的候选总数 ($\geq k$)
c	其中通过全部测试的候选数
k	我们关心的排名阈值

无偏估计器 (Chen 2021 式 (1)) :

$$\text{pass}@k := \mathbb{E}_{\text{Problems}} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right]$$

直观上，分子是“从不正确样本里抽到 k 份全错”的组合数，分母是“从全部 n 份里抽 k 份”的组合数；用 1 减去它即得到“至少 1 份对”的概率。

作者证明该表达式无偏，且比朴素估计 $1 - (1 - \hat{p})^k$ 方差更低，避免随着 n 变化而系统性低估 ([arXiv][1])。

3) $\text{pass}@1$ 的特化

令 $k = 1$:

$$\text{pass}@1 = 1 - \frac{\binom{n-c}{1}}{\binom{n}{1}} = \frac{c}{n}.$$

若一次仅生成 1 份代码 ($n = 1$)，此式退化为“该份代码是否通过”——也是最常见的实验设定。

若一次生成 $n > 1$ 份，仍可算出 $\text{pass}@1 = c/n$ ；它表示“随机从这 n 份里抽 1 份就通过”的期望成功率，而不是“第 1 个候选是否通过”。

4) 计算示例（官方实现）

```
def pass_at_k(n: int, c: int, k: int):
    if n - c < k:
        return 1.0
    import numpy as np
    return 1 - np.prod(1 - k / np.arange(n-c+1, n+1))
```

这是 Codex 论文给出的 **数值稳定实现**，避免大阶乘溢出 ([arXiv][1])。

单元测试覆盖度：若测试不全，`pass @ 1` 可能“漏网”错误实现——Humaneval+之所以把每题测试扩大 $\times 80$ ，就是为提高置信度。

5) 小结

`pass @ 1` 是 **概率论严格推导的无偏估计量**：衡量“从模型当前输出中随机抽 1 份，就能一次性通过全部单元测试”的可信概率。它兼具

可解释性（直接映射到开发者体验）、

公平性（无偏、与样本量解耦），

可操作性（一行 NumPy 代码即可计算）。

因此已成为 HumanEval、Humaneval+、MBPP+ 等主流代码基准的核心指标，并被 GitHub Copilot、OpenAI Codex、BigCode StarCoder 等系统广泛采用。

[1]: <https://arxiv.org/pdf/2107.03374.pdf> "Evaluating Large Language Models Trained on Code"

测评脚本

代码块

```
1 conda create -n bigcode python=3.10.9
2 conda activate bigcode
3
4 git clone https://github.com/bigcode-project/bigcode-evaluation-harness.git
5 cd bigcode-evaluation-harness
6
7 pip install -e .
8 pip install -U torch>=2.2 torchvision torchaudio
9 pip install numpy==1.24.1
10
11 CUDA_VISIBLE_DEVICES=1 accelerate launch main.py \
```

```
12 --model $MODEL \      # 替换为你的模型地址
13 --max_length_generation 512 \
14 --precision bf16 \
15 --tasks humanevalplus \
16 --temperature 0.2 \
17 --n_samples 10 \
18 --batch_size 10 \
19 --allow_code_execution \
20 --metric_output_path $OUTPUT_PATH/code_eval.json \ # 替换为你的输出地址
21 --use_auth_token
```

得出评估结果：

pass@1: 0.5060975609756098

code_eval.json样例如下：

代码块

```
1  {
2      "humanevalplus": {
3          "pass@1": 0.5060975609756098,
4          "pass@10": 0.6646341463414634
5      },
6      "config": {
7          "prefix": "",
8          "do_sample": true,
9          "temperature": 0.2,
10         "top_k": 0,
11         "top_p": 0.95,
12         "n_samples": 10,
13         "eos": "<|endoftext|>",
14         "seed": 0,
15         "model": "../models--meta-llama--Meta-Llama-3-8B-
Instruct/snapshots/e1945c40cd546c78e41f1151f4db032b271faeaa",
16         "modeltype": "causal",
17         "peft_model": null,
18         "revision": null,
19         "use_auth_token": true,
20         "trust_remote_code": false,
21         "tasks": "humanevalplus",
22         "instruction_tokens": null,
23         "batch_size": 10,
24         "max_length_generation": 512,
25         "precision": "bf16",
26         "load_in_8bit": false,
27         "load_in_4bit": false,
```

```
28     "left_padding": false,
29     "limit": null,
30     "limit_start": 0,
31     "save_every_k_tasks": -1,
32     "postprocess": true,
33     "allow_code_execution": true,
34     "generation_only": false,
35     "load_generations_path": null,
36     "load_data_path": null,
37     "metric_output_path": "code_eval.json",
38     "save_generations": false,
39     "load_generations_intermediate_paths": null,
40     "save_generations_path": "generations.json",
41     "save_references": false,
42     "save_references_path": "references.json",
43     "prompt": "prompt",
44     "max_memory_per_gpu": null,
45     "check_references": false
46 }
```

2. 任务相关工作综述

本小节的目标是找到尽可能系统、全面的相关工作及其开源项目地址，以供后续Experiment Agent实验，选出最佳工作

利用Deep Research查询相关工作

登入google gemini: <https://gemini.google.com/app>

【选择Deep Research】

输入下述提示词：

代码块

- 1 模型融合(Model Merge)是机器学习领域一种高效的赋能技术，它无需收集原始训练数据，也不需要昂贵的计算资源。请帮我综述现有大模型融合的相关工作，要求：
 - 1、尽可能多的检索相关工作，按照技术特征进行分类，并讨论这些方法的优缺点；
 - 2、如果有开源项目，请帮我梳理出开源地址；
 - 3、仅限于大模型的模型融合方法。

生成综述报告如下：

 大模型融合技术综述_.pdf

人工筛选

由于现有Deep Research工具生成的综述报告，只提供了有限的几个代表工作，为了进一步检索更加系统和全面的工作，还需要人工基于检索到的综述报告去进行梳理和查找，一般步骤如下：

- 1、以Deep Research检索到的工作为基础，查看里面是否检索到本领域综述论文，一般成熟且系统研究的领域会有最新综述论文，研究人员会将已有工作进行梳理和分类；
- 2、如果Deep research没有检索到综述论文，则需要人工确认是否被落下，人工在Google scholar、Arxiv等平台手动检索确认；
- 3、如果人工确认没有相关领域综述，则以Deep Research检索到的论文为起点，分析论文里的相关工作，可按照Deep research给出的分类，人工检索更多论文和开源项目。

在本任务中，Deep research检索到了本领域的最新综述论文：

《Model Merging in LLMs, MLLMs, and Beyond: Methods, Theories, Applications and Opportunities》

论文地址：<https://arxiv.org/pdf/2408.07666>

该综述已经将每项工作按照技术特征进行分类，开源地址为：

<https://github.com/EnnengYang/Awesome-Model-Merging-Methods-Theories-Applications.git>

根据综述可知，目前LLM Merge领域的相关工作分为以下几类（每类工作仅列举5项开源项目作为后续实验选项）：

基于权重的合并方法

这类方法旨在为不同的模型或任务向量分配不同的重要性权重，从而更有效地合并模型。

代表工作：

- RegMean++: Enhancing Effectiveness and Generalization of Regression Mean for Model Merging

论文地址：<https://arxiv.org/pdf/2508.03121>

github地址：<https://github.com/nthehai01/RegMean-plusplus>

- CALM: Consensus-Aware Localized Merging for Multi-Task Learning (ICML 2025)

论文地址：<https://arxiv.org/pdf/2506.13406>

github地址：<https://github.com/yankd22/CALM/tree/main>

- Arcee's MergeKit: A Toolkit for Merging Large Language Models

论文地址: <https://arxiv.org/pdf/2403.13257.pdf>

github地址: <https://github.com/arcee-ai/MergeKit>

- Evolutionary Optimization of Model Merging Recipes

论文地址: <https://arxiv.org/pdf/2403.13187.pdf>

github地址: <https://github.com/SakanaAI/evolutionary-model-merge>

- *Sens-Merging: Sensitivity-Guided Parameter Balancing for Merging Large Language Models*

论文地址: <https://arxiv.org/pdf/2502.12420.pdf>

github未发布，这篇文章的单位之一是华为诺亚方舟实验室，华为开源了一个toolkit

(<https://github.com/hahahawu/Long-to-Short-via-Model-Merging.git>)，将一系列llm merging的论文集成在这个toolkit里面。但这篇文章的代码还没集成进来。可测试这个toolkit已集成的其他方法：

- Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch <https://github.com/yule-BUAA/MergeLM>
- TIES-Merging: Resolving Interference When Merging Models
<https://github.com/prateeky2806/ties-merging>

基于子空间的合并方法

这类方法旨在将多个模型投影到稀疏子空间中进行合并，从而减轻任务间的干扰。

代表作：

- Training-free LLM Merging for Multi-task Learning (ACL2025)

论文: <https://arxiv.org/pdf/2506.12379.pdf>

github地址: <https://github.com/Applied-Machine-Learning-Lab/Hi-Merging>

- Adaptive LoRA Merge with Parameter Pruning for Low-Resource Generation (ACL2025)

论文: <https://arxiv.org/pdf/2505.24174.pdf>

github地址: https://github.com/mr0223/adaptive_lora_merge

- LoRI: Reducing Cross-Task Interference in Multi-Task LowRank Adaptation (COLM2025)

论文: <https://arxiv.org/pdf/2504.07448>

github地址: <https://github.com/juzhengz/LoRI>

- AdaRank: Adaptive Rank Pruning for Enhanced Model Merging

论文: <https://arxiv.org/pdf/2503.22178>

github地址: <https://github.com/david3684/AdaRank>

- STAR: Spectral Truncation and Rescale for Model Merging (NAACL 2025)

论文: <https://arxiv.org/pdf/2502.10339>

github地址: <https://github.com/IBM/STAR>

- Task Vector Quantization for Memory-Efficient Model Merging (ICCV 2025)

论文: <https://arxiv.org/pdf/2503.06921>

github地址: <https://github.com/AM-SKKU/TVQ>

基于路由的合并方法

这类方法是一种动态合并策略，根据输入样本的特征在推理阶段动态地决定如何合并模型。

代表工作:

- Dynamic Fisher-weighted Model Merging via Bayesian Optimization (NAACL2025)

论文: <https://arxiv.org/pdf/2504.18992>

github地址: <https://github.com/sanwooo/df-merge>

- MASS: MoErging through Adaptive Subspace Selection

论文: <https://arxiv.org/pdf/2504.05342>

github地址: <https://github.com/crisostomi/mass>

- CAMEX: CURVATURE-AWARE MERGING OF EXPERTS (ICLR2025)

论文: <https://arxiv.org/pdf/2502.18821>

github地址: <https://github.com/kpup1710/CAMEX>

- DAWIN: TRAINING-FREE DYNAMIC WEIGHT INTERPOLATION FOR ROBUST ADAPTATION (ICLR 2025)

论文: <http://arxiv.org/pdf/2410.03782>

github地址: <https://github.com/naver-ai/dawin>

- Learning to Route Among Specialized Experts for Zero-Shot Generalization

论文: <https://arxiv.org/pdf/2402.05859>

Github地址: <https://github.com/r-three/phatgoose>

基于后校准的合并方法

代表作

- Why Train Everything? Tint a Single Layer for Multi-task Model Merging

论文: <https://arxiv.org/pdf/2412.19098>

github地址: <https://github.com/AIM-SKKU/ModelTinting>

- Fine-tuning Aligned Classifiers for Merging Outputs: Towards a Superior Evaluation Protocol in Model Merging (ICML 2024)

论文: <https://arxiv.org/pdf/2412.13526>

github地址: <https://github.com/fskong/FT-Classifier-for-Model-Merging>

- SurgeryV2: Bridging the Gap Between Model Merging and Multi-Task Learning with Deep Representation Surgery

论文: <https://arxiv.org/pdf/2410.14389>

github地址: <https://github.com/EnnengYang/SurgeryV2>

- Representation Surgery for Multi-Task Model Merging (ICML 2024)

论文: <https://openreview.net/pdf/602906ec02919eb95d78d634321fcba1b68a2f03.pdf>

github地址: <https://github.com/EnnengYang/RepresentationSurgery>

小结

综述任务相关工作对于领域研究非常重要，现有Deep Research可以根据现有技术特征进行分类，但是仅仅输出代表性的几项工作，而Research Agent的目标是找出尽可能系统、全面的找到本任务相关

的所有工作，通过实验从其中选择最佳工作，当前的Deep Research无法满足要求，仅仅只能作为相关工作检索的入口。

思考：

如何优化现有Deep Research工作以使其满足Research Agent要求？

3. Experiment Agent实验适配

openhands环境安装

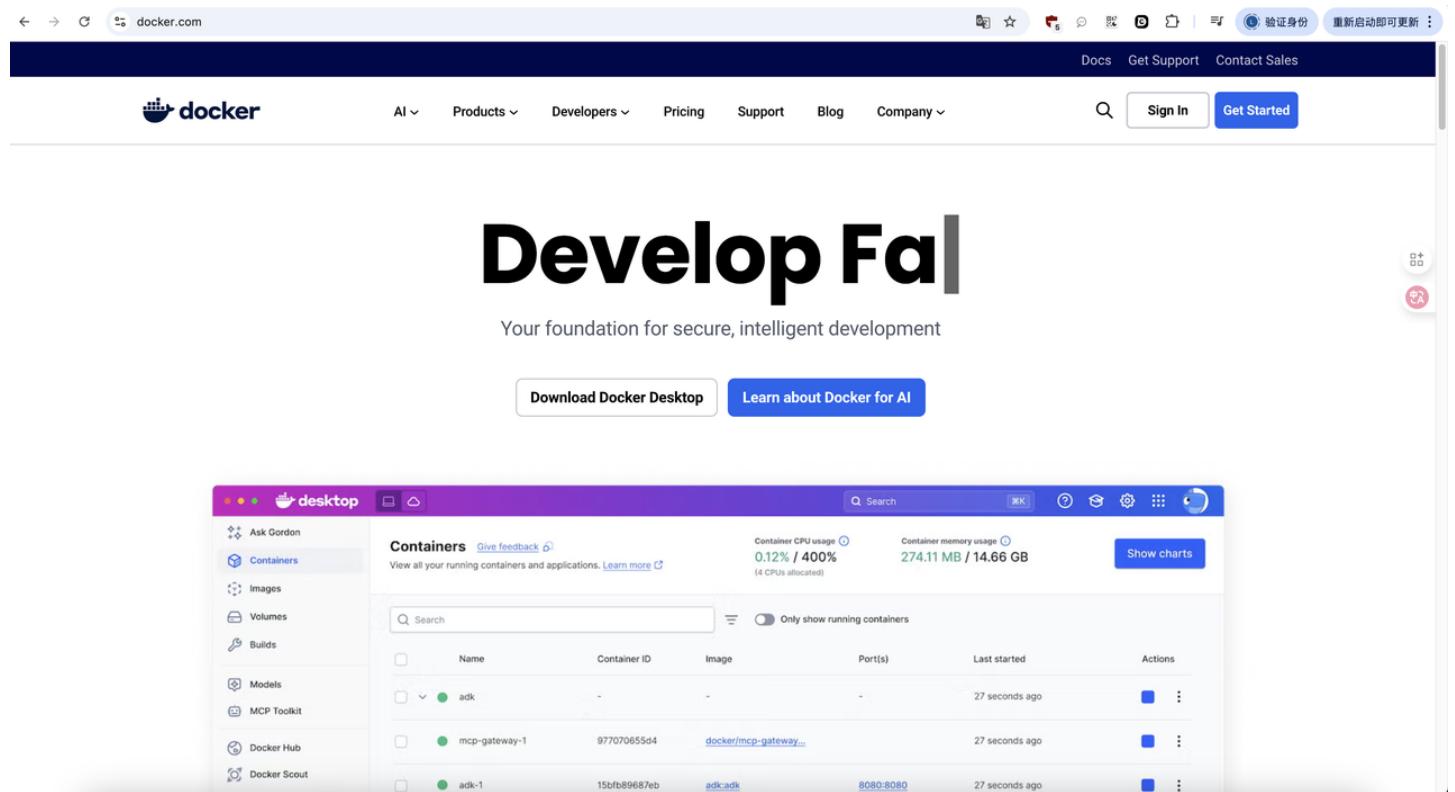
依赖准备

Docker 安装

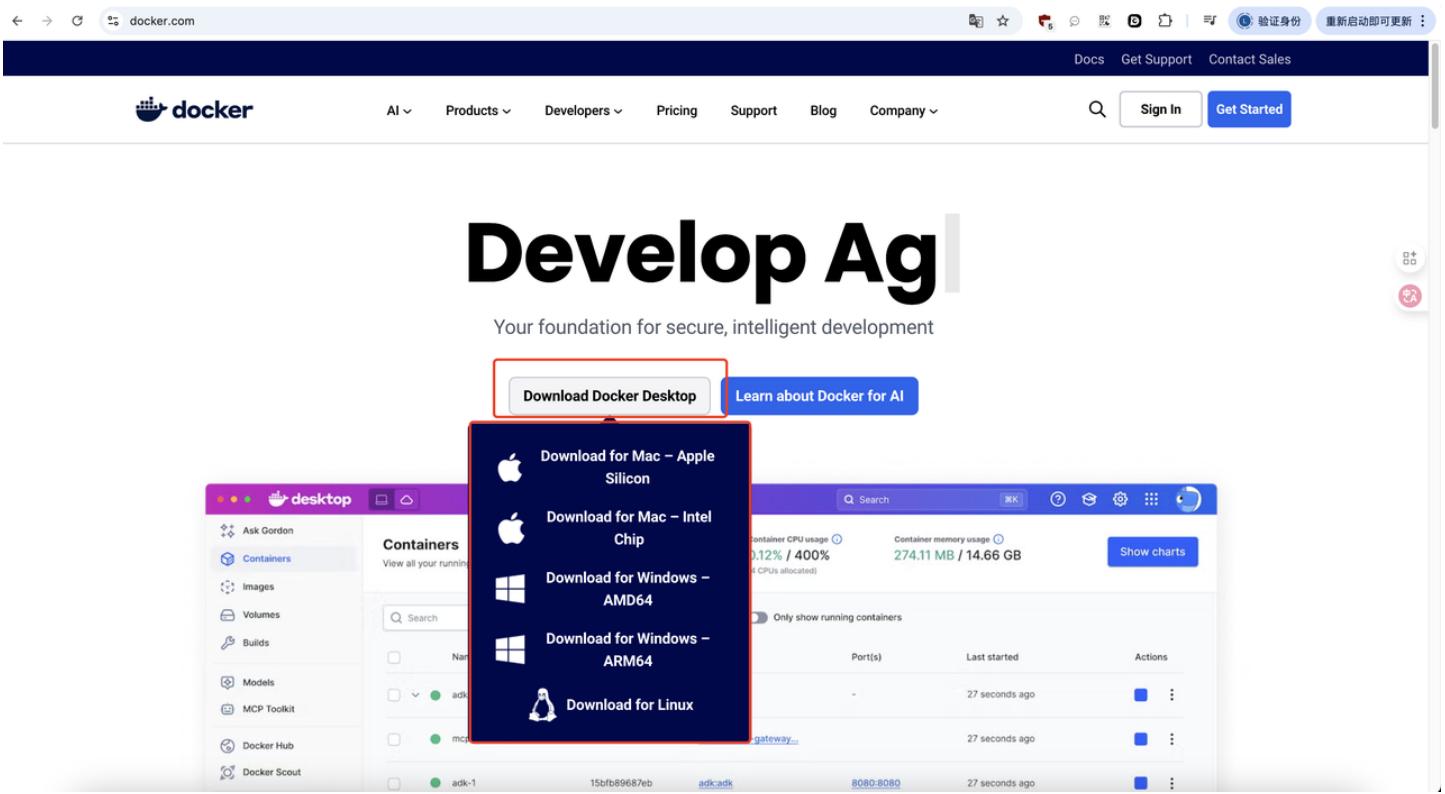
软件下载

官网地址: <https://www.docker.com/>

点击地址进入官网



根据自身电脑版本下载安装包

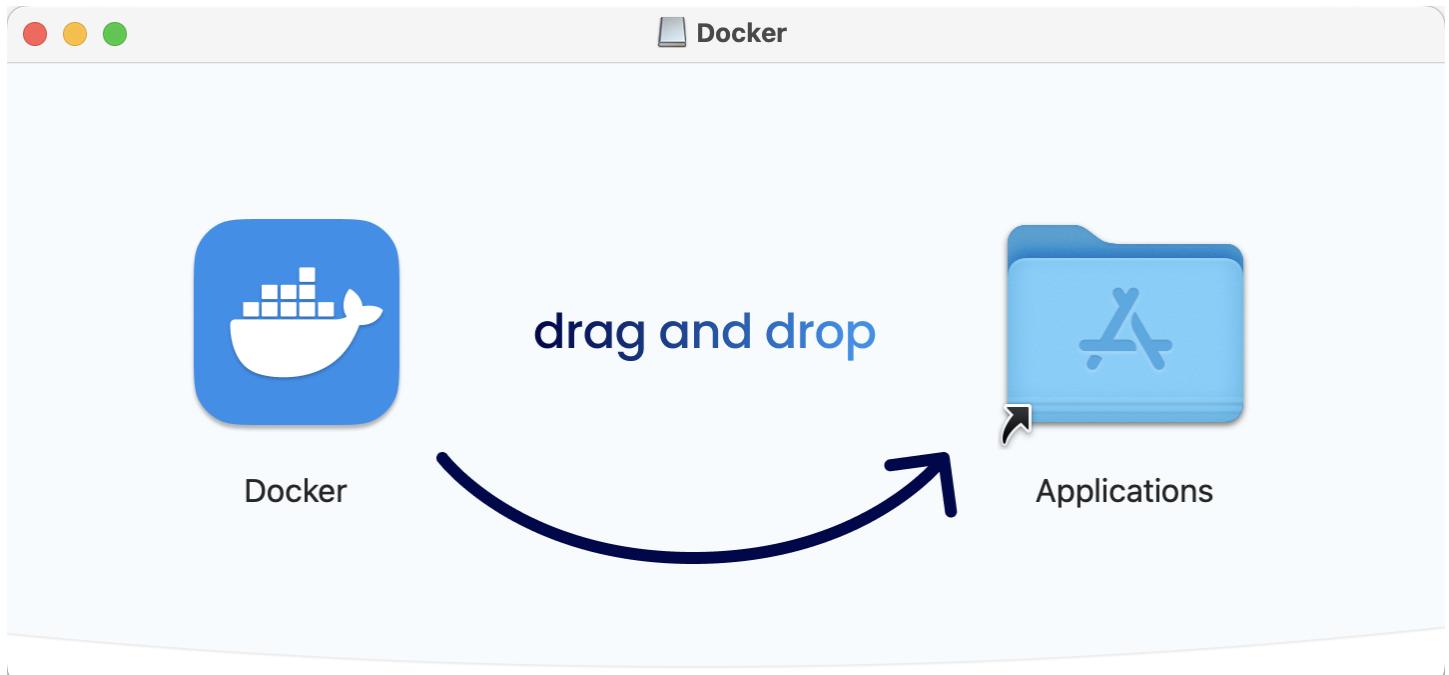


软件安装

[Windows安装参考视频](#)

MacOS:

打开安装包,拖拽图标到Applications



软件启动

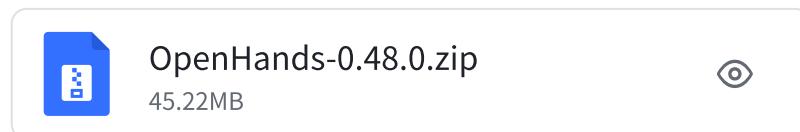
点击Docker Desktop应用打开界面,启动docker完成

Docker Desktop interface showing the Containers tab. It displays three running containers: my-nginx (nginx:latest), mysql8 (mysql:8.0), and mindflow. The interface includes a search bar, CPU usage stats (0.50% / 800%), memory usage stats (687.66MB / 22.89GB), and a charting link. The status bar at the bottom shows system resources like RAM and CPU usage.

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
my-nginx	29d11fbf3913	nginx:latest	80:80	0%	5 months ago	[Edit, More, Delete]
mysql8	11e9fdab1bb2	mysql:8.0	3306:3306	0.5%	5 days ago	[Edit, More, Delete]
mindflow	-	-	-	0%	1 day ago	[Edit, More, Delete]

Openhands-0.48.0

源码下载:

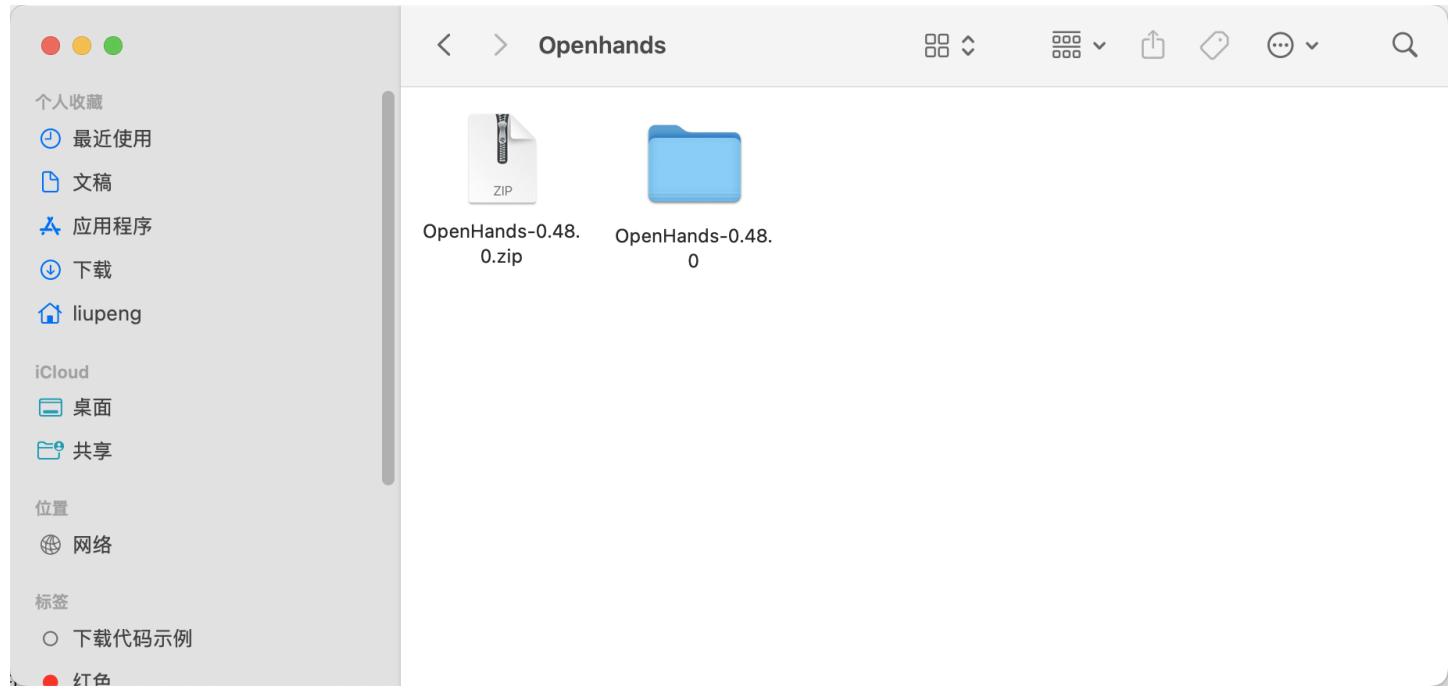


点击下载地址下载源码压缩包

A Mac OS X Finder window titled "Openhands". Inside the window, there is a single item: "OpenHands-0.48.0.zip". The sidebar on the left shows various folder categories like Personal, Recent, and Downloads.

File	Size
OpenHands-0.48.0.zip	45.22MB

解压缩文件夹



镜像准备:

终端执行命令

代码块

```
1 docker pull ghcr.io/all-hands-ai/runtime:0.48-nikolaik
```

```
Last login: Tue Aug 26 18:22:19 on ttys089
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/H1208050.
(base) lpMacBook-Pro:~ liupeng$ docker pull ghcr.io/all-hands-ai/runtime:0.48-nikolaik
```

```
(base) lpMacBook-Pro:~ liupeng$ docker pull ghcr.io/all-hands-ai/runtime:0.48-nikolaik
0.48-nikolaik: Pulling from all-hands-ai/runtime
8fbff1dd6492c: Already exists
c5e137b9ec17: Already exists
14f5719d6358: Already exists
78f00d2fce16: Already exists
7b9f5f869a6e: Already exists
0bbff3be7591: Already exists
34294ea665eb: Already exists
e2de77e4ff9b: Already exists
4f4fb700ef54: Already exists
2da3aa321bc6: Already exists
6eb5cf8c1997: Already exists
c0b663c72021: Already exists
920006c4bcdcd: Already exists
d24c281f6d77: Already exists
a816b2ad2180: Already exists
375b3d326689: Already exists
ab7f3a44fe3: Already exists
85d7bf06d575: Already exists
200f1be9d431: Already exists
94590749f5d6: Already exists
3e8fb36c59c0: Already exists
02e6f9e9355c: Already exists
d3d6521450f6: Already exists
d5af86ea2209: Already exists
d6c42738892c: Already exists
e16e4e5f3abb: Already exists
3de1a7e6b119: Already exists
4f19ecac4e35: Already exists
16cd45196016: Already exists
701bb818876e: Already exists
Digest: sha256:9dd8fc082ffd875b502deef0b78f2d4ded058cb831b97f73704761ac0ceccc6a
Status: Downloaded newer image for ghcr.io/all-hands-ai/runtime:0.48-nikolaik
ghcr.io/all-hands-ai/runtime:0.48-nikolaik
```

下载完成后

执行镜像查看命令查看镜像

代码块

1 docker images

```
● ● ●
Last login: Tue Aug 26 19:00:07 on ttys009
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT2008050.
(base) lpMacBook-Pro:~ liupeng$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
dyfadet            latest   0727b8e81b0a  2 weeks ago  9.81GB
<none>              <none>   b61dcad78a3e  3 weeks ago  9.64GB
dlinear             latest   af668e78ef10  3 weeks ago  9.24GB
ghcr.io/all-hands-ai/runtime  local   d51466800246  3 weeks ago  7.78GB
ghcr.io/all-hands-ai/runtime  0.48-nikolaik  95441abe3af2  7 weeks ago  6.92GB
ghcr.io/all-hands-ai/runtime  0.47-nikolaik  5f93bb44fd09  8 weeks ago  6.97GB
nfl                 latest   9b4785fc2c3c  2 months ago  11.8GB
nfl                 local    e46a644e99e1  2 months ago  14.1GB
text_gcn            latest   0382cb9aaef7  2 months ago  11.3GB
all-hands-ai/runtime  local    a07df05a5f57  2 months ago  5.92GB
<none>              <none>   78259cb172e7  2 months ago  10GB
ghcr.io/all-hands-ai/runtime  0.40-nikolaik  29fdf203d5fb  2 months ago  7.58GB
dlinear             openhands  26277bc3829f  3 months ago  6.3GB
dlinear             dlinear   cb3dd9779989  3 months ago  7.65GB
epf                latest   fccdfab0f899  3 months ago  10.1GB
ghcr.io/all-hands-ai/runtime  0.38-nikolaik  ccc29dc71b45  3 months ago  6.21GB
all-hands-ai/runtime  dlinear   4dee1e8da741  4 months ago  7.99GB
python              3.9-slim  9a0431538811d  4 months ago  126MB
sonatype/nexus3     latest   bc4931444c0dd  4 months ago  640MB
node               18-alpine  ee77c6cd7c18  5 months ago  127MB
node               18-slim   fc665eaf5031  5 months ago  192MB
briefercloud/briefer-api  latest   16bdb4fad9ed  5 months ago  3.65GB
moby/buildkit       buildx-stable-1  37895bc5b3fd  5 months ago  210MB
dataagent/web       latest   8f4f912eee66  5 months ago  173MB
code_adaptation_agent  latest   cca69e37a10e  6 months ago  6.82GB
mindflow-jupyter_server  latest   1d06191da663  6 months ago  2.98GB
mindflow-db_migration  latest   6da6eff937d88  6 months ago  777MB
mindflow-ecommerce   latest   16bdbe23d258  6 months ago  451MB
postgres            latest   1ec2b14946bd  6 months ago  459MB
nikolaik/python-nodejs  python3.12-nodejs22  05048535369f  6 months ago  1.38GB
nginx              <none>   2c9168b3c9a8  6 months ago  197MB
python              3.12-slim  e868845c8f83  6 months ago  150MB
hello-world         latest   f1f77a0f96b7  7 months ago  5.2KB
mysql              8.0      5ea077402e99  7 months ago  760MB
tjbtech1/paperagent  latest   4eac24296abc  7 months ago  41.4GB
python              3.10-slim  48ceb3b1c775  8 months ago  153MB
pytorch/pytorch     2.4.0-cuda12.4-cudnn9-runtime  e2a75ae5a502  13 months ago  7.85GB
```

可以在docker desktop中查看

Docker Desktop interface showing the Images tab. The search bar at the top has "ghcr.io/all-hands-ai/x" entered. A table below lists one image:

Name	Tag	Image ID	Created	Size	Actions
ghcr.io/all-hands-ai/runtime	0.48-nikolaik	95441abe3af2	2 months ago	6.91 GB	

At the bottom, status information shows "Engine running", "RAM 10.27 GB CPU 0.13%", "Disk: 163.94 GB used (limit 1006.85 GB)", and "Showing 1 item".

依赖安装

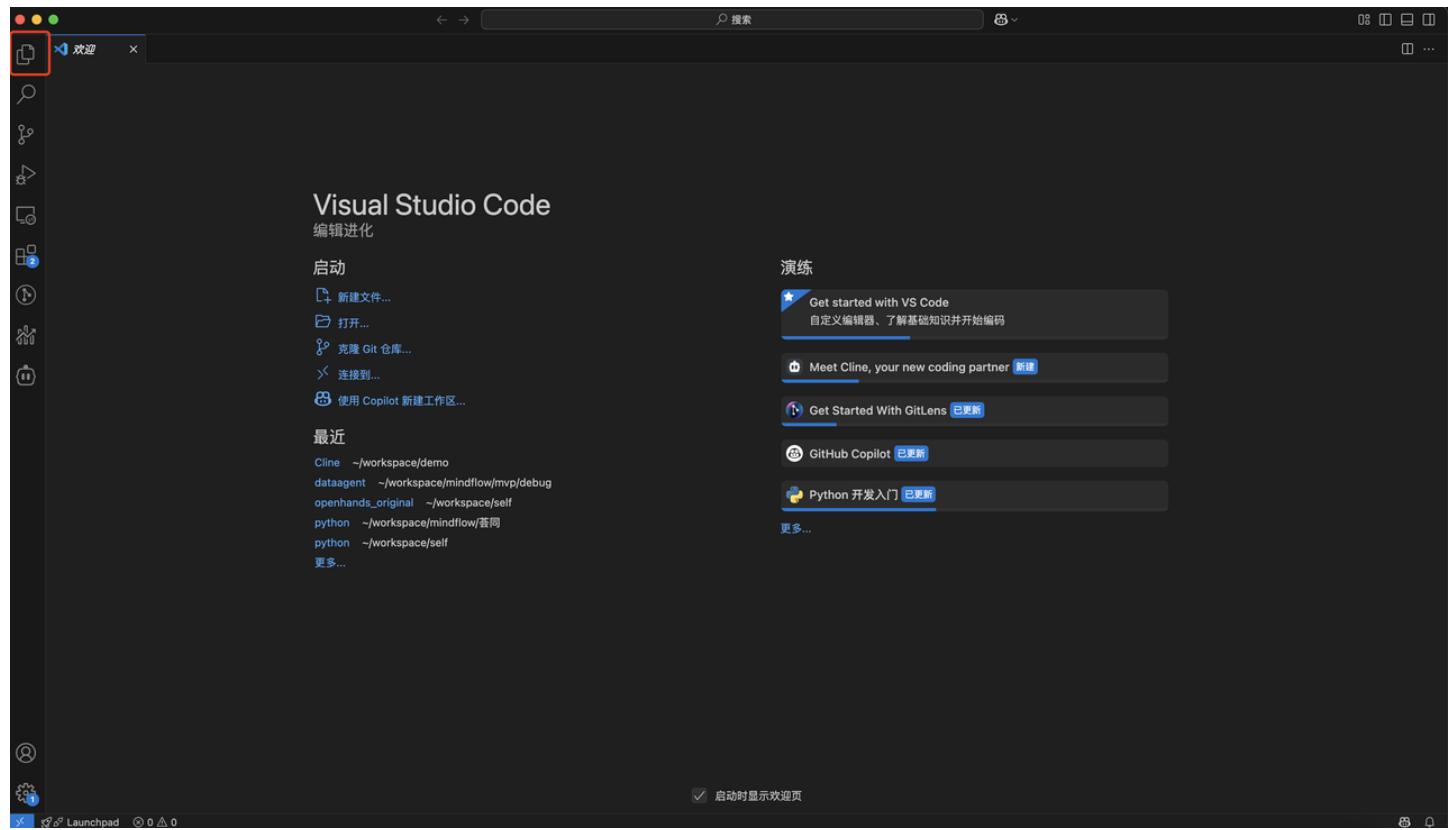
1、打开VSCode

Visual Studio Code welcome screen. The sidebar on the left includes icons for File, Edit, View, Insert, Search, and Terminal. The main area displays the following content:

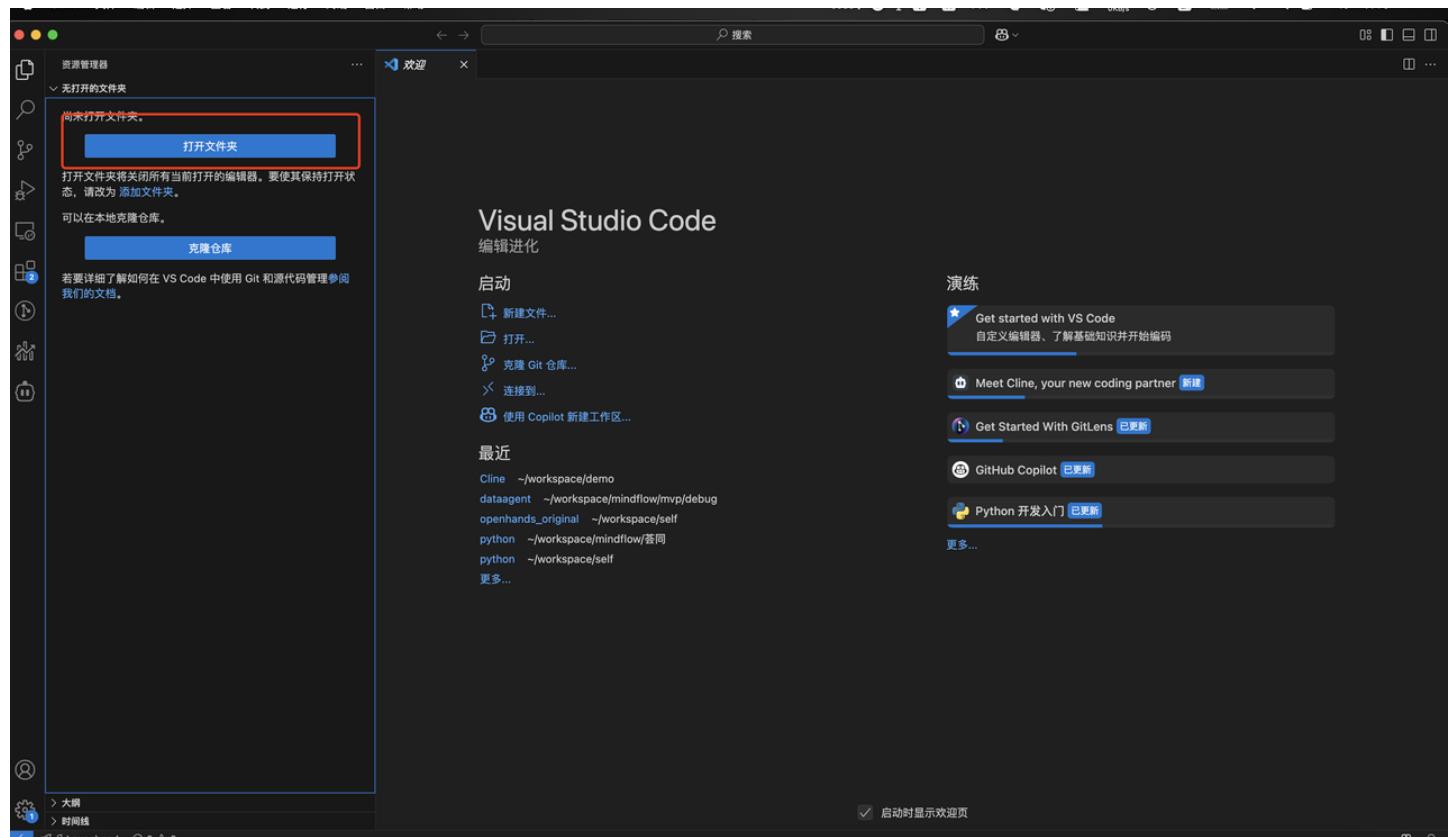
- 启动**:
 - 新建文件...
 - 打开...
 - 克隆 Git 仓库...
 - 连接到...
 - 使用 Copilot 新建工作区...
- 最近**:
 - Cline ~/workspace/demo
 - dataagent ~/workspace/mindflow/mvp/debug
 - openhands_original ~/workspace/self
 - python ~/workspace/mindflow/套同
 - python ~/workspace/self
- 演练**:
 - Get started with VS Code (自定义编辑器、了解基础知识并开始编码)
 - Meet Cline, your new coding partner (新建)
 - Get Started With GitLens (已更新)
 - GitHub Copilot (已更新)
 - Python 开发入门 (已更新)

At the bottom, there is a checkbox for "启动时显示欢迎页" (Show welcome page on start) and a status bar showing "Launchhead" and "0.0.0.0:4000".

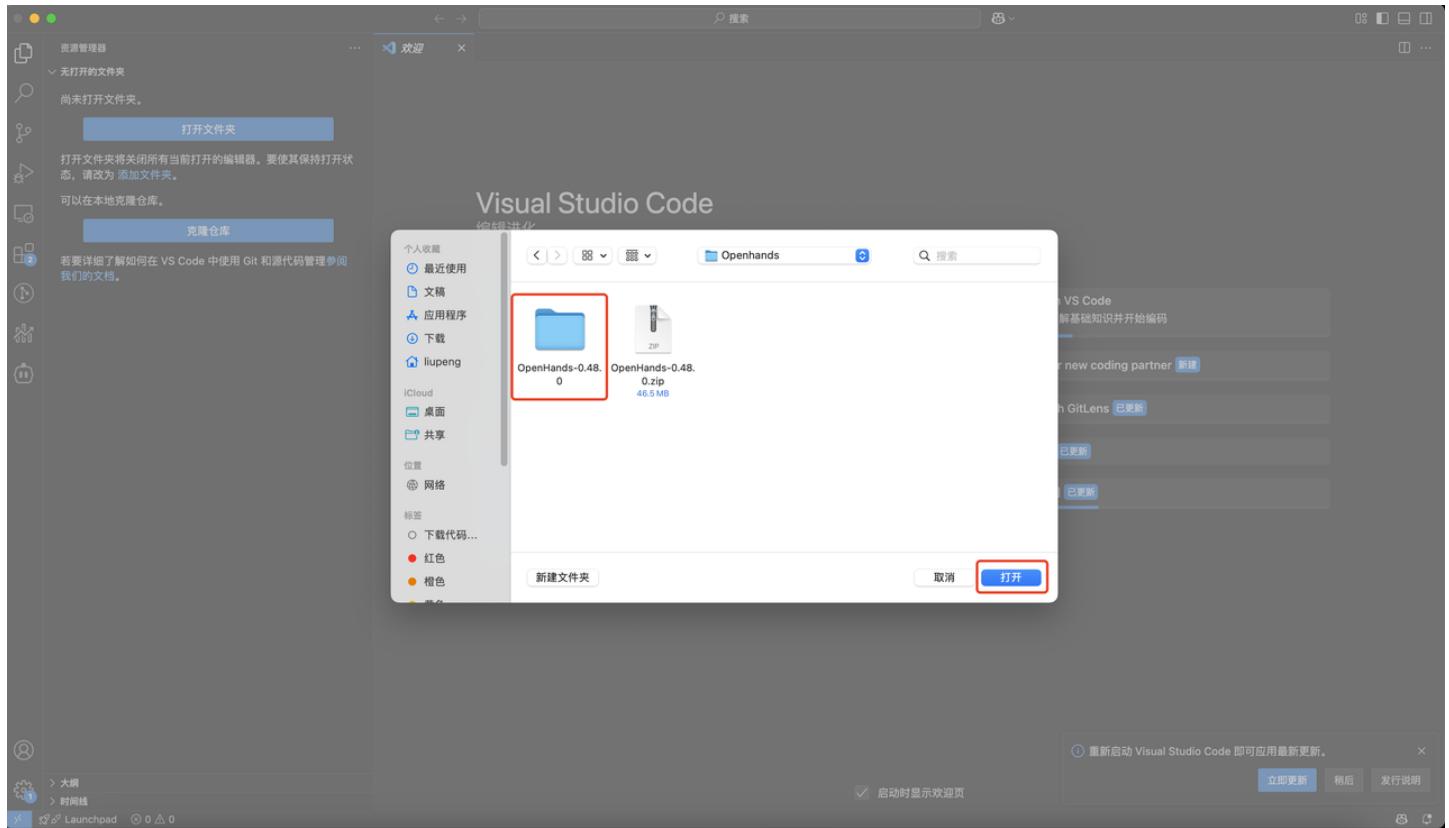
2、打开资源管理器



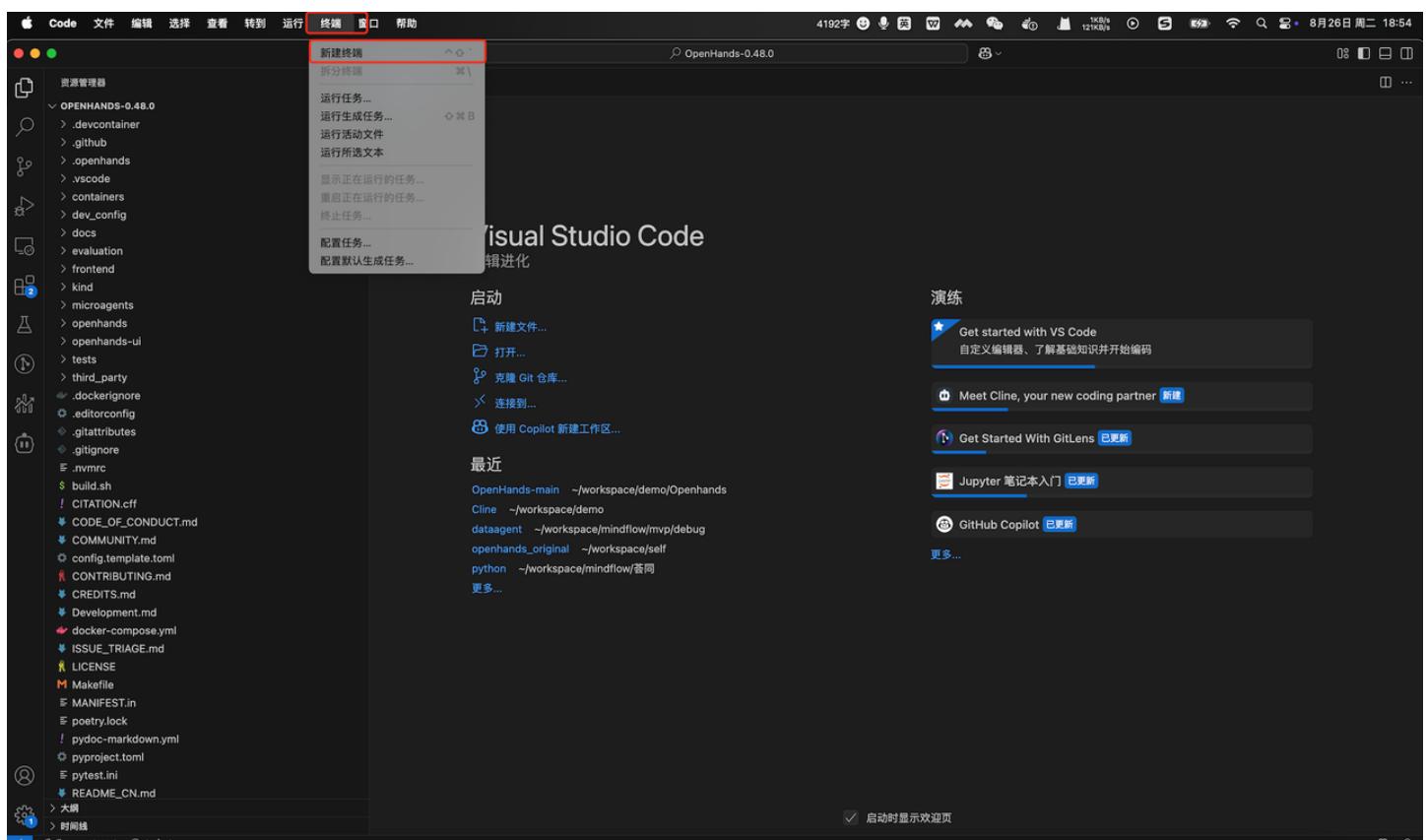
3、点击打开文件夹



4、选择解压完成的openhands代码文件夹并点击打开按钮



5、新建终端

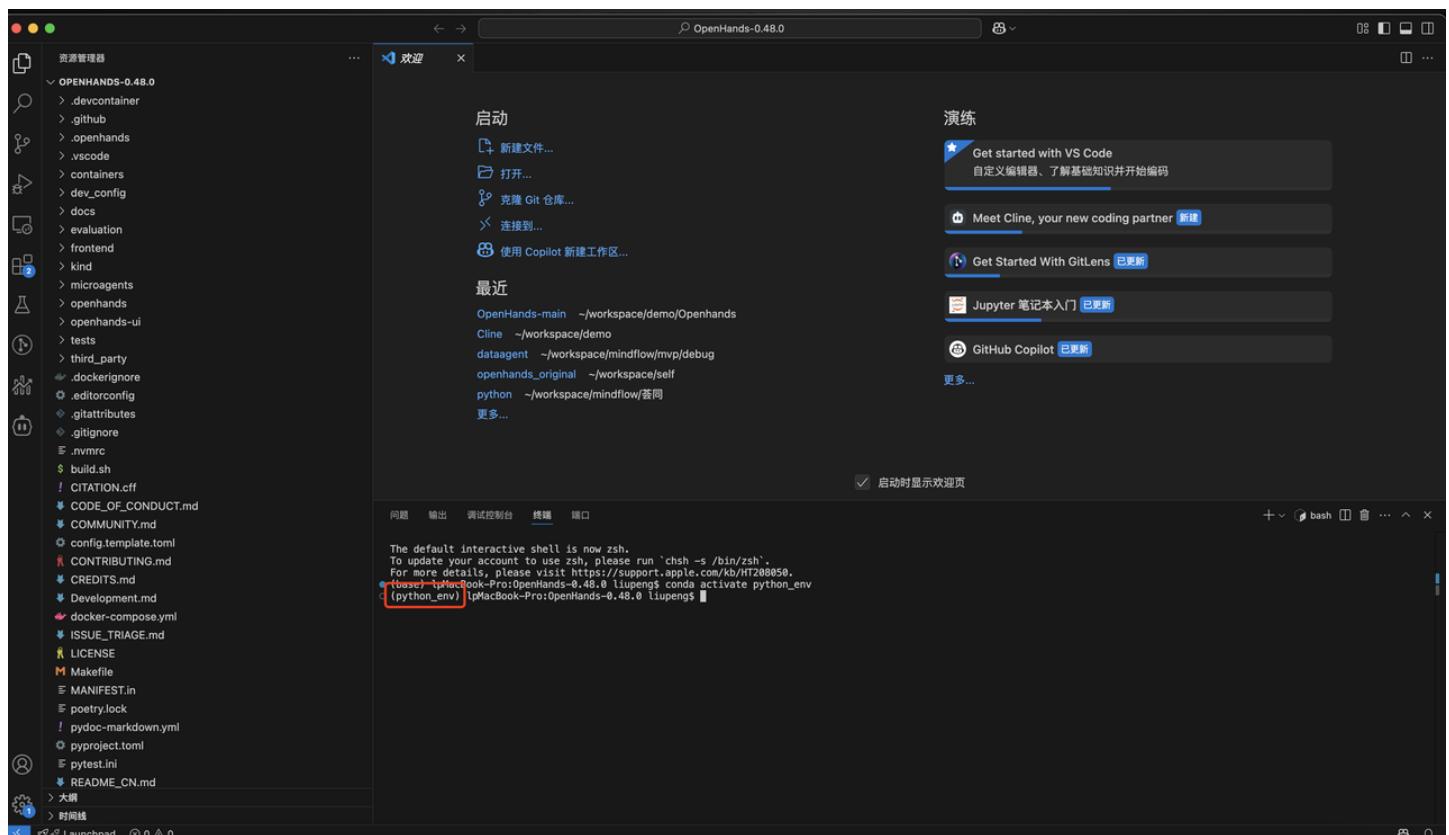
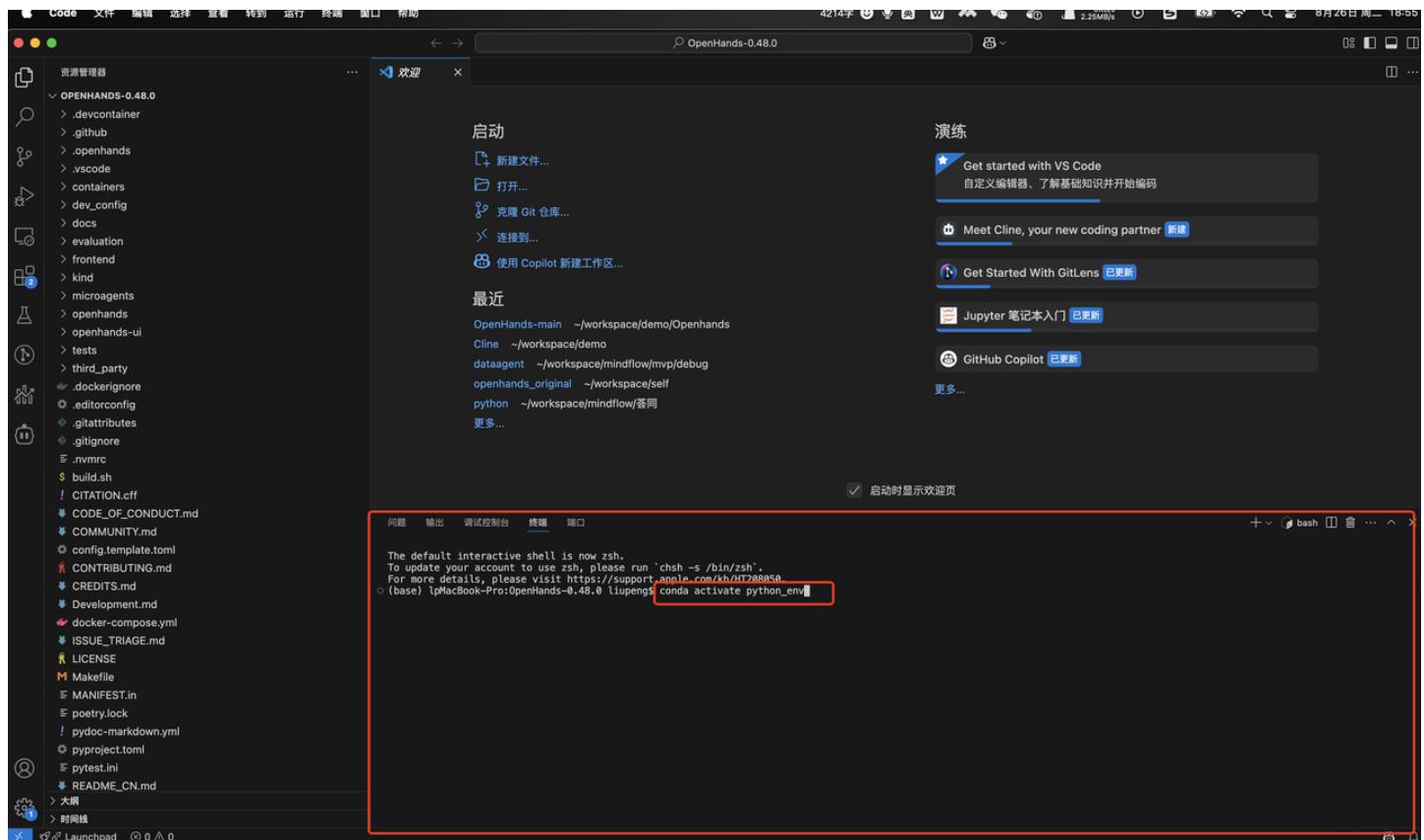


6、选择创建的自定义的python环境,若没有创建参考第二部分yhton环境准备（此处的python环境必须>=3.12）中Conda下的创建自定义python环境

终端输入以下代码并执行

代码块

1 conda activate python_env

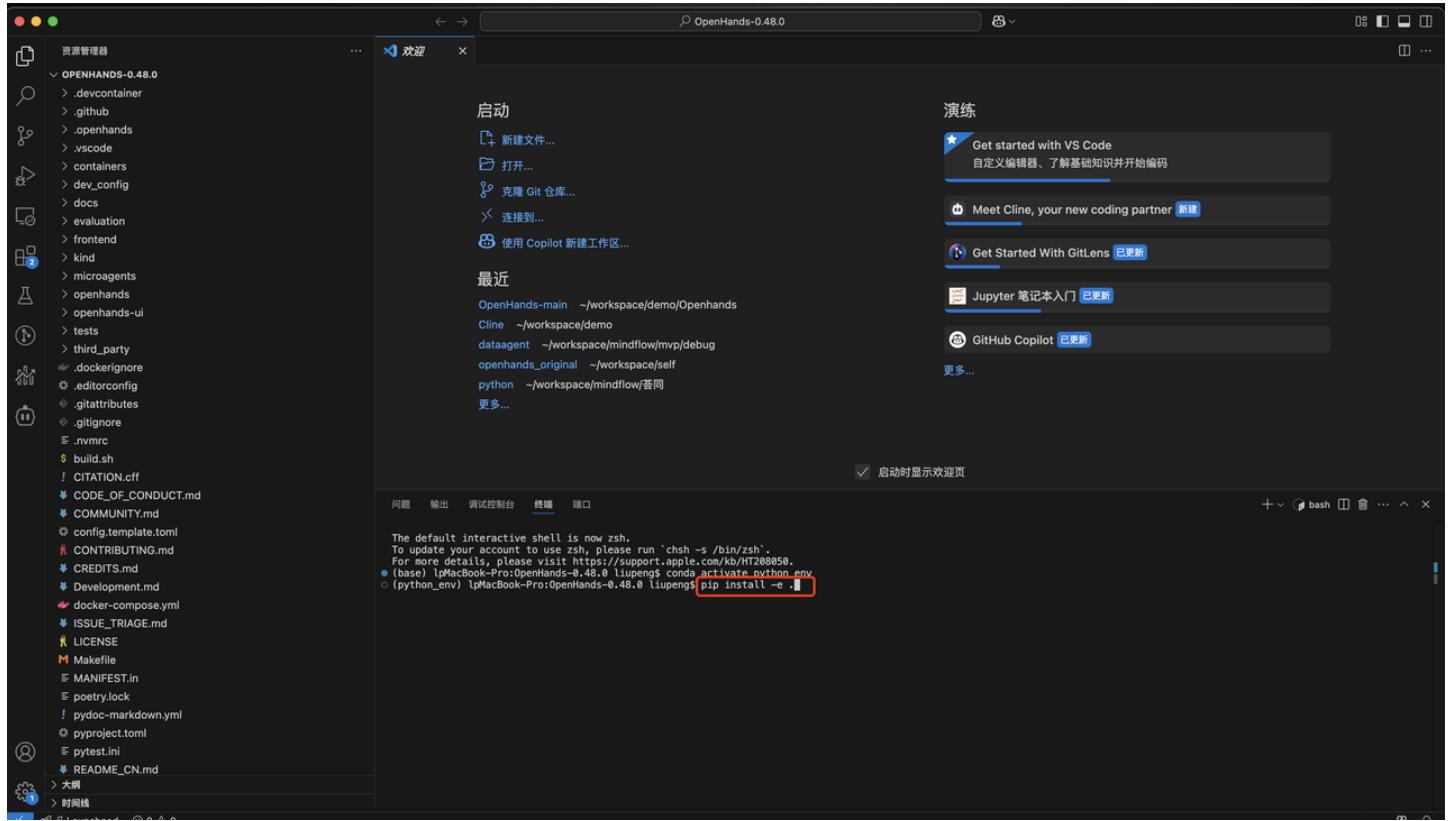


7、终端输入命令安装程序依赖

代码块

```
1 pip install -e .
```

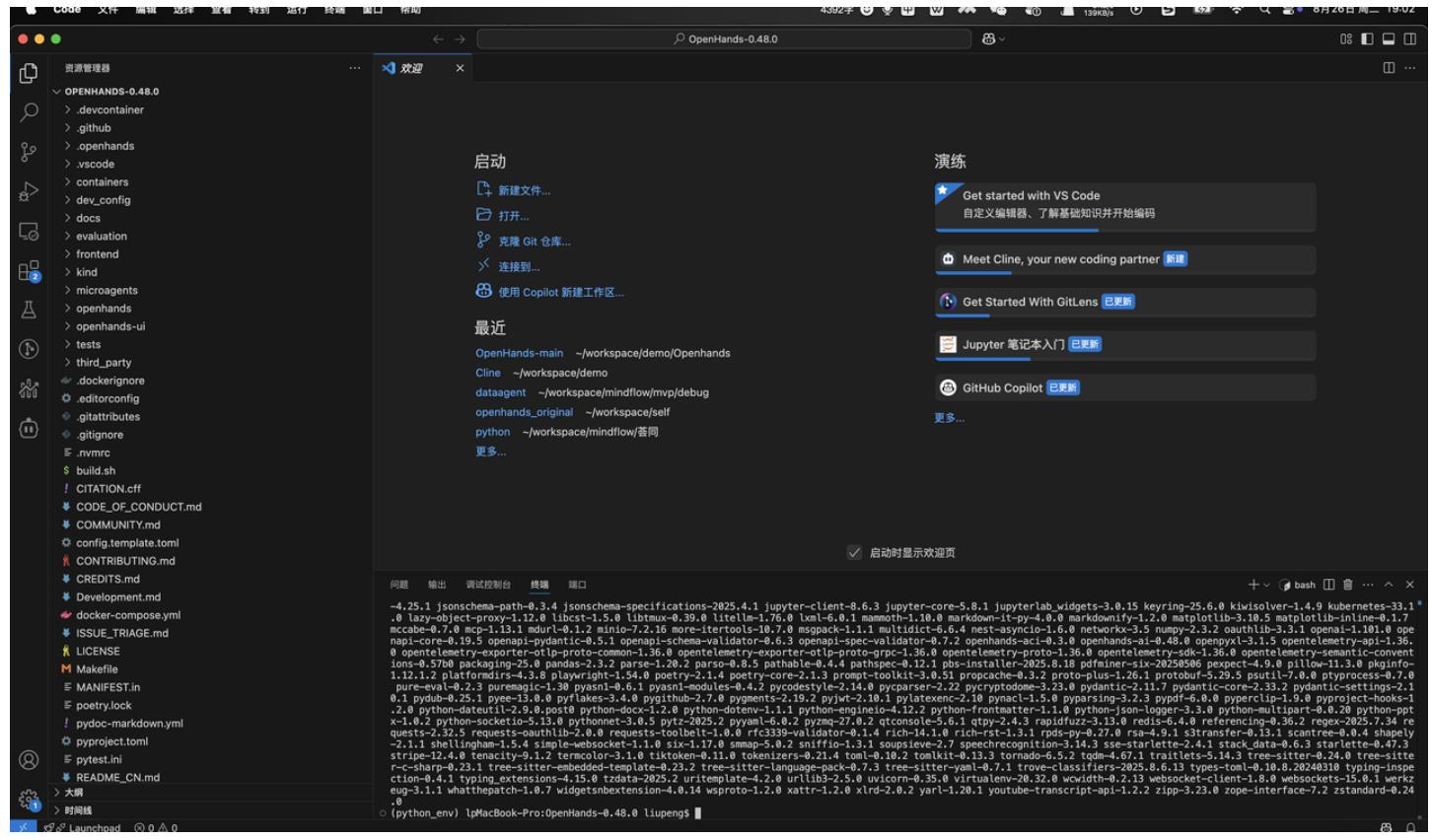
2 pip install Deprecated



This screenshot is identical to the one above, showing the same VS Code interface and terminal output. A red box highlights the terminal command 'pip install -e .' and its subsequent execution and output.

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
● (base) lMacBook-Pro:OpenHands-0.48.0 liupeng$ conda activate python_env  
● (python_env) lMacBook-Pro:OpenHands-0.48.0 liupeng$ pip install -e .  
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple  
Requirement already satisfied: OpenHands in /Users/liupeng/PycharmProjects/OpenHands/OpenHands-0.48.0  
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/3a/1d/c840e611c594330284b1aea8a4b02ca0858fb458614fa35754cab4209c/aiohttp-3.12.15-cp312-cp312-macosx_11_0_arm64.whl (46 kB)  
Collecting anthropic[vertex] (from openhands==0.48.0)  
  Checking if build backend supports build_editable ... done  
  Getting requirements to build editable ... done  
  Preparing editable metadata (pyproject.toml) ... done  
  Collecting PyYAML (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/8e/c86a5643653825d3c913719e788e41386bee415c2b7bf955432f2de6b2/pypdf2-3.0.1-py3-none-any.whl (232 kB)  
  Collecting aiotools<3.11.13,>=3.9.0 (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/3a/1d/c840e611c594330284b1aea8a4b02ca0858fb458614fa35754cab4209c/aiohttp-3.12.15-cp312-cp312-macosx_11_0_arm64.whl (46 kB)  
  Collecting anthropic[vertex] (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/a9/b2/2d268bcd5d6441df9dc0ebec67187657edb8b0150d3fd1a5b81d1bec45/anthropic-0.64.0-py3-none-any.whl (297 kB)  
  Collecting pyyaml (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/a1/ee/48ca17c89ffec8b6a0c5d02b89c305671d5ffd8d3c94ac18b8c488575bb/anyio-4.9.0-py3-none-any.whl (100 kB)  
  Collecting bashlex<0.19,>=0.18 (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/f4/be/6985abb1011dab5a523fce21ed9629e397d6e06fb5bae9750402b25c95b/bashlex-0.18-py2,p3-none-any.whl (69 kB)  
  Collecting boto3 (from openhands==0.48.0)  
    Using cached https://pypi.tuna.tsinghua.edu.cn/packages/91/b5/1973ed8c107beb2fe5a510d56096983537a4e18a92eade021bed5699c5437/boto3-1.40.17-py3-none-any.whl (148 kB)
```

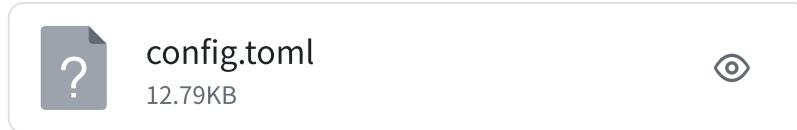
安装完成



配置文件准备

需要将以下配置文件放入到openhands项目根目录下

openhands项目配置文件



启动脚本文件

run.sh

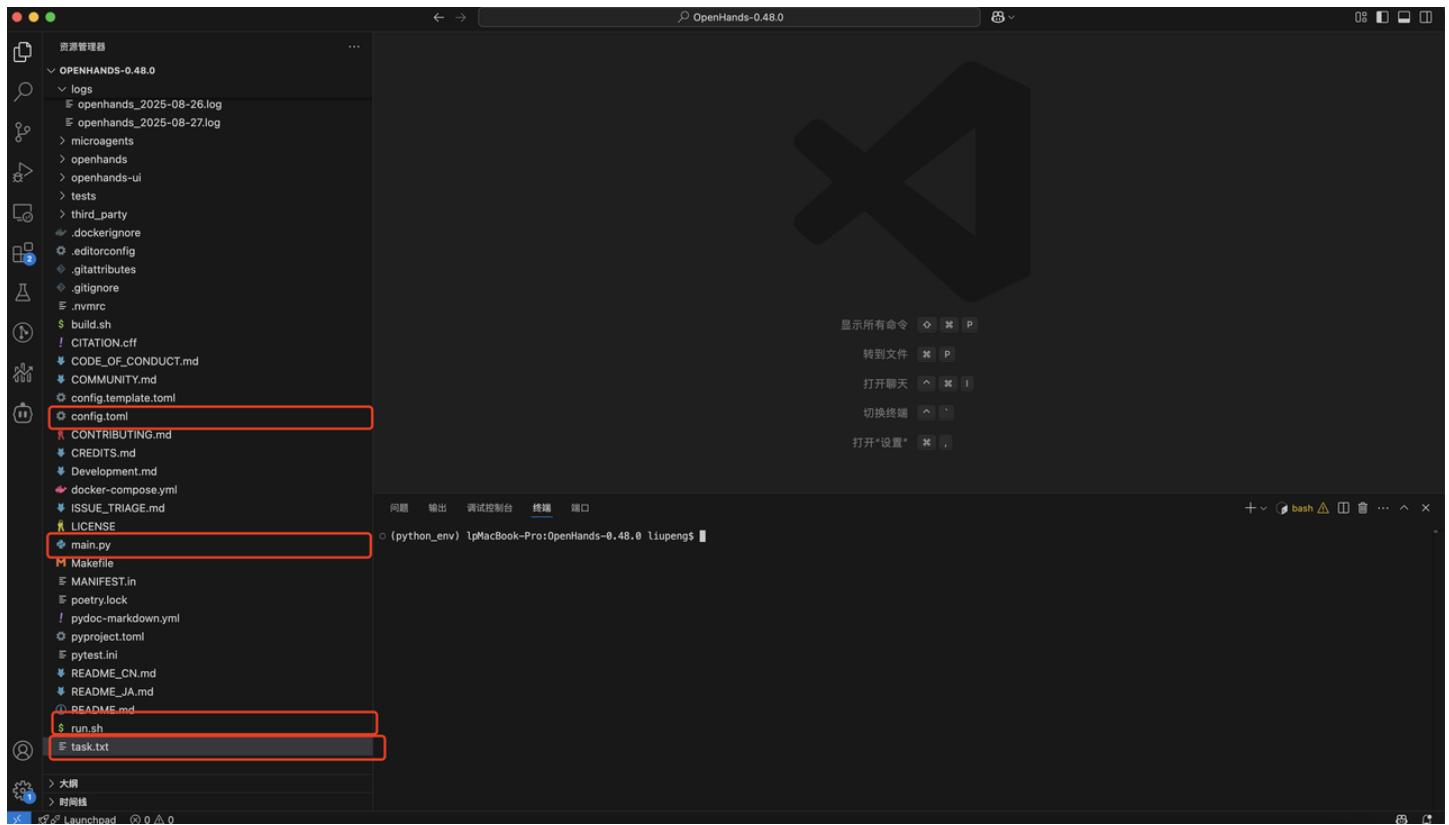
任务描述文件,这里以创建一个hello world的python文件为例

task.txt

入口函数

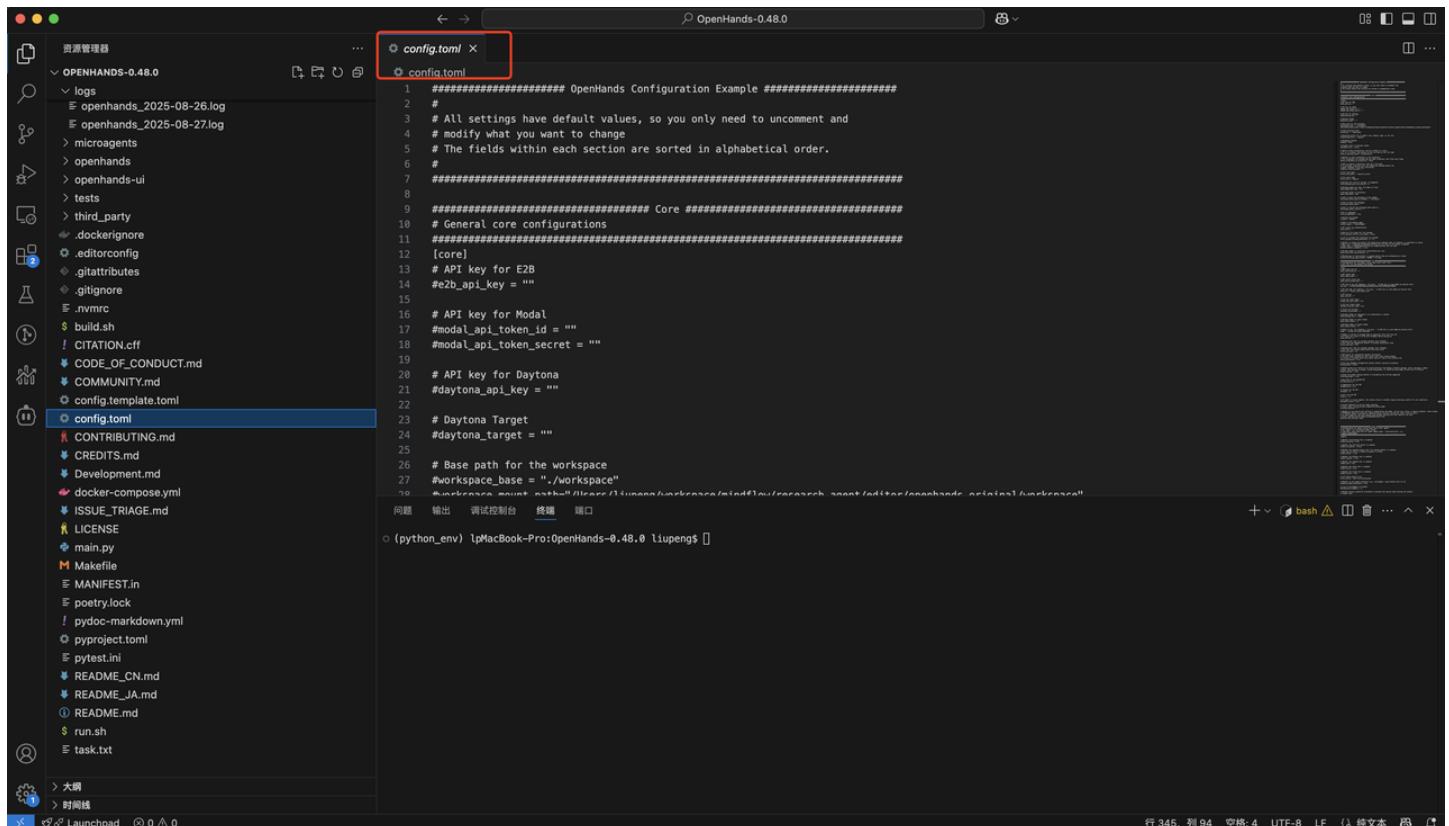
main.py

1、下载配置文件,并将配置文件复制到VSCode中Openhands项目下



2、添加DeepSeek-API Key

双击打开config.toml



找到[llm.deepseek]配置

```
237 [llm.deepseek]
238 model = "deepseek/deepseek-chat"
239 base_url = "https://api.deepseek.com"
240 api_key = "deepseek_key"
```

替换为我们已经申请的deepseek-key

```
238 #llm_config = 'your-lm-config-group'
239
240 # Whether to use prompt extension (e.g., microagent, repo/runtime info) at all
241 #enable_prompt_extensions = true
242
243 # List of microagents to disable
244 #disabled_microagents = []
245
246 # Whether history should be truncated to continue the session when hitting LLM context
247 # length limit
248 enable_history_truncation = true
249
250
251 [llm.deepseek]
252 model = "deepseek/deepseek-chat"
253 base_url = "https://api.deepseek.com"
254 api_key = "sk-e5a32c3fbf54a6c98aa55aa5bca1d89"
```

3、确认切换python环境

终端确认已经切换到openhands所应用的python环境,如果没有执行命令切换

代码块

```
1 conda activate python_env
```

```
config.toml
...
[llm.deepseek]
model = "deepseek/deepseek-chat"
base_url = "https://api.deepseek.com"
api_key = "sk-ee5a32c3fbf54a6c98aa55aa5bca1d88"

[llm.gpt4o-mini]
api_key = ""
model = "gpt-4o"

[agent.CodeActAgent]
llm_config="deepseek"
# [agent.RepoExplorerAgent]
## Example: use a cheaper model for RepoExplorerAgent to reduce cost, especially
## useful when an agent doesn't demand high quality but uses a lot of tokens
```

4.运行脚本文件

代码块

```
1 sh run.sh
```

```
显示所有命令 显示所有命令
转到文件 转到文件
打开聊天 打开聊天
切换终端 切换终端
打开“设置” 打开“设置”

+ - bash 命令 ... ^

(terminal) lMacBook-Pro:OpenHands-0.48.0 liupeng$ which python
/usr/local/miniconda3/envs/python_venv/bin/python
(terminal) lMacBook-Pro:OpenHands-0.48.0 liupeng$ sh run.sh
```

openhands运行启动容器



```
问题 脱机控制台 终端 窗口 + \ bash ▲ □ ☰ ... ×

● (python_env) lpMacBook-Pro:OpenHands-0.48.0 liupeng$ which python
/Users/liupeng/miniconda3/envs/python/bin/python
○ (python_env) lpMacBook-Pro:OpenHands-0.48.0 liupeng$ sh run.sh \
>
○ (python_env) lpMacBook-Pro:OpenHands-0.48.0 liupeng$ sh run.sh
DEBUG:openhands:DEBUG mode enabled.
11:04:33 - openhands:DEBUG: logger.py:375 - Logging initialized
11:04:33 - openhands:DEBUG: config_utils.py:38 - Configuration loaded in: /Users/liupeng/workspace/demo/Openhands/OpenHands-0.48.0/logs
2025-08-27 11:04 - openhands:DEBUG: util.py:257 - Default condenser configuration loaded from config.toml and assigned to default agent
2025-08-27 11:04 - openhands:ac1.editor.file_cache:DEBUG - Current size updated: 0
2025-08-27 11:04 - openhands:ac1.editor.file_cache:DEBUG - FileCache initialized with directory: /var/folders/wb/8cshq152tl_nmfv1r4000gn/T/oh_editor_history_zldqxbs, size limit: None, current_size: 0
11:04:35 - openhands:core/config/utils.py:55 - DeprecationWarning: deprecated field value = getattr(model, name)
11:04:35 - openhands:core/config/utils.py:257 - Default condenser configuration loaded from config.toml and assigned to default agent
11:04:35 - openhands:core/config/utils.py:71 - DeprecationWarning: deprecated field value = getattr(sub_config, field_name)
11:04:35 - openhands:core/config/utils.py:323 - DeprecationWarning: deprecated if cfg.workspace_base is not None or cfg.workspace_mount_path is not None:
11:04:35 - openhands:core/config/utils.py:370 - DeprecationWarning: deprecated if cfg.workspace_base is not None or cfg.workspace_mount_path is not None:
11:04:35 - openhands:DEBUG: util.py:174 - Model info: {
  "model": "deepleepick/deepleepick-chat",
  "base_url": "https://api.deepleepick.com"
}
11:04:35 - openhands:DEBUG: codeact_agent.py:94 - Using condenser: <class 'openhands.memory.condenser.impl.no_op_condenser.NoOpCondenser'>
11:04:35 - openhands:DEBUG: event_store.py:63 - No events found for session bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1 at sessions/bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1
11:04:35 - openhands:DEBUG: util.py:167
11:04:35 - openhands:DEBUG: setup.py:72 - Initializing runtime: DockerRuntime
11:04:35 - openhands:DEBUG: shutdown_listener.py:48 - register_signal_handlers
11:04:35 - openhands:DEBUG: shutdown_listener.py:56 - register_signal_handlers:not_main_thread
11:04:35 - openhands:DEBUG: setup.py:81 - Runtime created with plugins: ['agent_skills', 'jupyter']
11:04:35 - openhands:INFO: docker_runtime.py:165 - [runtime bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fda6ebea8cc3e62] Starting runtime with image: ghr.io/all-hands-ai/runtime:0.48-nikotalk
11:04:35 - openhands:DEBUG: docker_runtime.py:275 - [runtime bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fda6ebea8cc3e62] Preparing to start container...
```

5、查看容器

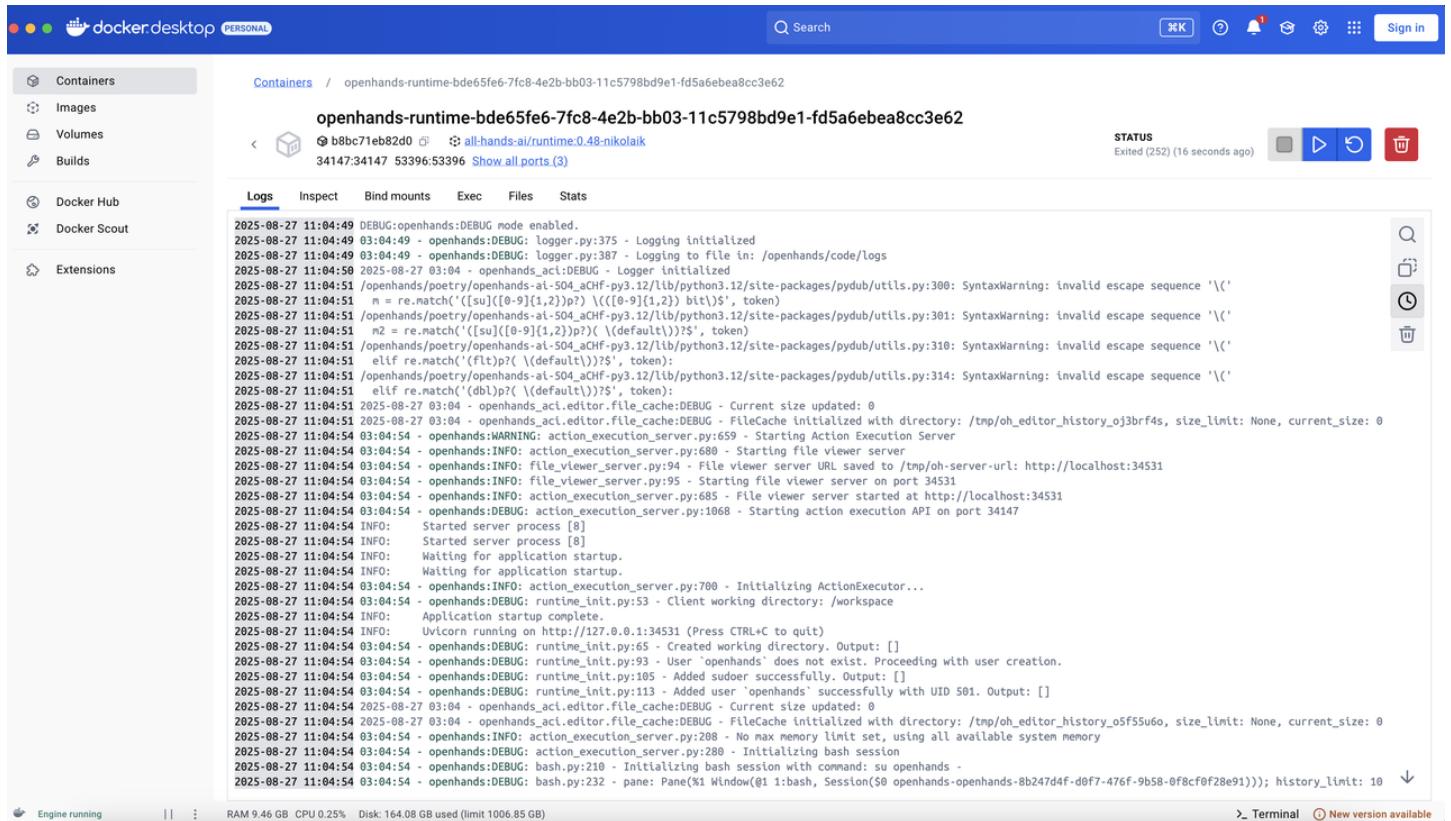
打开docker desktop可以查看到已经启动了一个openhands容器,点击打开

The screenshot shows the Docker Desktop application window. The left sidebar has 'Containers' selected. The main area displays container statistics and a list of running containers:

- Container CPU usage:** 76.97% / 800% (8 CPUs available)
- Container memory usage:** 1.82GB / 22.89GB
- Show charts** button
- Search bar** and **Only show running containers** checkbox
- Table of running containers:**

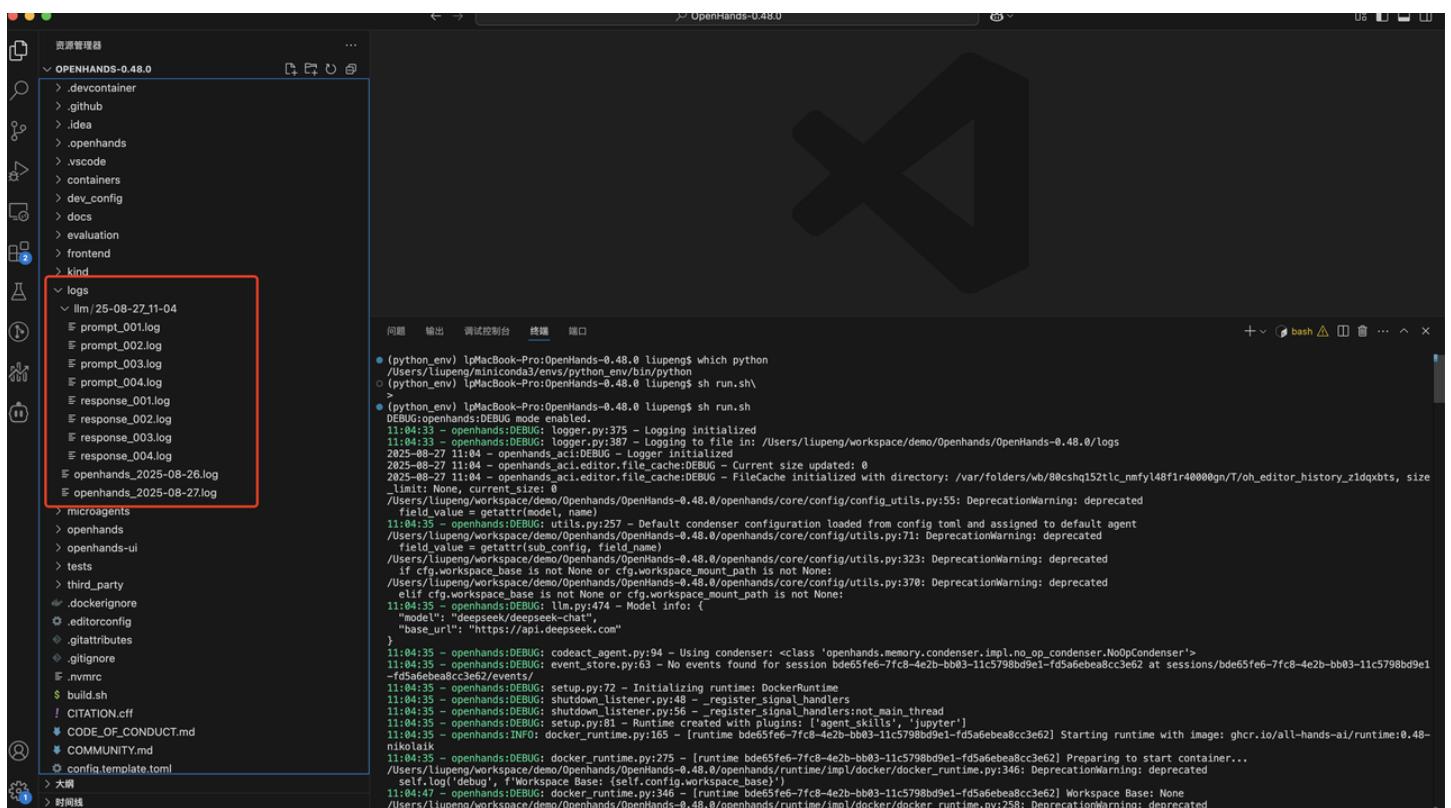
Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
my-nginx	29d11fbf3913	nginx:latest	80:80	0%	5 months ago	[More]
mysql8	11e9fdab1bb2	mysql:8.0	3306:3306	0.32%	6 days ago	[More]
openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6eb-b8bc71eb82d0	all-hands-ai/runtime	34147:34147	Show all ports (3)	76.65%	16 seconds ago	[More]
mindflow	-	-	-	0%	2 days ago	[More]

容器内已经在启动程序



6、查看大模型对话

查看openhands目录下logs/lm文件夹,该文件夹是agent的对话信息,prompt和response是成对生成



7、查看运行结果

终端输出已经完成创建python代码文件并输出hello_world

8、验证结果

查看任务容器已经结束,点击运行重新启动容器

点击exec进入命令窗口

```

Containers / openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62
openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62
< b8bc71eb82d0 ⚡ ⚡ all-hands-ai/runtime:0.48-nikolaik
34147:34147 ⚡ 53396:53396 ⚡ Show all ports (3)

STATUS
Running (0 seconds ago) [Stop] [Start] [Restart] [Delete]

Logs Exec Bind mounts Files Stats

2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: bash.py:232 - pane: Pane(%1 Window(@1 bash, Session($0 openhands-openhands-2a70e419-c9b2-45bf-bac0-9cea7049f86d))); history_limit: 10 000
2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: bash.py:246 - Bash session initialized with work dir: /workspace
2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: action_execution_server.py:282 - Bash session initialized
2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: action_execution_server.py:286 - Browser initialization started in background
2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: action_execution_server.py:226 - Initializing browser asynchronously
2025-08-27 11:17:58 03:17:58 - openhands:DEBUG: browser_env.py:57 - Starting browser env...
2025-08-27 11:17:59 03:17:59 - openhands:DEBUG: DEBUG mode enabled.
2025-08-27 11:17:59 03:17:59 - openhands:DEBUG: logger.py:375 - Logging initialized
2025-08-27 11:17:59 03:17:59 - openhands:DEBUG: logger.py:387 - Logging to file in: /openhands/code/logs
2025-08-27 11:17:59 2025-08-27 03:17 - openhands_aci:DEBUG - Logger initialized
2025-08-27 11:18:00 2025-08-27 03:18 - openhands_aci.editor.file_cache:DEBUG - Current size updated: 0
2025-08-27 11:18:00 2025-08-27 03:18 - openhands_aci.editor.file_cache:DEBUG - FileCache initialized with directory: /tmp/oh_editor_history_epxo0fv3, size_limit: None, current_size: 0
2025-08-27 11:18:02 03:18:02 - openhands:INFO: browser_env.py:109 - Successfully called env.reset
2025-08-27 11:18:02 03:18:02 - openhands:INFO: browser_env.py:121 - Browser env started.
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: shutdown_listener.py:48 - _register_signal_handlers
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: shutdown_listener.py:52 - _register_signal_handlers:main_thread
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: action_execution_server.py:229 - Browser initialized asynchronously
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: action_execution_server.py:320 - Initializing plugin: agent_skills
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: __lnt__.py:113 - Jupyter launch command: su - openhands -s /bin/bash << 'EOF'
2025-08-27 11:18:02 cd /openhands/code
2025-08-27 11:18:02 export POETRY_VIRTUALENVS_PATH=/openhands/poetry;
2025-08-27 11:18:02 export PYTHONPATH=/openhands/code:$PYTHONPATH;
2025-08-27 11:18:02 export MAMBA_ROOT_PREFIX=/openhands/micromamba;
2025-08-27 11:18:02 /openhands/micromamba/bin/micromamba run -n openhands poetry run jupyter kernelgateway --KernelGatewayApp.ip=0.0.0.0 --KernelGatewayApp.port=49606
2025-08-27 11:18:02 EOF
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: shutdown_listener.py:48 - _register_signal_handlers
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: shutdown_listener.py:52 - _register_signal_handlers:main_thread
2025-08-27 11:18:02 03:18:02 - openhands:DEBUG: __init__.py:133 - Jupyter Kernel gateway started at port 49606. Output: [I 2025-08-27 03:18:02.793 KernelGatewayApp] Jupyter Kernel Gateway 3.0.1 is available at http://0.0.0.0:49606
2025-08-27 11:18:02
2025-08-27 11:18:03 03:18:03 - openhands:DEBUG: action_execution_server.py:320 - Initializing plugin: jupyter
2025-08-27 11:18:03 03:18:03 - openhands:DEBUG: action_execution_server.py:412 - /workspace != None -> reset Jupyter PWD
2025-08-27 11:18:03 03:18:03 - openhands:DEBUG: action_execution_server.py:422 - Changed working directory in IPython to: /workspace. Output: **IPythonRunCellObservation**
2025-08-27 11:18:03 [Code executed successfully with no output]
2025-08-27 11:18:03 03:18:03 - openhands:DEBUG: action_execution_server.py:292 - All plugins initialized
2025-08-27 11:18:03 03:18:03 - openhands:DEBUG: action_execution_server.py:297 - Initializing AgentSkills

```

Engine running | RAM 9.72 GB CPU 9.45% Disk: 164.09 GB used (limit 1006.85 GB) | Terminal | New version available

输入以下命令并执行

代码块

1 bash

```

Containers / openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62
openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62
< b8bc71eb82d0 ⚡ ⚡ all-hands-ai/runtime:0.48-nikolaik
34147:34147 ⚡ 53396:53396 ⚡ Show all ports (3)

STATUS
Running (29 seconds ago) [Stop] [Start] [Restart] [Delete]

Logs Exec Bind mounts Files Stats

Docker Debug brings the tools you need to debug your container with one click.
Requires a paid Docker subscription. Learn more.

# bash
root@b8bc71eb82d0:/openhands/code# 

Debug mode Open in external terminal ⓘ

Sign in ×

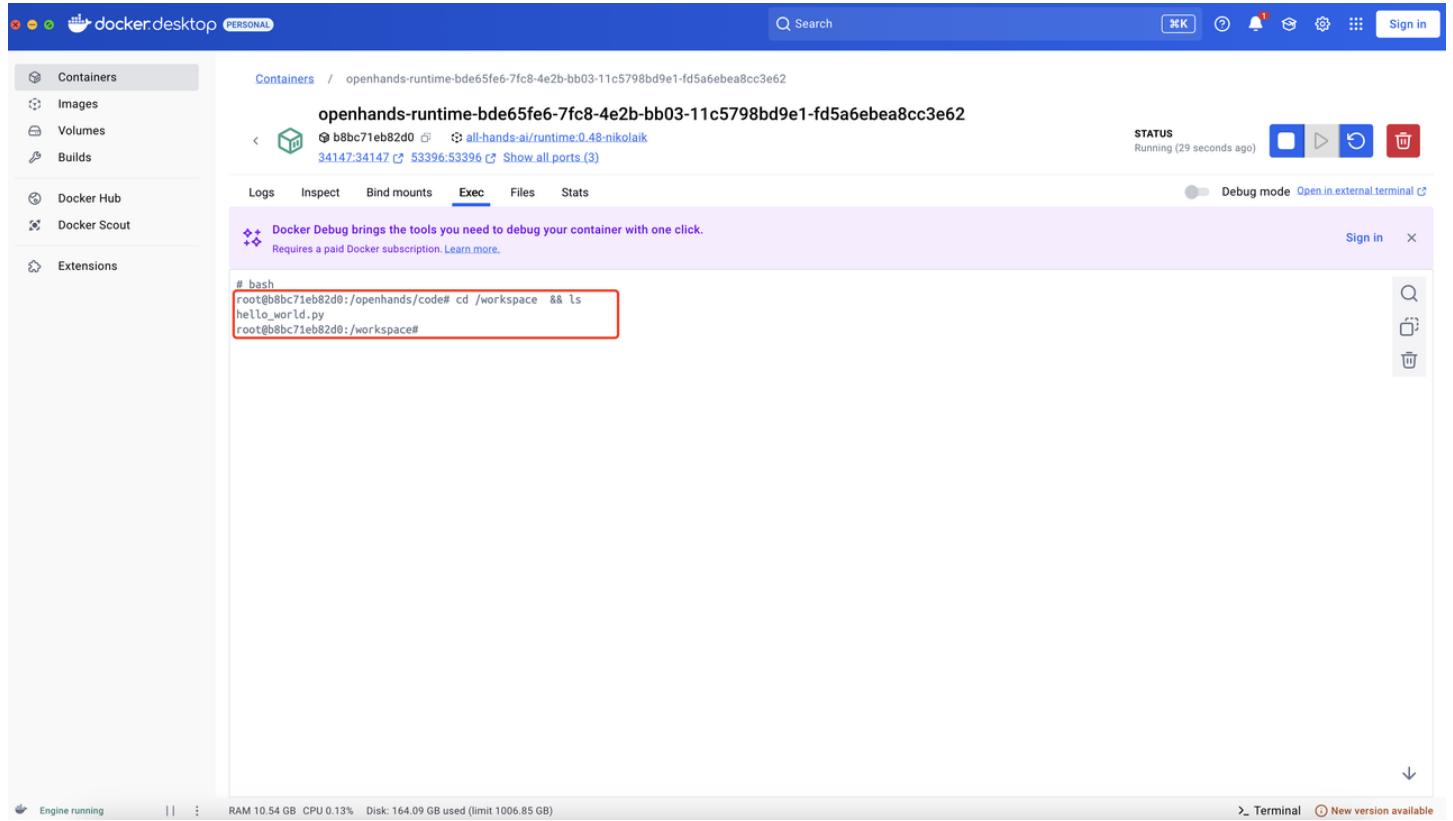
# bash
root@b8bc71eb82d0:/openhands/code# 


```

Engine running | RAM 10.52 GB CPU 0.38% Disk: 164.09 GB used (limit 1006.85 GB) | Terminal | New version available

执行命令查看是否有生成文件

代码块 cd /workspace && ls



查看文件内容

代码块

```
1 cat hello_world.py
```

Containers / openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62

b8bc71eb82d0 ⚡ all-hands-ai/runtime:0.48-nikolaik
34147:34147 53396:53396 Show all ports (3)

STATUS
Running (29 seconds ago)

Logs Inspect Bind mounts Exec Files Stats

Docker Debug brings the tools you need to debug your container with one click.
Requires a paid Docker subscription. Learn more.

bash
root@b8bc71eb82d0:/openhands/code# cd /workspace && ls
hello_world.py
root@b8bc71eb82d0:/workspace# cat hello_world.py
#!/usr/bin/env python3
This is a simple Python script that outputs Hello World

print("Hello World")
root@b8bc71eb82d0:/workspace#

helloworld代码已创建,运行校验

Containers / openhands-runtime-bde65fe6-7fc8-4e2b-bb03-11c5798bd9e1-fd5a6ebea8cc3e62

b8bc71eb82d0 ⚡ all-hands-ai/runtime:0.48-nikolaik
34147:34147 53396:53396 Show all ports (3)

STATUS
Running (29 seconds ago)

Logs Inspect Bind mounts Exec Files Stats

Docker Debug brings the tools you need to debug your container with one click.
Requires a paid Docker subscription. Learn more.

bash
root@b8bc71eb82d0:/openhands/code# cd /workspace && ls
hello_world.py
root@b8bc71eb82d0:/workspace# cat hello_world.py
#!/usr/bin/env python3
This is a simple Python script that outputs Hello World

print("Hello World")
root@b8bc71eb82d0:/workspace# python hello_world.py
Hello World
root@b8bc71eb82d0:/workspace#

正确输出hello world 验证完成

openhands配置和实验跑通

以一个RegMean++这个sota工作为例跑通实验

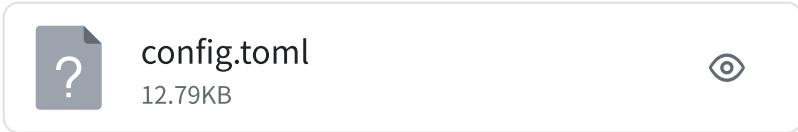
Sota项目地址:<https://github.com/nthehai01/RegMean-plusplus>

1. 添加实验的任务描述

experiment-task.txt
5.01KB

```
1 You are a senior machine learning engineer tasked with
2 Your goal is to ensure that the project can process the
3 # Instructions
4
5 - Start by reading `data_description.md` in Dataset to
6 - Review the project documentation (e.g., `README.md`)
7 - You are working in a pre-configured conda environment
8 - Your final output should follow the `Submission Re
9 - You must obtain valid evaluation metric values (i.e.
10   - If any of the evaluation metrics is NaN, it indic
11   - You must identify the issue and rerun the process
12
13 ## Two Stage Adaption
14 You should run the fast training and evaluation in Stage
15
16 - Stage 1: Quick Validation
17 Configure the training with a small number of epochs (e
18 The goal is to verify that the code works correctly wi
19
20 - Stage 2: Full Training
21 Once the code is validated, switch to the normal or re
22
23 This two-stage approach ensures efficient debugging and
24
25
26 ## Constraints:
27 - **Do not modify the core algorithm or neural network**
28 - Only adjust the dataset processing and submission ge
29 - You may change the model's input/output layer if nee
30
31
32 # Resource Locations
33
34 1. Task:
35   - Description: `/workspace/dataset/data_description.
36
37 2. Dataset:
38   - Data directory: `/workspace/dataset/`
39   - Description: `/workspace/dataset/data_description.
40
41 3. Project Code:
42   - Location: `/workspace/{project}`
43   - Environment: conda environment `{conda_env}`
44   - python: `/opt/conda/envs/{conda_env}/bin/python`
```

2. openhands配置文件修改

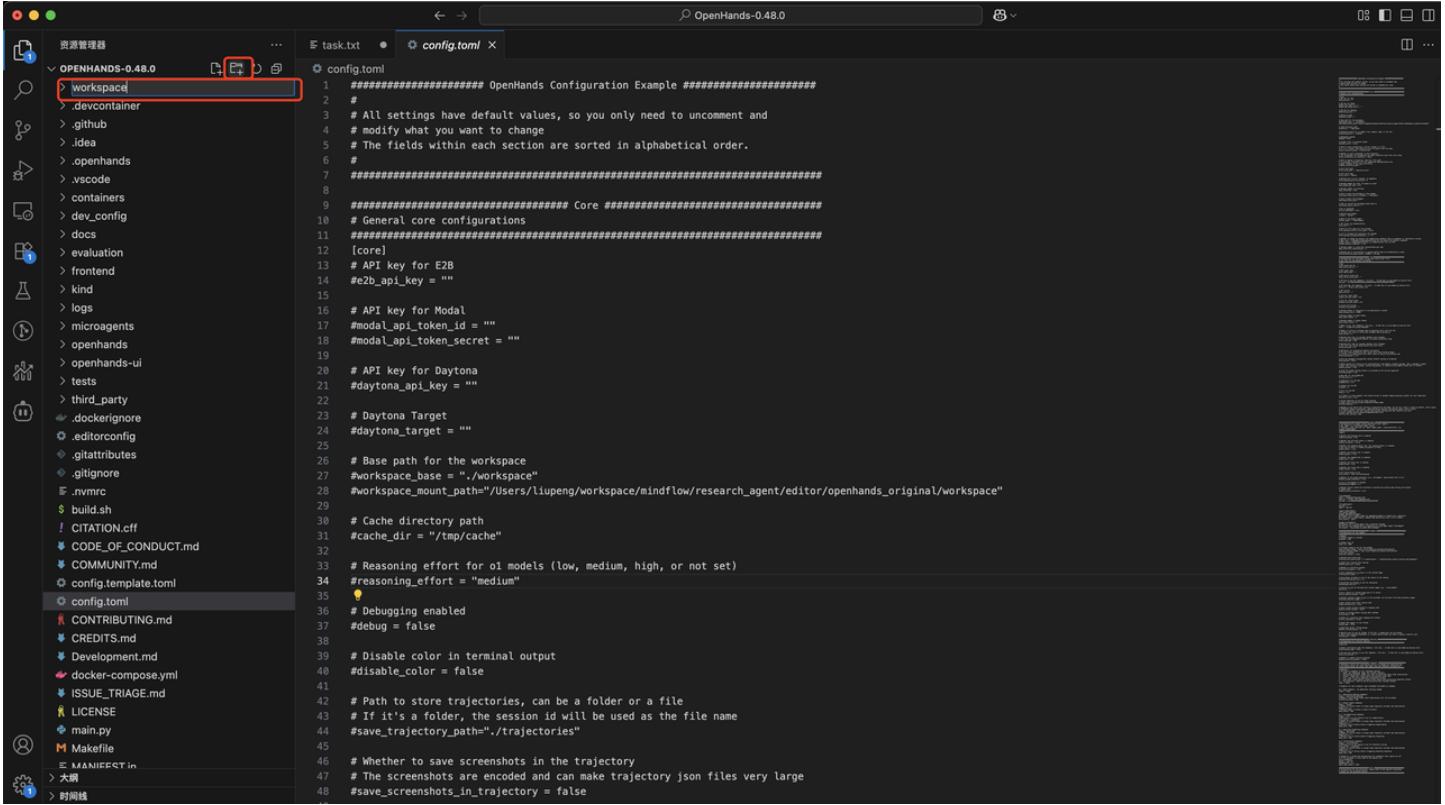


需要修改config.toml配置文件

代码块

```
1 # Base path for the workspace
2 workspace_base = "./workspace"
3 #修改为需要挂载到容器内的workspace目录。默认直接在openhands项目下创建一个workspace文件夹,将其挂载到容器内
4 workspace_mount_path={本地项目文件(workspace}
```

创建workspace文件夹在项目目录下



在终端中执行pwd,查看当前项目路径

代码块

```
1 pwd
```

```
config.toml
1 ##### OpenHands Configuration Example #####
2 #
3 # All settings have default values, so you only need to uncomment and
4 # modify what you want to change
5 # The fields within each section are sorted in alphabetical order.
6 #
7 #####
8 #
9 ##### Core #####
10 # General core configurations
11 #####
12 [core]
13 # API key for E2B
14 #e2b_api_key = ""
15
16 # API key for Modal
17 #modal_api_token_id = ""
18 #modal_api_token_secret = ""
19
20 # API key for Dayton
21 #daytona_api_key = ""
22
23 # Dayton Target
24 #daytona_target = ""
25
26 # Base path for the workspace
27 #workspace_base = "./workspace"
28 #workspace_mount_path="/Users/liupeng/workspace/mindflow/research_agent/editor/openhands_original/workspace"
29
30 # Cache directory path
```

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.

找到需要修改的config.toml配置文件中挂载地址配置项

```
config.toml
1 #
2 # All settings have default values, so you only need to uncomment and
3 # modify what you want to change
4 # The fields within each section are sorted in alphabetical order.
5 #
6 #####
7 #####
8 #####
9 ##### Core #####
10 # General core configurations
11 #####
12 [core]
13 # API key for E2B
14 #e2b_api_key = ""
15
16 # API key for Modal
17 #modal_api_token_id = ""
18 #modal_api_token_secret = ""
19
20 # API key for Dayton
21 #daytona_api_key = ""
22
23 # Dayton Target
24 #daytona_target = ""
25
26 # Base path for the workspace
27 #workspace_base = "./workspace"
28 #workspace_mount_path="/Users/liupeng/workspace/mindflow/research_agent/editor/openhands_original/workspace"
29
30 # Cache directory path
```

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.

去掉"#"符号放开注释使配置生效

```
config.toml
...
# Base path for the workspace
workspace_base = "./workspace"
workspace_mount_path="/Users/liupeng/workspace/mindflow/research_agent/editor/openhands_original/workspace"

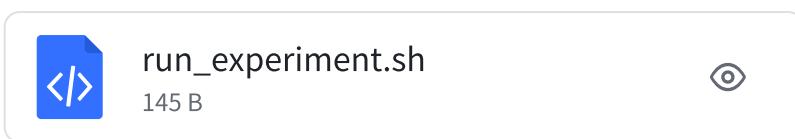
# Cache directory path
#cache_dir = "/tmp/cache"
```

复制pwd命令显示的路径来替换workspace_mount_path配置项成为
workspace_mount_path="{pwd_path}/workspace"

```
config.toml
...
# Base path for the workspace
workspace_base = "./workspace"
workspace_mount_path="/Users/liupeng/workspace/demo/Openhands-0.48.0/workspace"

# Cache directory path
#cache_dir = "/tmp/cache"
```

3. 创建启动脚本文件



The screenshot shows a terminal window with two panes. The left pane displays a file tree for the directory `OpenHands-0.48.0`. The right pane shows a shell script with line numbers.

File Tree (Left Pane):

- OpenHands-0.48.0 (~/Downloads/OpenHands-0.48.0)
 - .devcontainer
 - .github
 - .openhands
 - .vscode
 - containers
 - dev_config
 - docs
 - evaluation
 - frontend
 - kind
 - logs
 - microagents
 - openhands
 - openhands-ui
 - tests
 - third_party
 - workspace
 - .dockerignore
 - .editorconfig
 - .gitattributes
 - .gitignore
 - .nvmrc
 - build.sh
 - CITATION.cff
 - CODE_OF_CONDUCT.md
 - COMMUNITY.md
 - config.template.toml
 - config.toml
 - CONTRIBUTING.md
 - CREDITS.md
 - Development.md
 - docker-compose.yml
 - experiment-task.txt
 - ISSUE_TRIAGE.md
 - LICENSE
 - Makefile
 - MANIFEST.in
 - poetry.lock
 - pydoc-markdown.yml
 - pyproject.toml
 - pytest.ini
 - README.md
 - README_CN.md
 - README_JA.md
 - run.sh
 - run_experiment.sh
 - start.py
 - task.txt
 - External Libraries
 - Scratches and Consoles

Shell Script (Right Pane):

```
1 > #!/bin/bash
2 export LOG_ALL_EVENTS="1"
3 export DEBUG="1"
4 export LOG_TO_FILE="1"
5 export no_proxy="127.0.0.1,127.0.0.0/8,localhost"
6
7 python start.py
8
```

4. 把准备好的benmark（数据集和测评脚本）复制到workspace文件夹目录

```

task.txt
8 - You are working in a pre-configured conda environment named 'DLlinear'.
9 - Your final output should follow the 'Submission Requirements' document.
10 - You must obtain valid evaluation metric values (i.e., they must not be NaN):
11 | - If any of the evaluation metrics is NaN, it indicates the task has not been successfully completed.
12 | - You must identify the issue and rerun the process until all evaluation metrics return valid (non-NaN) values.
13
14 ## Two Stage Adaption
15 You should run the fast training and evaluation in Stage 1 first, then perform full training in Stage 2, and report the final evaluation results of Stage 2.
16
17 - Stage 1: Quick Validation
18 Configure the training with a small number of epochs (e.g., epoch=1) to quickly run the code and obtain preliminary results.
19 The goal is to verify that the code works correctly with minimal time investment.
20
21 - Stage 2: Full Training
22 Once the code is validated, switch to the normal or recommended number of epochs and perform full training to achieve optimal model performance.
23
24 This two-stage approach ensures efficient debugging and effective model training.
25
26
27 ## Constraints:
28 - **Do not modify the core algorithm or neural network structure.**
29 - Only adjust the dataset processing and submission generation part.
30 - You may change the model's input/output layer if needed, but do **not** alter the core neural network structure.
31
32 # Resource Locations
33
34 1. Task:
35 | - Description: '/workspace/dataset/transformer-data/data_description.md'
36
37 2. Dataset:
38 | - Data directory: '/workspace/dataset/'
39 | - Description: '/workspace/dataset/transformer-data/data_description.md'
40
41

```

问题 输出 调试控制台 终端 端口

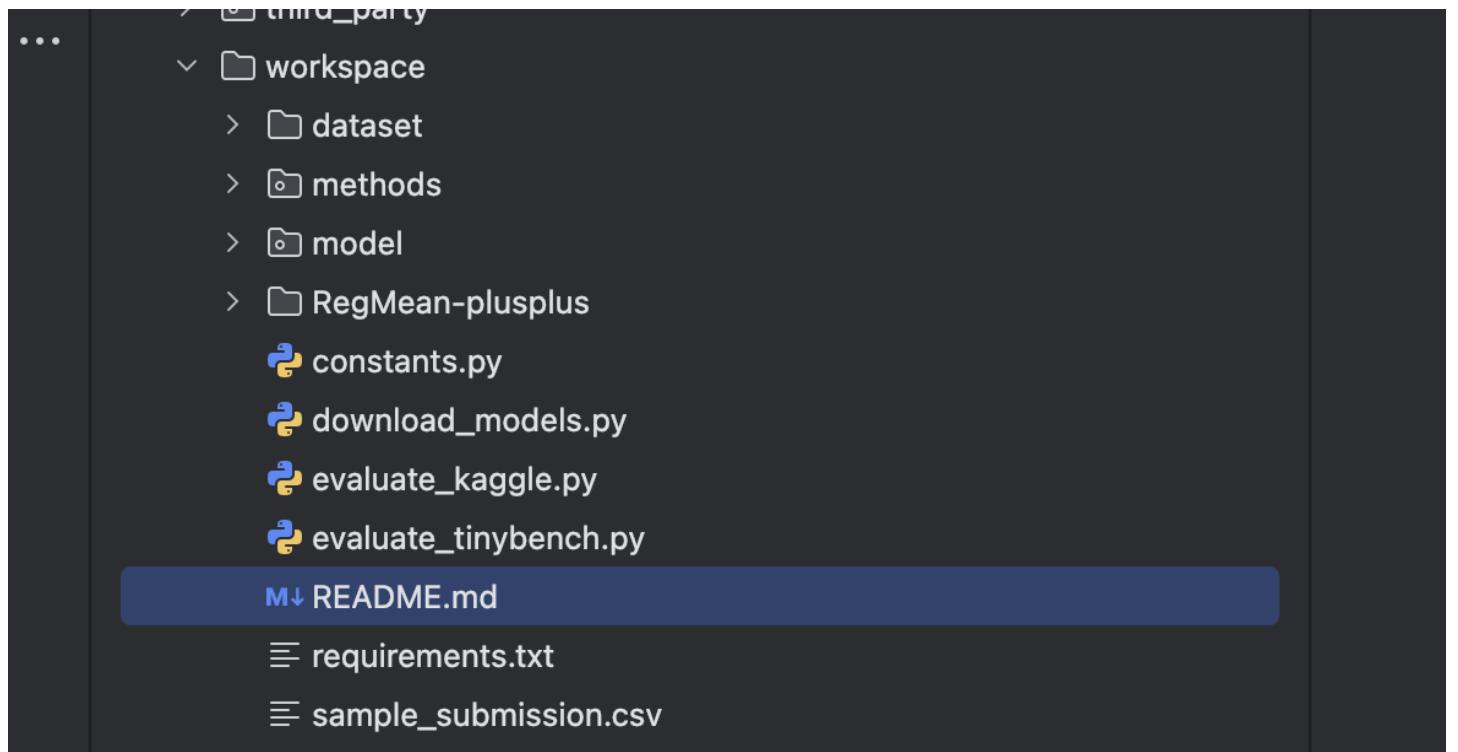
The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit: https://support.apple.com/kb/HT208050.

```

(base) lpMacBook-Pro:OpenHands-0.48.0 lipeng$ pwd
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0
(base) lpMacBook-Pro:OpenHands-0.48.0 lipeng$ []

```

5. 将Sota项目代码放到workspace文件夹下



6. 创建启动openhands的python代码文件start.py

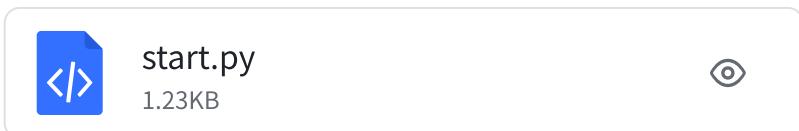
OpenHands-0.48.0 ~/Downloads/OpenHands-0.48.0

```

1 import asyncio
2 from pathlib import Path
3
4 from openhands.core.config import load_openhands_config, OpenHandsConfig
5 from openhands.core.main import run_controller
6 from openhands.core.setup import generate_sid
7 from openhands.events.action import MessageAction
8
9
10 def do_openhands_original(): 1 usage
11     config: OpenHandsConfig = load_openhands_config(config_file='./config.toml')
12     task_file = './task.txt'
13     template = Path(task_file).read_text(
14         encoding='utf-8')
15     prompt = template.format(
16         conda_env='填写具体python执行的环境名称，可以是任意的但必须是小写',
17         project='填写具体的sota工程代码对应的目录名称，如 (RegMean-plusplus) ',
18         evaluation_bash='填写你的测试脚本（根据具体的benchmark来设置）'
19     )
20     task_str = prompt
21
22     # Create actual initial user action
23     initial_user_action = MessageAction(content=task_str)
24
25     # Set session name
26     session_name = ''
27     sid = generate_sid(config, session_name)
28
29     asyncio.run(
30         run_controller(
31             config=config,
32             initial_user_action=initial_user_action,
33             sid=sid,
34             fake_user_response_fn=None
35         )
36     )
37
38 if __name__ == '__main__':
39     do_openhands_original()
40

```

start.py



对以下图中的代码进行修改（修改为对应需要运行的sota工作的名字）

```
def do_openhands_original(): 1 usage
    config: OpenHandsConfig = load_openhands_config(config_file="./config.toml")
    task_file = './experiment-task.txt'
    template = Path(task_file).read_text(
        encoding='utf-8')
    prompt = template.format(
        conda_env='填写具体python执行的环境名称，可以是任意的但必须是小写',
        project='填写具体的sota工程代码对应的目录名称，如 (RegMean-plusplus) ',
        evaluation_bash='填写你的测试脚本（根据具体的benchmark来设置）'
    )
    task_str = prompt

    # Create actual initial user action
    initial_user_action = MessageAction(content=task_str)
    # Set session name
    session_name = ''
    sid = generate_sid(config, session_name)

    asyncio.run(
        run_controller(
            config=config,
            initial_user_action=initial_user_action,
            sid=sid,
            fake_user_response_fn=None
        )
    )

if __name__ == '__main__':
    do_openhands_original()
```

修改后如下（注意此处的evaluation_bash是你准备好的benchmark中的测评脚本的启动命令）：

```
def do_openhands_original(): 1 usage
    config: OpenHandsConfig = load_openhands_config(config_file="../config.toml")
    task_file = './experiment-task.txt'
    template = Path(task_file).read_text(
        encoding='utf-8')
    prompt = template.format(
        conda_env='regmean',
        project='RegMean-plusplus',
        evaluation_bash='python evaluate_kaggle.py --submission_path submission.csv --kaggle_username ycp11111 --')
    task_str = prompt

    # Create actual initial user action
    initial_user_action = MessageAction(content=task_str)
    # Set session name
    session_name = ''
    sid = generate_sid(config, session_name)

    asyncio.run(
        run_controller(
            config=config,
            initial_user_action=initial_user_action,
            sid=sid,
            fake_user_response_fn=None
        )
    )

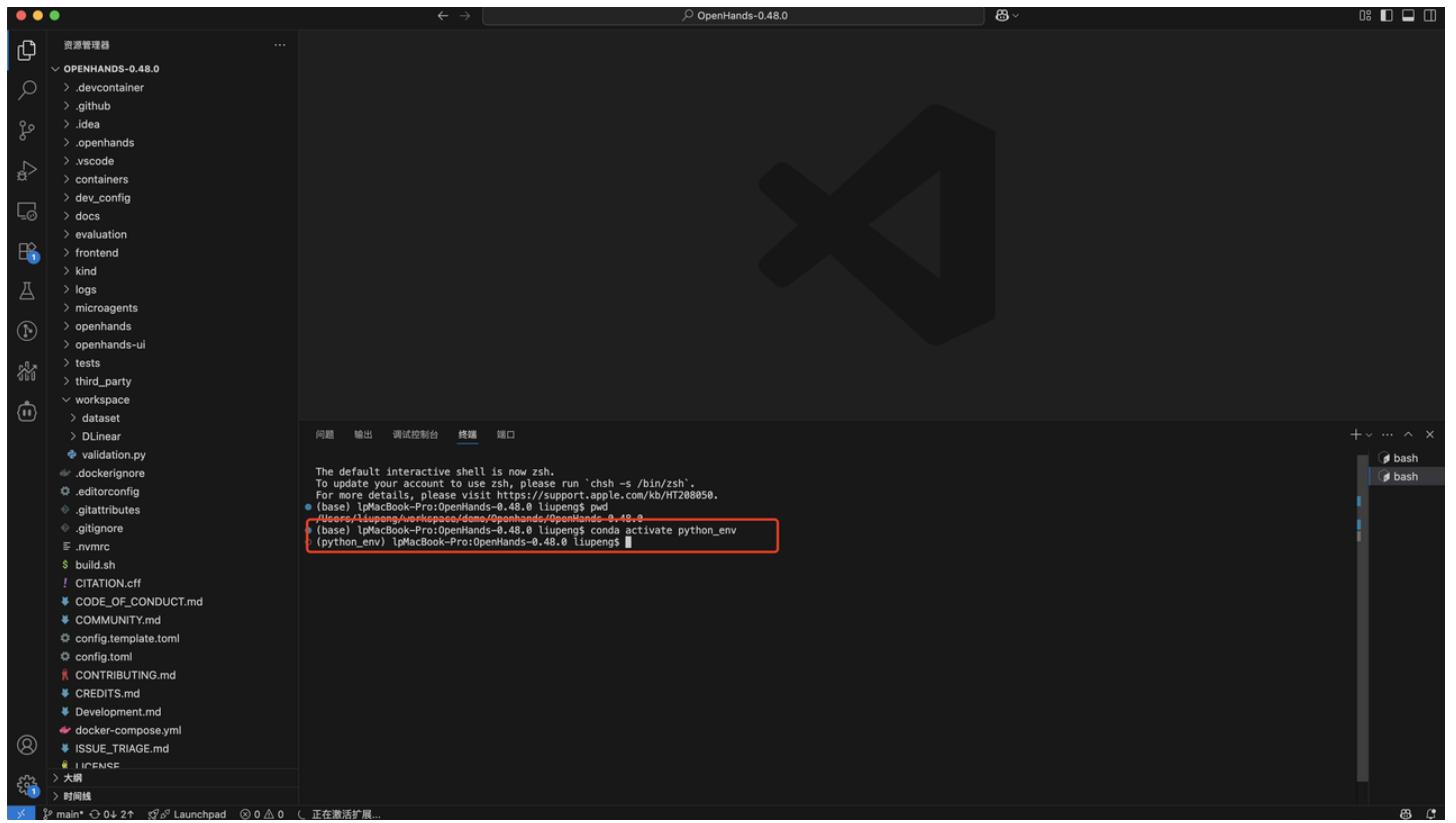
if __name__ == '__main__':
    do_openhands_original()
```

7. 运行项目

切换到python_env环境

代码块

```
1  conda activate python_env
```

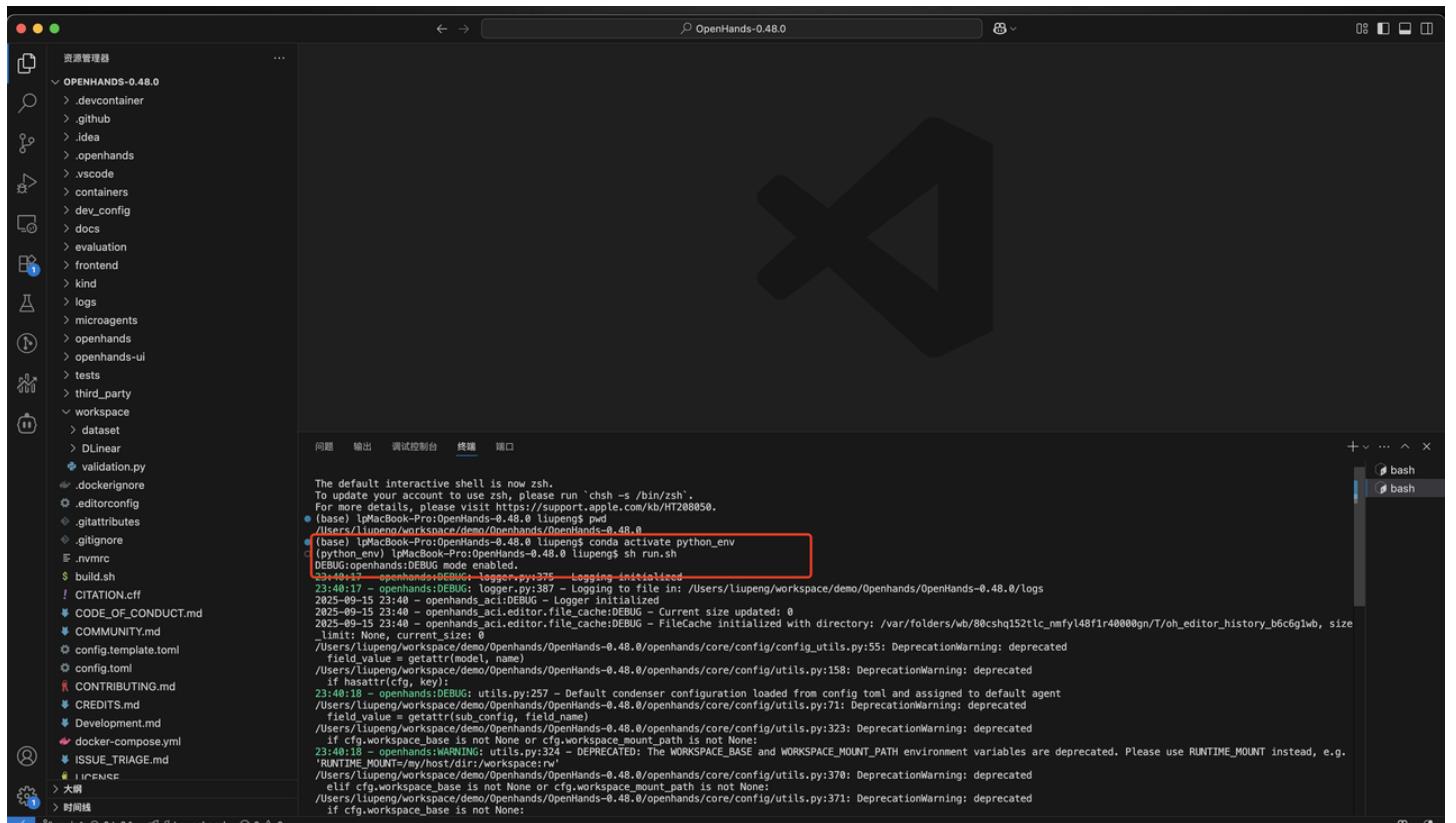


The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT288058.
● (base) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$ pwd
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0
● (base) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$ conda activate python_env
(python_env) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$

运行启动脚本

代码块

```
1 sh run_experiment.sh
```



The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT288058.
● (base) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$ pwd
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0
● (base) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$ conda activate python_env
(python_env) lpMacBook-Pro:OpenHands-0.48.0 lipeng\$ sh run.sh
DEBUG:openhands:DEB
23:40:17 - openhands:DEB - logger.py:375 - Logging initialized
2025-09-15 23:40 - openhands:DEB - Logging initialized
2025-09-15 23:40 - openhands_ac.editor.file_cache:DEBUG - Current size updated: 0
2025-09-15 23:40 - openhands_ac.editor.file_cache:DEBUG - FileCache initialized with directory: /var/folders/wb/00shq152tlc_nmfy148f1r40000gn/T/o_h_editor_history_b6c6g1wb, size
_limit: None, current_size: 0
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0/openhands/core/config/config_utils.py:55: DeprecationWarning: deprecated
field_value = getattr(sub_config, field_name)
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0/openhands/core/config/utils.py:323: DeprecationWarning: deprecated
if 'cfg.workspace_base' is not None or 'cfg.workspace_mount_path' is not None:
23:40:18 - openhands:WARNING: utils.py:324 - DEPRECATED: The WORKSPACE_BASE and WORKSPACE_MOUNT_PATH environment variables are deprecated. Please use RUNTIME_MOUNT instead, e.g.
'RUNTIME_MOUNT=/my/host/dir/workspace';
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0/openhands/core/config/utils.py:370: DeprecationWarning: deprecated
elif cfg.workspace_base is not None or cfg.workspace_mount_path is not None:
/Users/lipeng/workspace/demo/Openhands/OpenHands-0.48.0/openhands/core/config/utils.py:371: DeprecationWarning: deprecated
if cfg.workspace_base is not None:

查看DockerDesktop容器

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage: 3.43% / 800% (8 CPUs available)

Container memory usage: 1.74GB / 11.4GB

Show charts

Search

Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Actions
openhands-runtime-64a708df-957c-40aa-8774-f1a6da1eb363-8e8170c 50a7c4780d43	all-hands-a	35767:35767	3.43%		
dataagent	-	-	-		

Showing 2 items

Engine running | RAM 2.64 GB CPU 0.62% Disk: 9.64 GB used (limit 998.98 GB) | Terminal | New version available

openhands容器已经启动,点击查看容器日志

Containers / openhands-runtime-64a708df-957c-40aa-8774-f1a6da1eb363-8e8170c8ecdd6590

openhands-runtime-64a708df-957c-40aa-8774-f1a6da1eb363-8e8170c8ecdd6590

50a7c4780d43 ↻ all-hands-a/runtime:0.48-nikolaik 35767:35767 ↻ 50213:50213 ↻ Show all ports (3)

Status: Running (2 minutes ago)

Logs Inspect Bind mounts Exec Files Stats

```

2025-09-15 23:40:23 DEBUG:openhands:DEBUG mode enabled.
2025-09-15 23:40:23 15:40:23 - openhands:DEBUG logger.py:375 - Logging initialized
2025-09-15 23:40:23 15:40:23 - openhands:DEBUG logger.py:387 - Logging to file in: /openhands/code/logs
2025-09-15 23:40:23 2025-09-15 15:40 - openhands_aci:DEBUG - Logger initialized
2025-09-15 23:40:25 /openhands/poetry/openhands-ai-504_aCHF-py3.12/lib/python3.12/site-packages/pydub/utils.py:300: SyntaxWarning: invalid escape sequence '\(
'
2025-09-15 23:40:25 m = re.match('([su][0-9]{1,2})p?(\{[0-9]{1,2}\})?$', token)
2025-09-15 23:40:25 /openhands/poetry/openhands-ai-504_aCHF-py3.12/lib/python3.12/site-packages/pydub/utils.py:301: SyntaxWarning: invalid escape sequence '\(
'
2025-09-15 23:40:25 m2 = re.match('([su][0-9]{1,2})p?(\{[0-9]{1,2}\})?$', token)
2025-09-15 23:40:25 /openhands/poetry/openhands-ai-504_aCHF-py3.12/lib/python3.12/site-packages/pydub/utils.py:310: SyntaxWarning: invalid escape sequence '\(
'
2025-09-15 23:40:25 elif re.match('(flt)p?(\{[0-9]{1,2}\})?$', token):
2025-09-15 23:40:25 /openhands/poetry/openhands-ai-504_aCHF-py3.12/lib/python3.12/site-packages/pydub/utils.py:314: SyntaxWarning: invalid escape sequence '\(
'
2025-09-15 23:40:25 elif re.match('(dbl)p?(\{[0-9]{1,2}\})?$', token):
2025-09-15 23:40:25 2025-09-15 15:40 - openhands_aci.editor.file_cache:DEBUG - Current size updated: 0
2025-09-15 23:40:25 2025-09-15 15:40 - openhands_aci.editor.file_cache:DEBUG - FileCache initialized with directory: /tmp/oh_editor_history_bfq976c, size_limit: None, current_size: 0
2025-09-15 23:40:27 15:40:27 - openhands:WARNING: action_execution_server.py:659 - Starting Action Execution Server
2025-09-15 23:40:27 15:40:27 - openhands:INFO: action_execution_server.py:680 - Starting file viewer server
2025-09-15 23:40:27 15:40:27 - openhands:INFO: file_viewer_server.py:94 - File viewer server URL saved to /tmp/oh-server-url: http://localhost:36615
2025-09-15 23:40:27 15:40:27 - openhands:INFO: file_viewer_server.py:95 - Starting file viewer server on port 36615
2025-09-15 23:40:27 15:40:27 - openhands:INFO: action_execution_server.py:685 - File viewer server started at http://localhost:36615
2025-09-15 23:40:27 15:40:27 - openhands:DEBUG: action_execution_server.py:1068 - Starting action execution API on port 35767
2025-09-15 23:40:27 INFO: Started server process [8]
2025-09-15 23:40:27 INFO: Waiting for application startup.
2025-09-15 23:40:27 INFO: Application startup complete.
2025-09-15 23:40:27 INFO: Started server process [8]
2025-09-15 23:40:27 INFO: Waiting for application startup.
2025-09-15 23:40:27 15:40:27 - openhands:INFO: action_execution_server.py:700 - Initializing ActionExecutor...
2025-09-15 23:40:27 15:40:27 - openhands:DEBUG: runtime_init.py:53 - Client working directory: /workspace
2025-09-15 23:40:27 INFO: Uvicorn running on http://127.0.0.1:36615 (Press CTRL+C to quit)
2025-09-15 23:40:27 15:40:27 - openhands:DEBUG: runtime_init.py:65 - Created working directory. Output: []

```

Engine running | RAM 2.64 GB CPU 0.50% Disk: 9.64 GB used (limit 998.98 GB) | Terminal | New version available

8. 查看agent日志

打开logs/lm/{run_datetime} 查看agent交互日志

The screenshot shows the OpenHands-0.48.0 application window. On the left is a file tree with a red box highlighting the 'logs' folder under '25-09-15_23-40'. The right side has tabs for '问题', '输出', '调试控制台', and '终端'. The '终端' tab is active, showing log output from 23:42:46 to 23:42:46. The logs include several entries related to 'openhands:DEBUG' and 'conversation_memory.py' calls, indicating interactions with a tool or editor. A large 'X' watermark is overlaid on the terminal area.

```

23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/validation.py"}', name='str_replace_editor'), id='tool_u_07', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/dataset/transformer-data/sample_submission.csv"}', name='str_replace_editor'), id='tool_u_08', type='function']]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "head -n 5 /workspace/dataset/transformer-data/sample_submission.csv"}', name='execute_bash'), id='tool_u_09', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "head -n 5 /workspace/dataset/transformer-data/train.csv"}', name='execute_bash'), id='tool_u_10', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "head -n 5 /workspace/dataset/transformer-data/test.csv"}', name='execute_bash'), id='tool_u_11', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/scripts"}', name='str_replace_editor'), id='tool_u_12', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/scripts/EXP-LongForecasting/Dlinear", name='str_replace_editor'}, id='tool_u_13', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/scrpts/EXP-LongForecasting/Dlinear/excute_rate.h", name='str_replace_editor'}, id='tool_u_14', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/run_longExp.py", "view_range": [1, 58]}', name='str_replace_editor'), id='tool_u_15', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/run_longExp.py", "view_range": [59, 100]}', name='str_replace_editor'), id='tool_u_16', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/data_provider/data_factory.py", "view_range": [1, 80]}', name='str_replace_editor'), id='tool_u_17', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/data_provider/data_loader.py", "view_range": [1, 50]}', name='str_replace_editor'), id='tool_u_18', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "grep -n \\"\\Dataset_Custom\\\" /workspace/Dlinear/data_provider/data_loader.py"}', name='execute_bash'), id='tool_u_19', type='function')]
23:42:46 - openhands:DEBUG: conversation_memory.py:245 - Tool calls type: <class 'list'>, value: [ChatCompletionMessageToolCall(index=1, function=Function(arguments='{"command": "view", "path": "/workspace/Dlinear/data_provider/data_loader.py", "view_range": [192, 250]}', name='str_replace_editor'), id='tool_u_20', type='function')]
23:42:46 - openhands:DEBUG: logger.py:458 - Logging to /Users/lipengspace/demo/Openhands-0.48.0/logs/llm/25-09-15_23-40/prompt_021.log

```

9. 查看结果

任务完成后会根据你benchmark的测评要求输出结果：

```

69
70 评估脚本最终会输出:
71  ````text
72 Successfully submitted to LLM Merging Competition
73 Waiting for Kaggle leaderboard to refresh ...
74
75 Your merged model scores 0.49 out of 1.00 on the test set!
76 最终得分: 0.4900
77  ````


```

可以在当前目录的日志文件中查看：

34	## 评估
35	
36	### 开发阶段评估
37	``` bash
38	HF_ENDPOINT=https://
39	```
40	### 输出说明
41	
42	1. **评估结果**：保存
43	2. **控制台输出**：显
44	3. **最终得分**：显示
45	

≡ openhands_2025-09-16.log	46	评估脚本最终会输出：
> └ microagents	47	```text
> └ openhands	48	评估结果已保存到： ./d
> └ openhands-ui	49	
> └ tests	50	Your merged model :
> └ third_party	51	最终得分： 0.6438
> └ workspace	52	
≡ .dockerignore	53	...
ⓧ .editorconfig	54	
≡ .gitattributes	55	### 注意事项
ⓧ .gitignore	56	
≡ .nvmrc	57	1. 确保模型路径存在且
└ build.sh	58	2. 如果使用GPU，确保有
≡ CITATION.cff	59	3. 首次运行需要下载 T
M+ CODE_OF_CONDUCT.md	60	4. 评估时间取决于模型
M+ COMMUNITY.md	61	
T config.template.toml	62	
T config.toml	63	### 测试阶段评估
M+ CONTRIBUTING.md	64	
M+ CREDITS.md	65	生成submission.csv后
M+ Development.md	66	```bash
Y docker-compose.yml	67	python evaluate_kag
≡ experiment-task.txt	68	...
M+ ISSUE_TRIAGE.md	69	
≡ LICENSE	70	评估脚本最终会输出：
M Makefile	71	```text
≡ MANIFEST.in	72	Successfully submit
T poetry.lock	73	Waiting for Kaggle
Y pydoc-markdown.yml	74	
T pyproject.toml	75	Your merged model :
≡ pytest.ini	76	最终得分： 0.4900
M+ README.md	77	...
M+ README_CN.md		
M+ README_JA.md		
└ run.sh		
└ run_experiment.sh		
⚡ start.py		
≡ task.txt		
> └ External Libraries		
≡ Scatches and Consoles		