ML 2026 SpringHW2

TAs: 林禹融 劉建豐 楊樂霖

Email: ntu-ml-2026-spring-ta@googlegroups.com

Deadline: 2026/4/2 23:59:59 (UTC+8)

Links

- Course Website
- NTU COOL
- Colab Sample Code
- Judgeboi Link

Outline

- How to work with AI agents?
- Task Introduction
- Code Overview
- Grading
- FAQs
- Hints

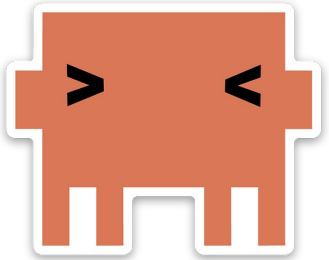
How to work with AI agents?

Powerful AI coding tools

Cursor

Antigravity

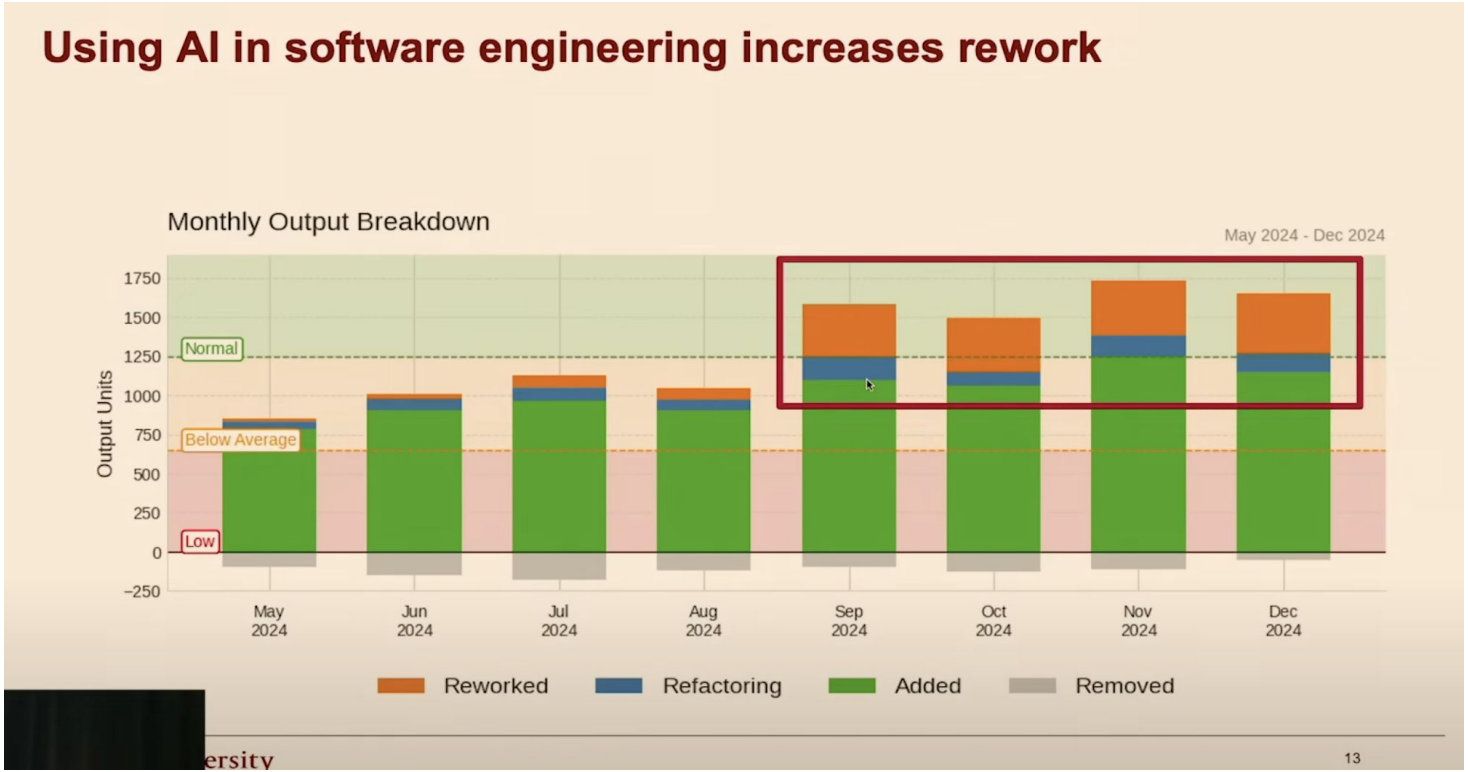
copilot



Claude Code

Codex

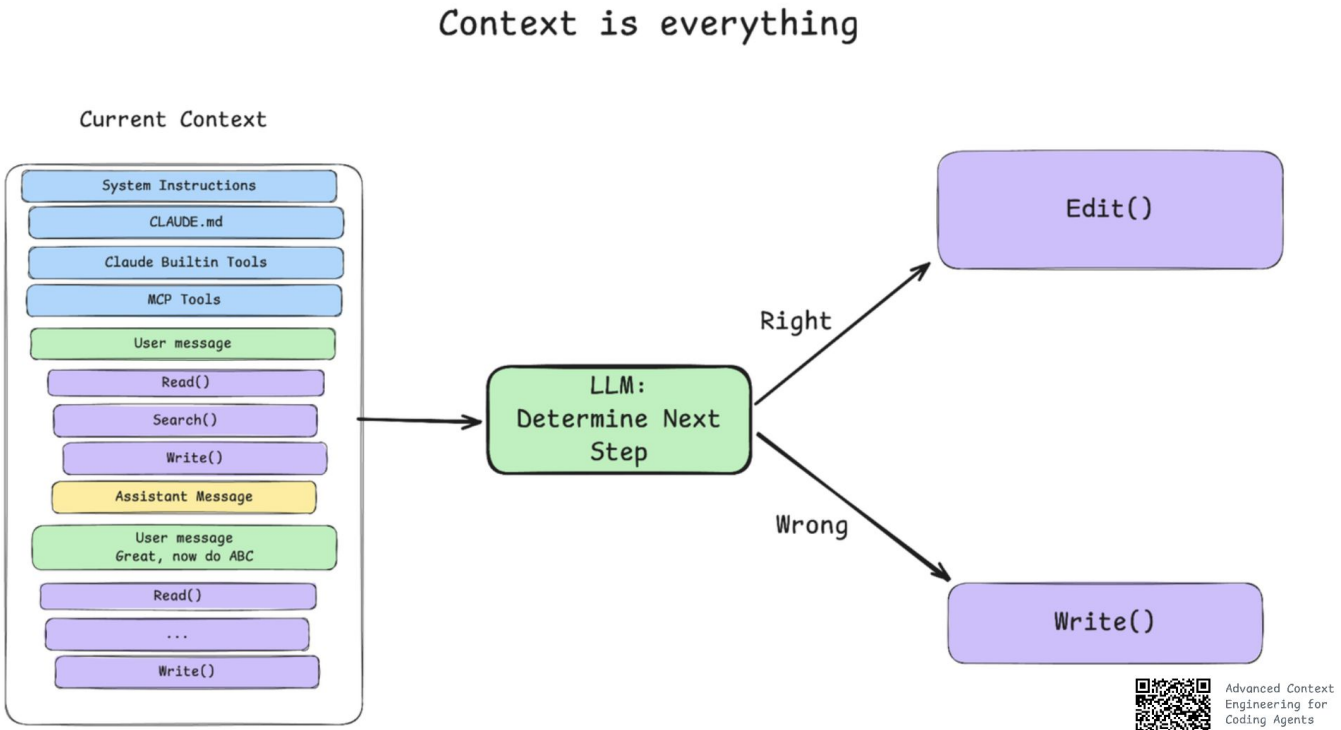
The problem of coding with AI agents



Credit: [Does AI Actually Boost Developer Productivity? (100k Devs Study) - Yegor Denisov-Blanch, Stanford]

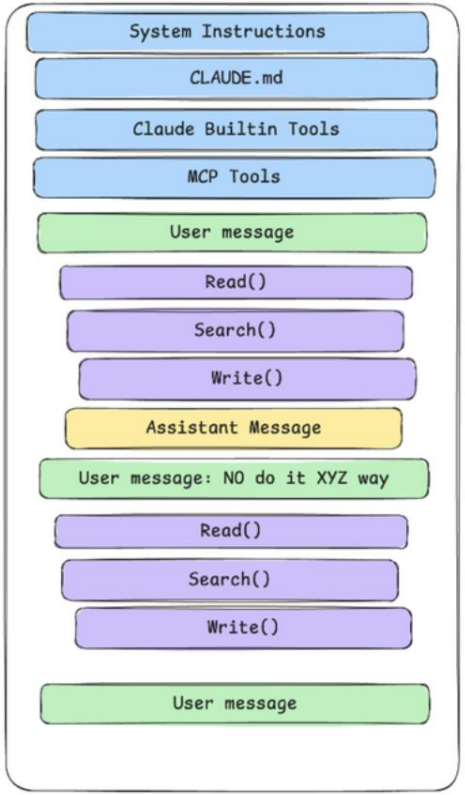
Context is Everything

Garbage in, garbage out



Credit: [advanced-context-engineering-for-coding-agents]

the naive way - work until you run out of context



Resteering in context:

"NO use XYZ Approach"

Advanced Context Engineering for Coding Agents

@dexhorthy

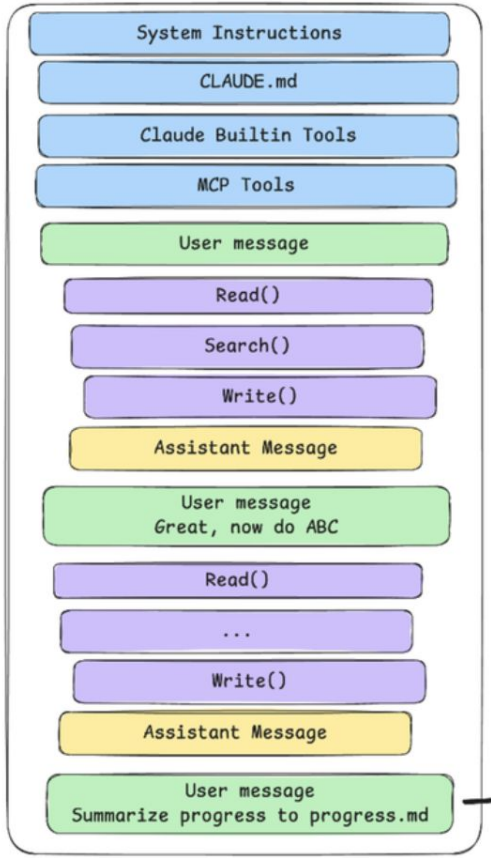
Credit: advanced-context-engineering-for-coding-agents

Intentional Compaction

100k lines codes

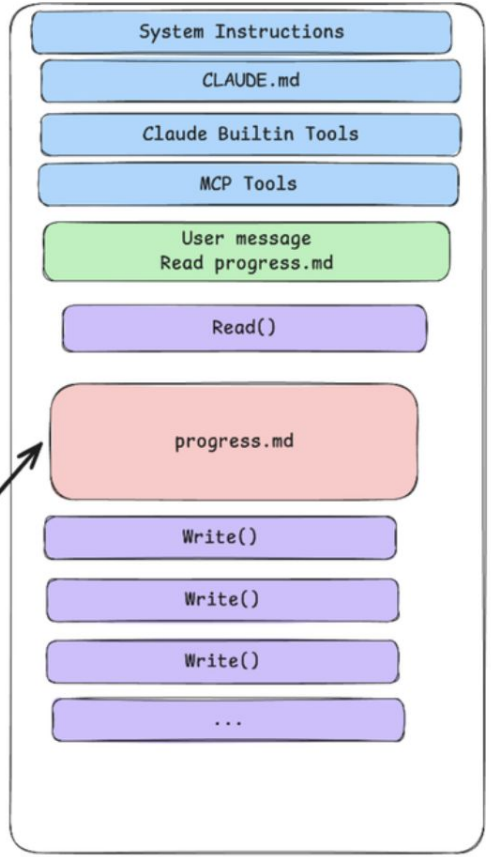
10k lines markdown

Slightly smarter - intentional compaction



Intentional Compaction

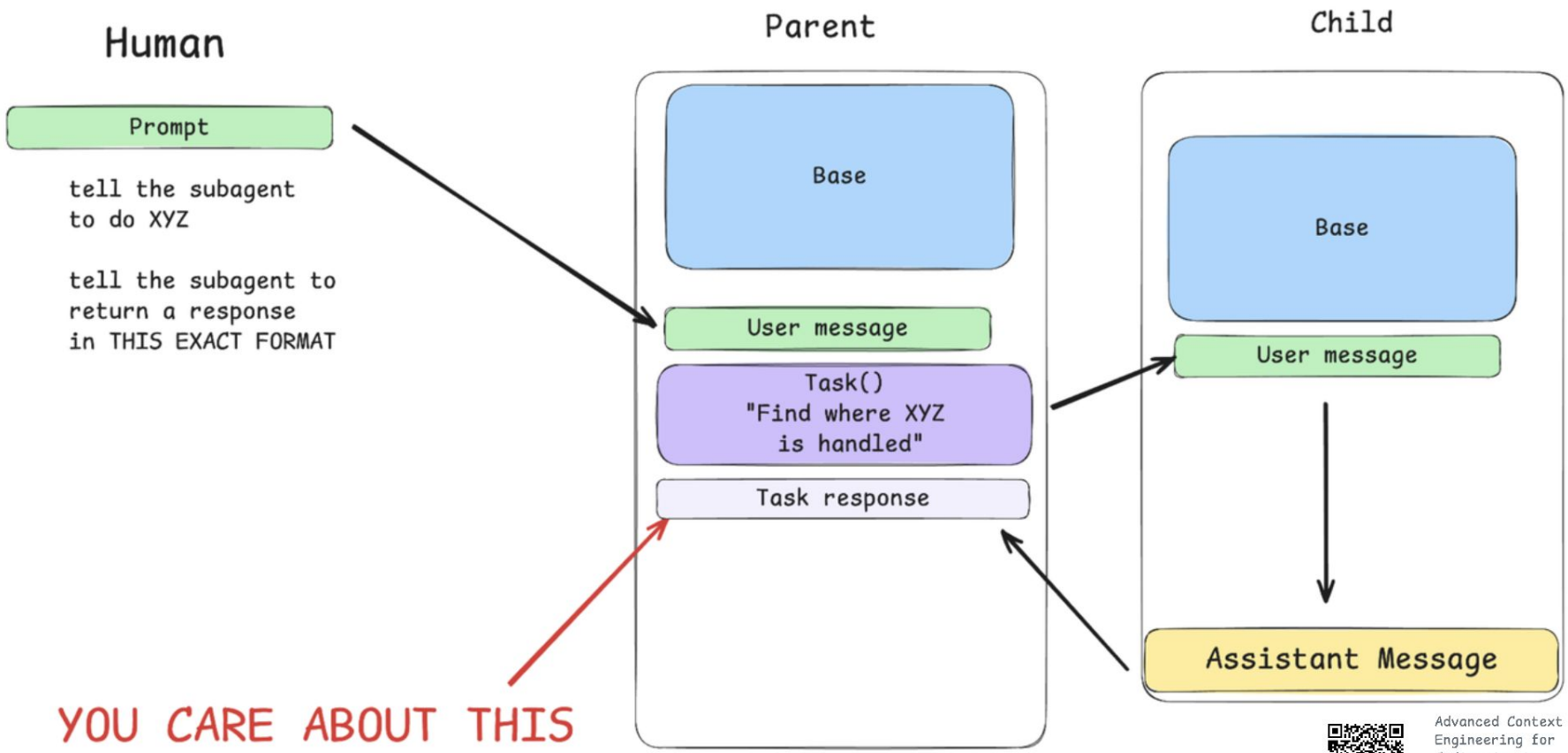
"Write everything we did
so far to progress.md,
ensure to note the approach
we're taking, the steps
we've done so far,
and the current failure
we're working on"



Advanced Context
Engineering for
Coding Agents

@dexhorthy

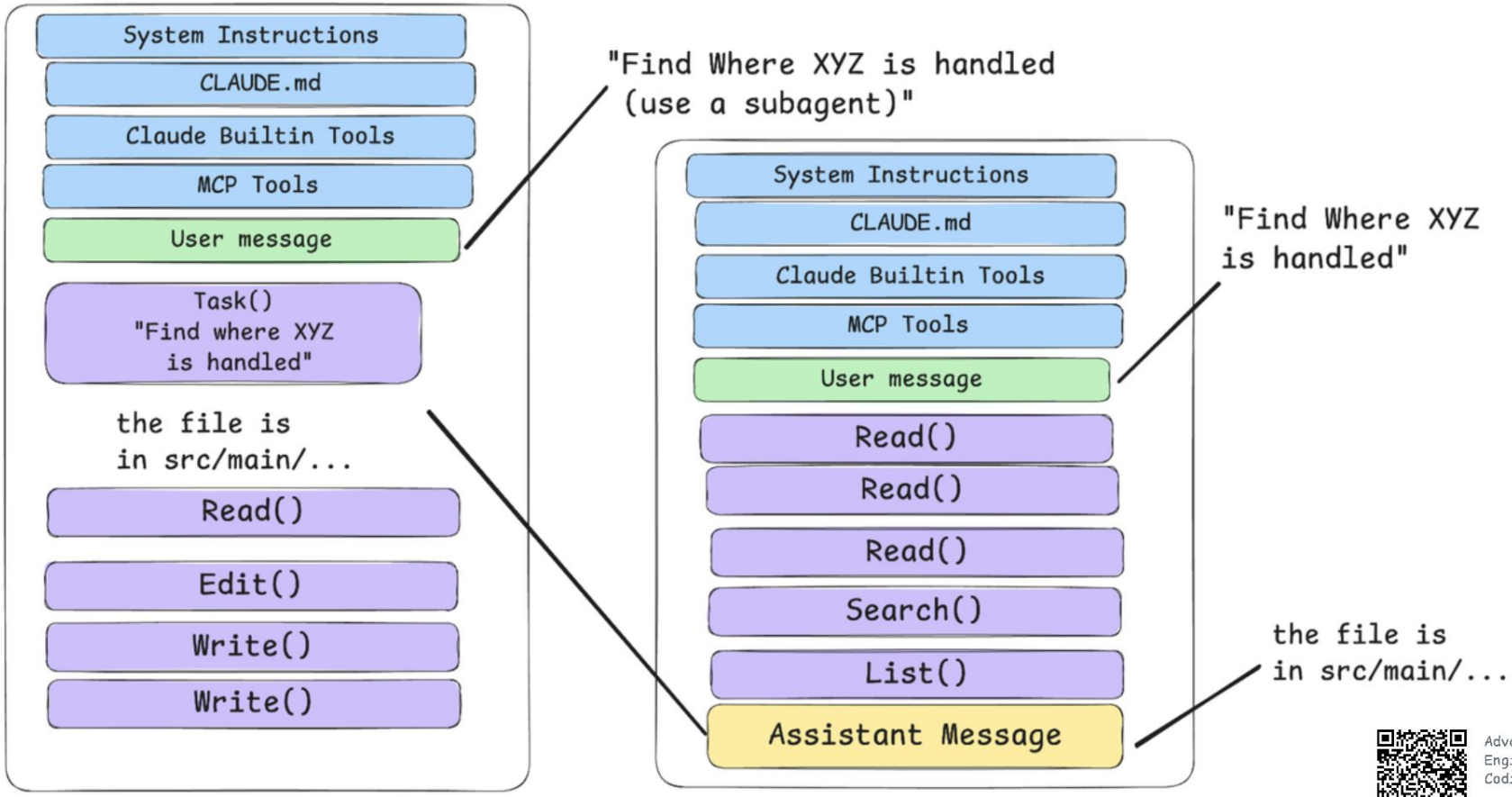
Playing Telephone



Advanced Context Engineering for Coding Agents

@dexhorthy

Managing Context with Sub Agents



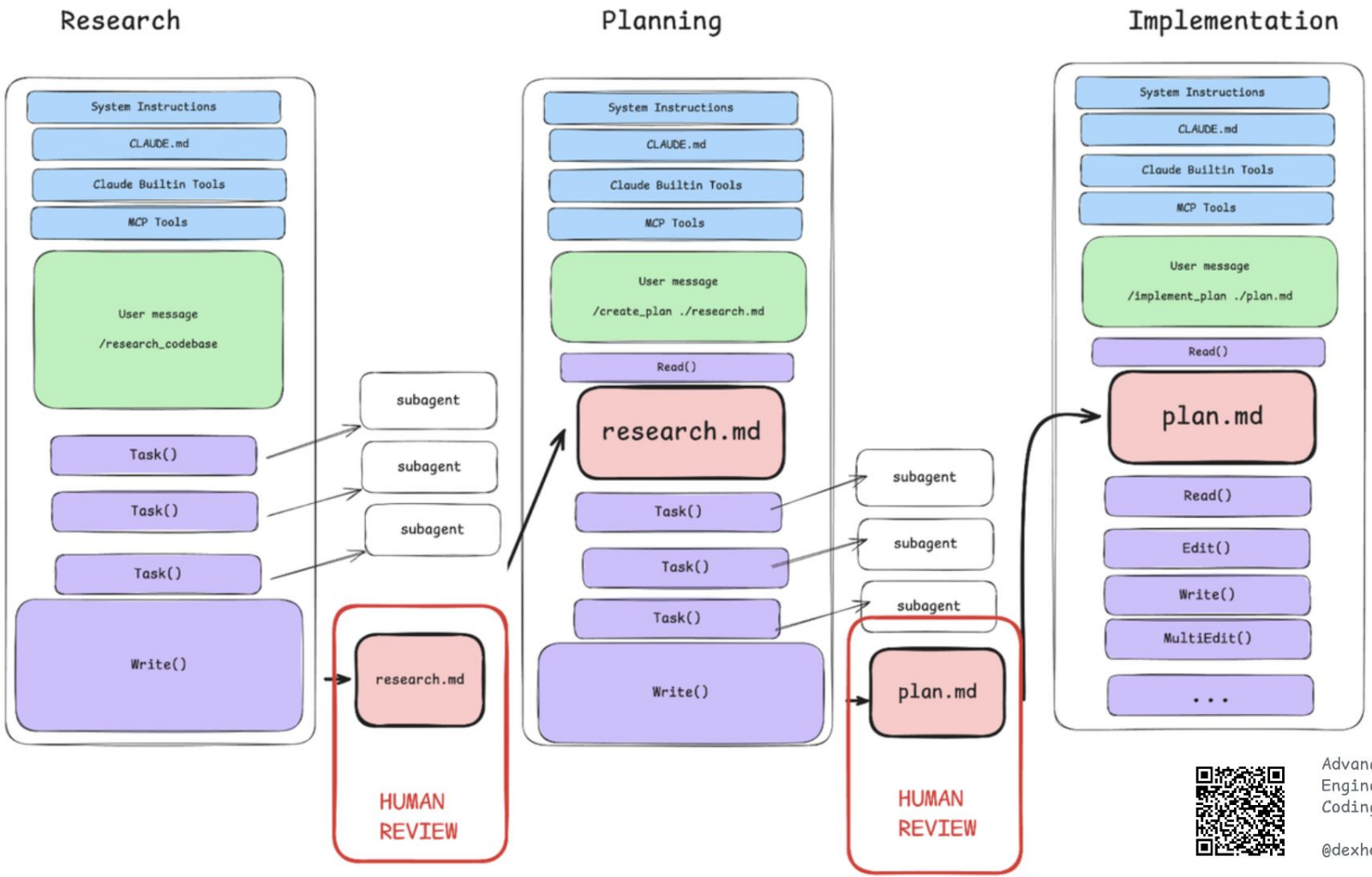
Advanced Context Engineering for Coding Agents

@dexhorthy

Frequent Intentional Compaction

- **Research (Optional)**
  - Understand the codebase, the files relevant to the issue, and how information flows, and perhaps potential causes of a problem.
- **Plan**
  - Outline the exact steps we'll take to fix the issue, and the files we'll need to edit and how, being super precise about the testing / verification steps in each phase.
- **Implement**
  - Step through the plan, phase by phase. For complex work, I'll often compact the current status back into the original plan file after each implementation phase is verified.

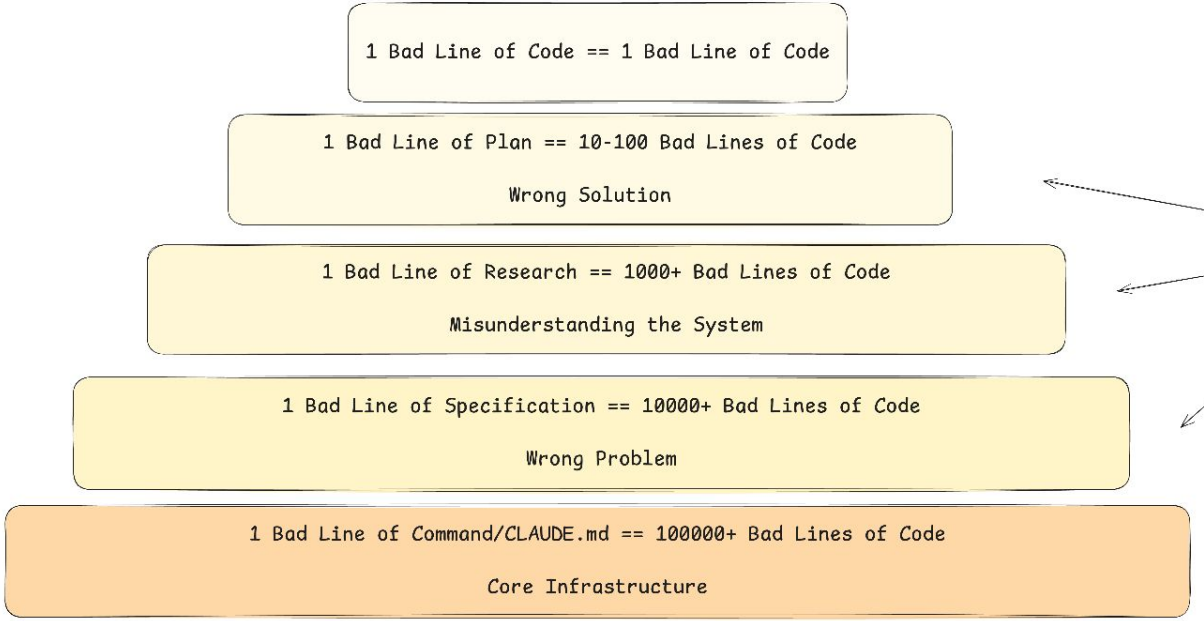
Research □ Plan □ Implement



Advanced Context Engineering for Coding Agents

@dexhorthy

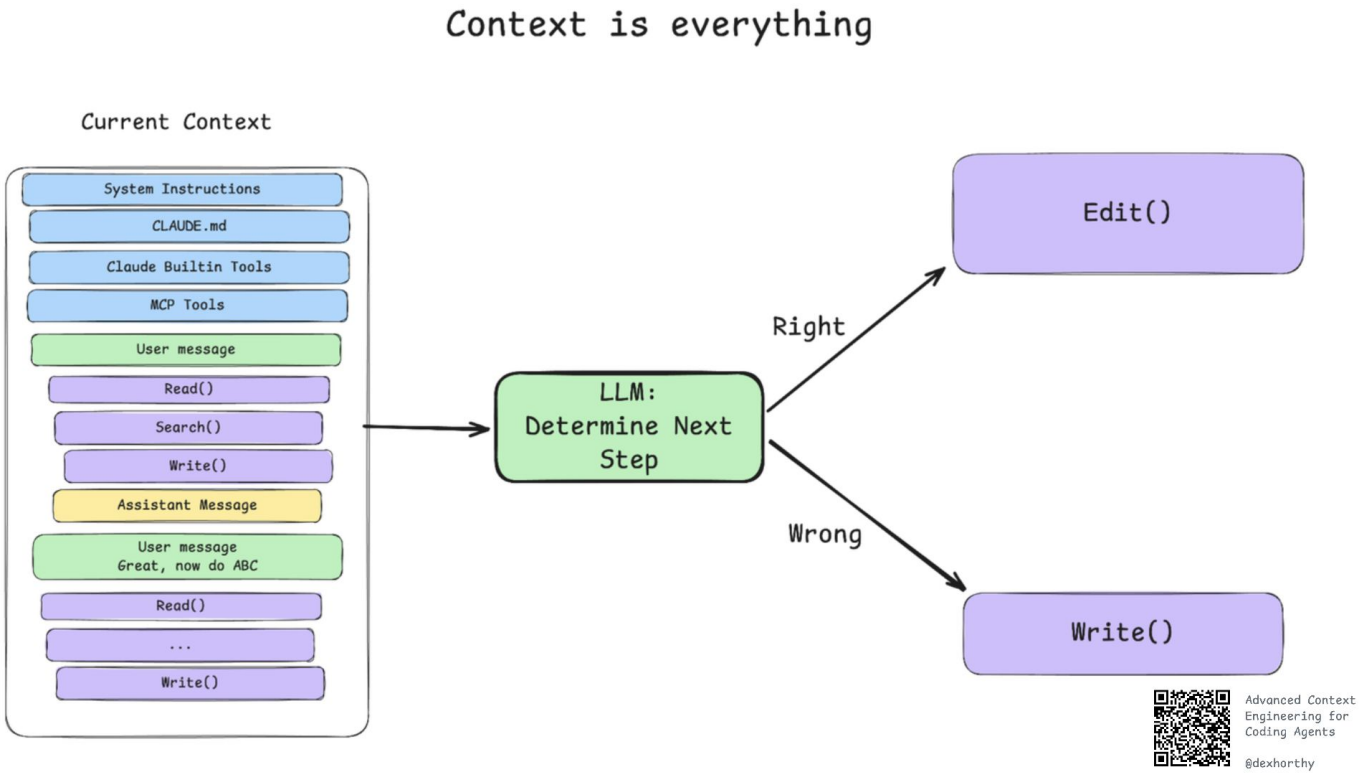
Impact Hierarchy for Coding Agents



Human Effort and Focus on the HIGHEST LEVERAGE parts of the pipeline

Takeaway

The Agent is only as smart as the context you provide.



Task Introduction

Task Introduction - AI Agent as an AI Engineer

- Use LLMs to automatically write code
- By only knowing the dataset, LLMs can generate customized solutions for each task

Credit: Task edited from 2025GenAI-ML-HW6 & 2025ML-HW2

Task - Celebrity Face Identification

- Given an image of an idol’s face, classify the individual into one of 10 predefined categories.
- Source: Speech Processing and Machine Learning Lab@NTU
- Do NOT search for or use additional data for training or the answers for the testing data.
- Can only use CNN & LLM models. VLM & other powerful models are forbidden.
- The LLM agent serves as your representative—if it violates the rules, it's as if you did.

Dataset - MyGO & Ave Mujica

- Total Samples: ~2,400
  - Training Set: 1,600
  - Validation Set: 300
  - Test Set: 500 (Public: 250 / Private: 250)
- Classes (10 total):
  - MyGO: tomori, anon, soyo, taki, rana
  - Ave Mujica: sakiko, nyamuchi, umiri, uika, mutsumi

Dataset - Format

- Size: 150 * 150 * 3
- Metadata: Unique ID and filename for each entry
- Label Format: String (Idol names)

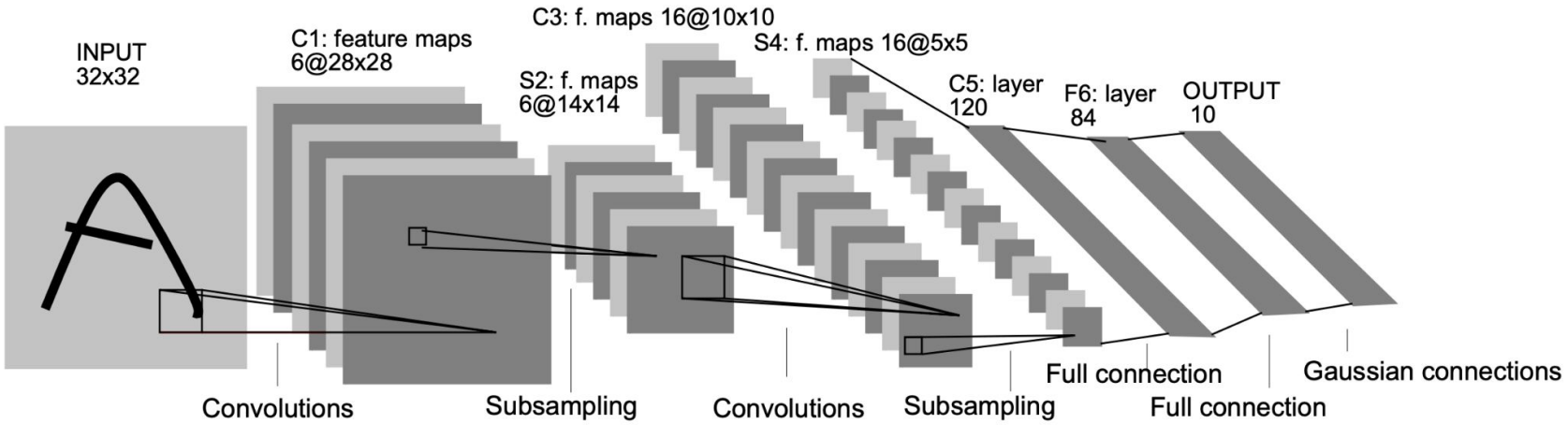
json format:

```
{
    "id": 6,
    "filename": "6.png",
    "label": "anon"
},
```

```
hw2_data/
|--train.json
|--val.json
|--test.json
|--train/
|--|--0.png
|--|...
|--|--1577.png
|--val/
|--|--1578.png
|--|...
|--|--1877.png
|--test/
|--|--1878.png
|--|...
|--|--2377.png
```

CNN

Recommendation: Fine-tuning ResNet18 for 5~10 epochs is sufficient to exceed the strong baseline.



GradientBased Learning Applied to Document Recognition

http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

CNN - Core Architecture

- **Convolutional Layer** — Extracting local patterns.
- **Activation Layer** — Enhancing non-linearity.
- **Pooling Layer** — Reducing spatial dimensions.
- **Fully Connected Layer** — Performing final classification.

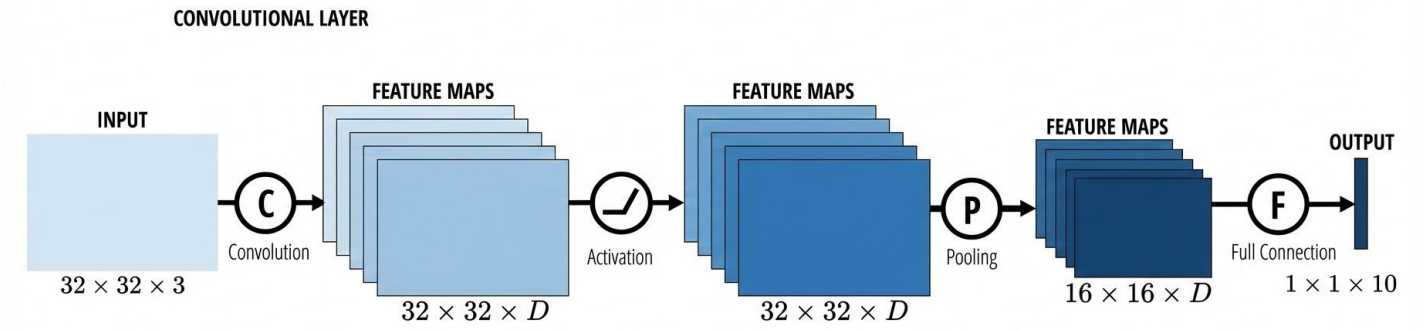


Image credit: Yu-Chiang Frank Wang

CNN - Core Properties

- **Local Connectivity:** Each neuron takes info only from a neighborhood of pixels.
- **Weight Sharing:** Neurons connecting each pixel and its neighborhoods have identical weights.

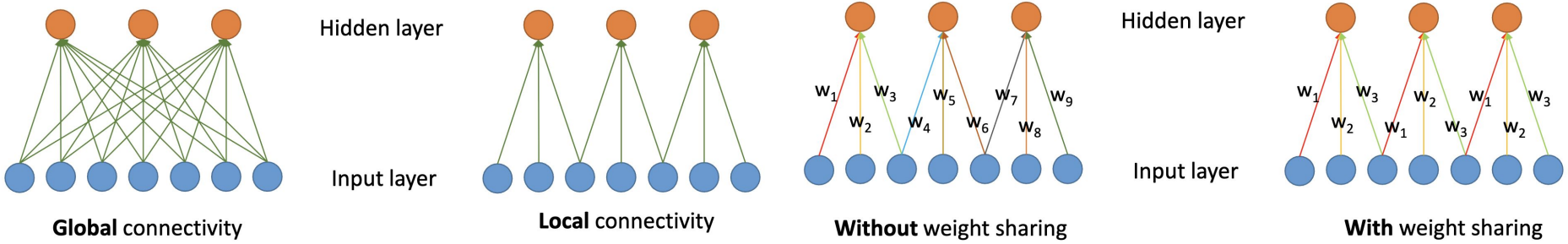


Image credit: Yu-Chiang Frank Wang

CNN - Core Properties

- Local Connectivity
- Weight Sharing
- **Handling multiple input channels**
- **Handling multiple output maps**

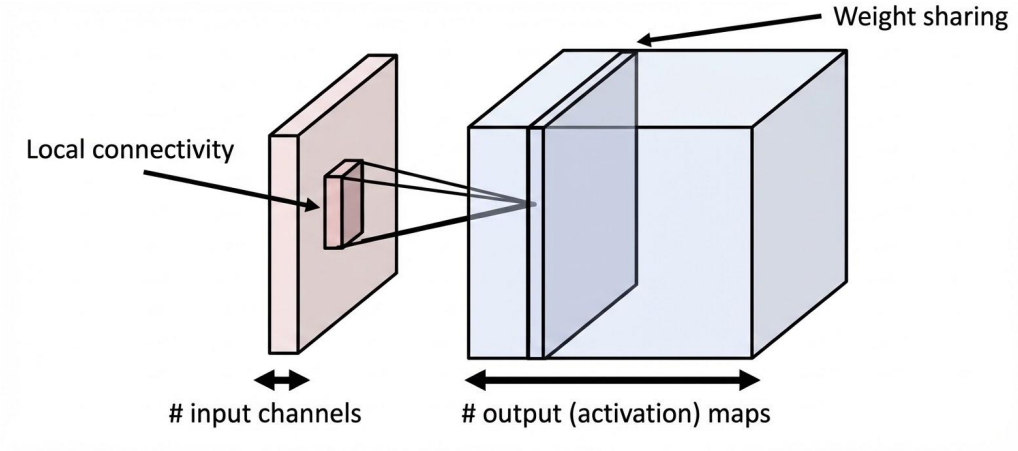


Image credit: A. Karpathy

Reference Materials

- ML Lecture 10: Convolutional Neural Network-ML2017Fall
- 【機器學習2021】卷積神經網路 (Convolutional Neural Networks, CNN)
- [ML 2021 (English version)] Lecture 9: Convolutional Neural Networks
- 【生成式人工智慧與機器學習導論2025】HW6 Deep Learning for Image Classification

Code Overview

AI-Driven Exploration

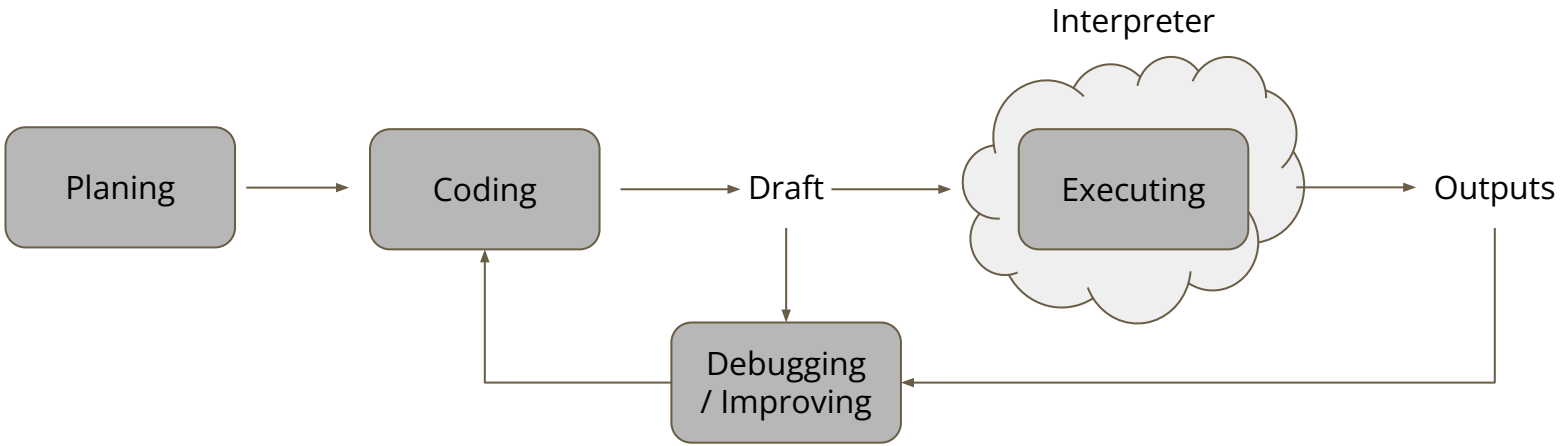
- **Main idea:**
  - Heuristic Best-First Search
- **Core modules:**
  - evaluator $h(s_n)$
  - solution tree T_n
  - search policy $\pi(T_n)$
  - summarization operator $\Sigma(T_{n-1})$
  - coding operator $f(s, \Sigma(T_{n-1}))$

Whichever code version produces the best results, the AI will use it as the foundation for the next optimization step.

AIDE: AI-Driven Exploration in the Space of Code
https://arxiv.org/pdf/2502.13138

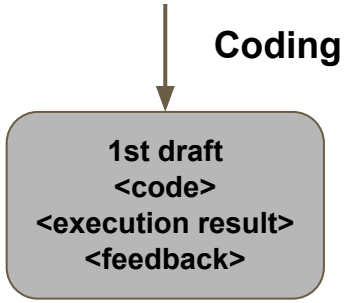
Sample Code Structure

- The only thing we need to do is instruct LLM agents to handle that!



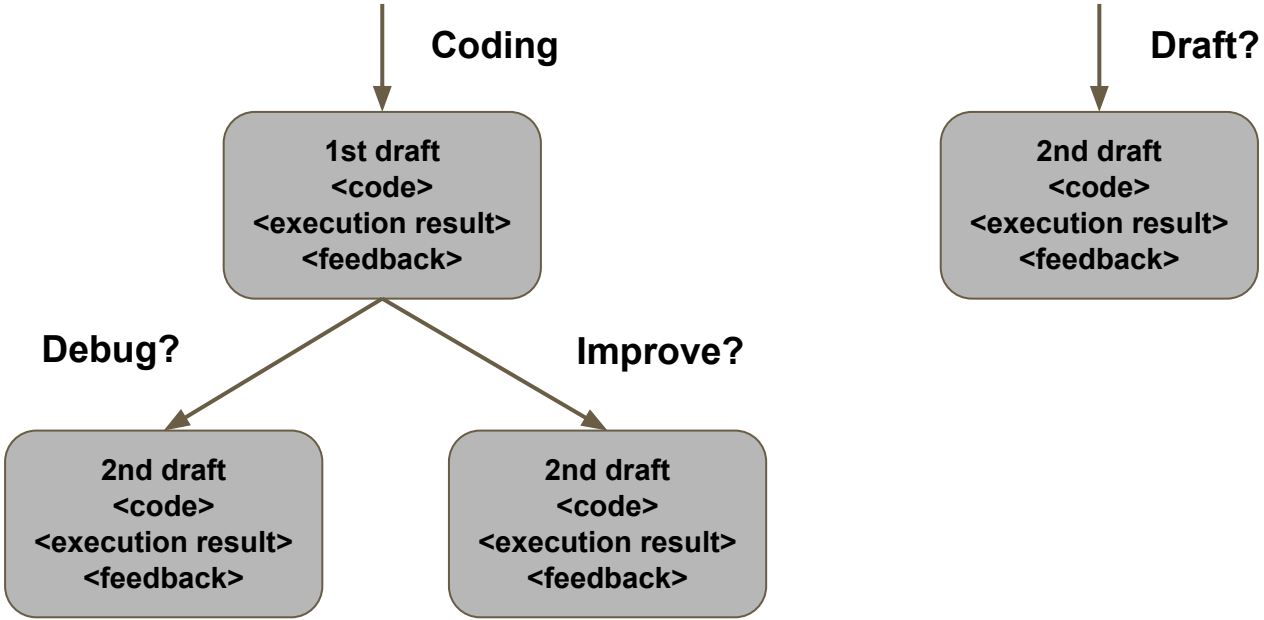
AIDE: AI-Driven Exploration in the Space of Code
https://arxiv.org/pdf/2502.13138

Action Space - First Step



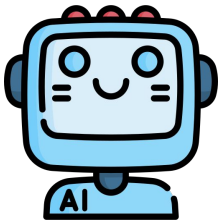
AIDE: AI-Driven Exploration in the Space of Code
https://arxiv.org/pdf/2502.13138

Action Space - Second Step



AIDE: AI-Driven Exploration in the Space of Code
https://arxiv.org/pdf/2502.13138

Artifact Examples - Plan

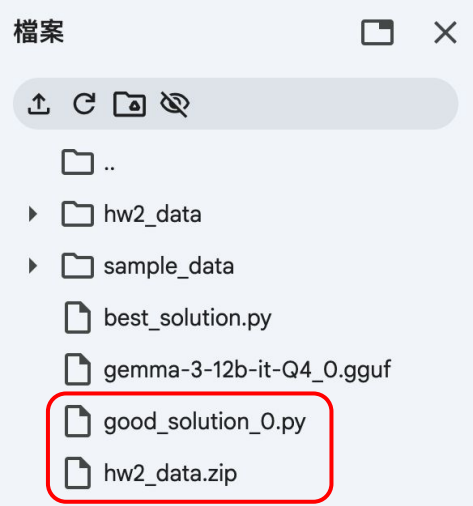


We will use the following steps:

....

1. Load the necessary libraries and data.
2. Preprocess the image data by resizing to a uniform shape, normalizing pixel values.
3. Split the data into training and validation sets.
4. Train a CNN model on the training data.
5. Make predictions on the validation data.
6. Save the predictions to a submission file.

Artifact Examples - Code



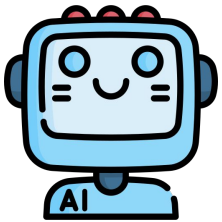

```python
# Define a function to save the best solution and other good solutions to files.
def save_run(cfg, journal):
    # Retrieve and save the best found solution.
    best_node = journal.get_best_node(only_good=False)  # Get the best node.
    with open("best_solution.py", "w") as f:
        f.write(best_node.code)

    good_nodes = journal.get_good_nodes()  # Retrieve all good solution nodes.
    for i, node in enumerate(good_nodes):
        filename = f"good_solution_{i}.py"
        with open(filename, "w") as f:
            f.write(node.code)
```

Artifact Examples - Execution

```
Traceback (most recent call last):
  File "/content/best_solution.py", line 54, in <module>
    img = preprocess_image(item['filename'], IMG_WIDTH, IMG_HEIGHT)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/content/best_solution.py", line 27, in preprocess_image
    img = load_img(filename, target_size=(width, height))
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/keras/src/utils/image_utils.py", line 235, in load_img
    with open(path, "rb") as f:
         ^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: '0.png'
```

Artifact Examples - Feedbacks



The error message indicates that a `FileNotFoundError` occurred because the system could not locate the file named **'0.png'**. This happens because the `load_img` function is receiving a relative filename that does not exist in the current working directory, or the path provided in your data structure is incomplete.

……

To fix this issue, you should ensure that the `filename` being passed to `load_img` includes the correct directory path. You can use `os.path.join()` to combine your image folder path with the filename.

Grading

Grading

Code Submission (+4 pts)	Submit your code to **NTU COOL**.
Public Simple Baseline (+1 pt)	Just run the provided sample code and submit '**pred.json**' to the **Judgeboi**.
Private Simple Baseline (+1 pt)	
Public Medium Baseline (+1 pt)	Improve your LLM agents with the following methods: Prompt Engineering Data augmentation Use different LLM & CNN model Drafting More & Improving & Debugging ……..
Private Medium Baseline (+1 pt)	
Public Strong Baseline (+1 pt)	
Private Strong Baseline (+1 pt)	

Grading - Code Submission

- Submit **ML2026Spring_hw2.ipynb** to NTU COOL
- **Deadline: 4/2 (Thu.) 23:59**
- Compress your code into zip if not using .ipynb file
- **No late submission is allowed**
- We can only see your last submission.
- Do not submit model weights or dataset.
- **If your code is not reasonable or reproducible, you will receive 0 points for this homework.**

Grading - Metrics

Evaluation Metric: Accuracy (%)

$$\text{Acc} = \frac{\#\text{ of correct}}{N}$$

tomori

tomori

Passing public baselines doesn't guarantee that you will pass private ones.

Grading - Baselines

Public Baseline	Score	Estimated Drafting Time	Estimated Training Time
Simple	0.00	5 ~ 10 minutes	0 ~ 3 minutes
Medium	0.65	5 ~ 10 minutes	3 ~ 5 minutes
Strong	0.87	10 ~ 30 minutes	3 ~ 5 minutes

Estimated runtime is evaluated on a Google Colab T4 GPU.

Passing public baselines doesn’t guarantee that you will pass private ones.

Grading - Judgeboi

- Please submit **pred.json** to Judgeboi. (Only .json files are allowed.)
- **5** submission quota per day, reset at **23:59 (UTC+8)**.
- Public scores will be visible after submission.

“pred.json” format:

```
[
    {
        "id": 1878,
        "pred": "anon"
    },
…
]
```

Grading - Regulations

The LLM agent serves as your representative —if it violates the rules, it's as if you did.

- You should NOT plagiarize, if you use any other resource, you should cite it in the reference.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- Do NOT search for or use additional data for training or the answers for the testing data.
- Do NOT use closed-source LLM APIs like GPT-5, Gemini-3, etc.
- You should NOT modify your input file or prediction files manually.
- Make sure that TAs can reproduce the predictions using the code you submit. (Fix the random seed)
- Your final grade x 0.9 and get a score 0 for that homework if you violate any of the above rules first time (within a semester).
- Your will get F for the final grade if you violate any of the above rules multiple (> 1) times.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

Grading - Release Date

The grading of the homework will be released by 2026/**04/05** 23:59:59 (UTC+8)

FAQs

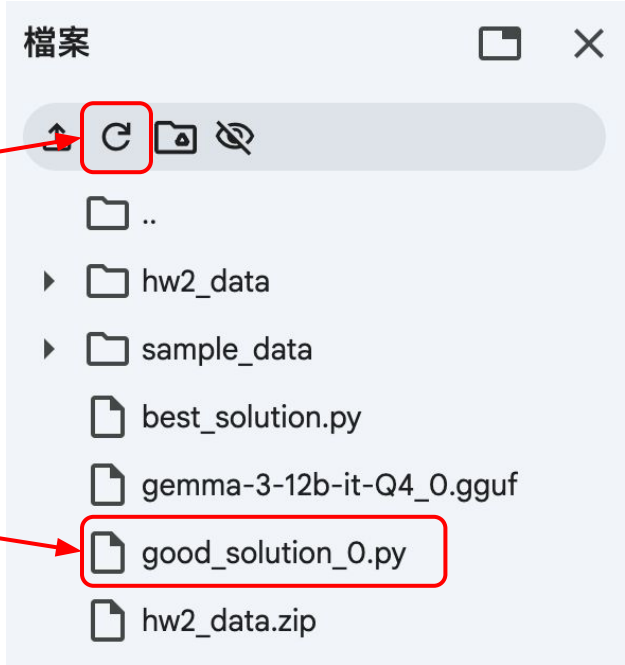
FAQs - No Submission File Generated

Q: I cannot find `submission.csv` after executing code…

A:

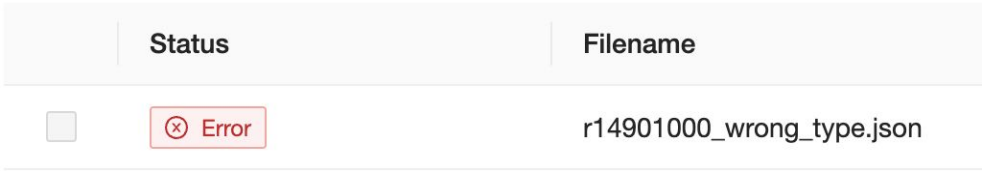
1. Refresh
2. Check best_solution.py and adjust your prompts (or try different LLMs)

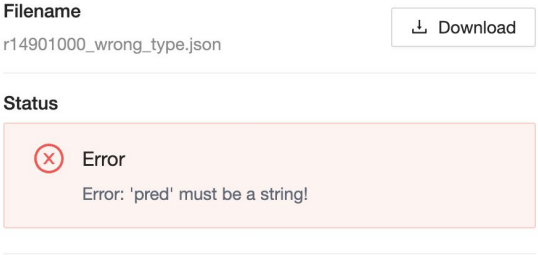
(Do NOT fix the python script manually).



FAQs - Judgeboi Submission Error

Q: When I submit `pred.json` to Judgeboi, an error occurs...





A: Open your `pred.json`, identify the errors, and adjust the prompts to fix it.
(Do NOT fix the prediction file manually).

Common errors include incorrect JSON format, invalid list length, incorrect prediction type, and missing 'pred' keys.

FAQs - Error Occurs When Loading LLMs

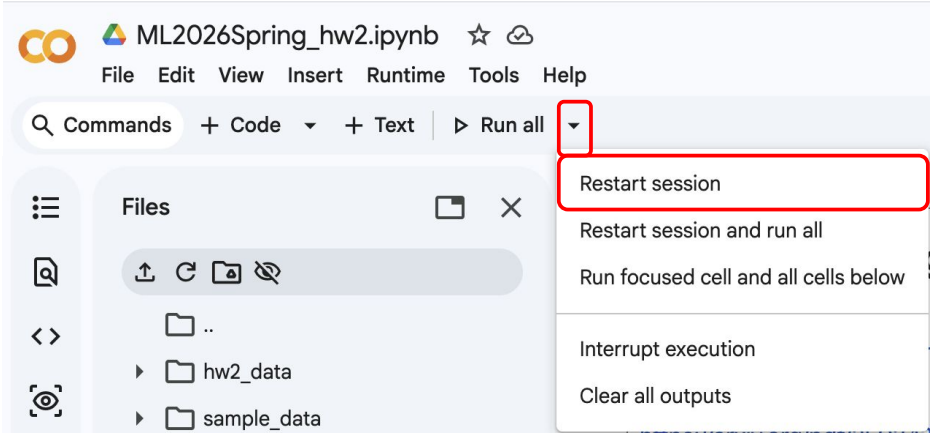
Q: When I load LLM, an error occurs...

1.

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-4216580818.py in <cell line: 0>()
      2
      3 # Load the model onto GPU
----> 4 myModel = Llama(
      5     # ========================= TODO: try different LLM =========================
      6     # !!! Before changing LLM setting, restart the session !!!
```

```
# Load the model onto GPU
myModel = Llama(
    # ========================= TODO: try different LLM =========================
    # !!! Before changing LLM setting, restart the session !!!
    "/content/gemma-3-12b-it-Q4_0.gguf",
    verbose=False,
    n_gpu_layers=-1,
    n_ctx=16384,    # This argument is how many tokens the model can take. The longer
)
```

2.



A:

1. Try to restart the session.
2. The checkpoint is too large for your GPU. Try a smaller one.

(Or modify context window size).

If any questions, you can ask us via...

- NTU Cool **HW2** 作業討論區
  - 如果同學的問題不涉及作業答案或隱私，請**一律使用** NTU Cool 討論區
  - 助教們會優先回答NTU Cool討論區上的問題
- Email: ntu-ml-2026-spring-ta@googlegroups.com
  - Title should start with [GenAI-ML 2026 Fall **HW2**]
  - Email with the wrong title will be moved to trash automatically
- TA hours
  - Each Friday before / after class:
    - (Fri.) 13.20 ~ 14.10 / 17:30~18:00
    - Location: 博理112

Hints

Hints - Prompt Engineering

Imagine that you're teaching a student to code!

```python
def _draft(self) -> Node:

    # ================ TODO: ask LLM agents to come up with a solution and then implement ================

    system_prompt = "You are an AI agent."

    user_prompt = [
        "You have to come up with a solution for image classification task and then implement this solution in Python."
        f"The task is to {str(self.cfg.task_goal)} ",
        f'All the provided input data is stored in "{self.cfg.data_dir}" directory.',
        f"{str(self.data_preview)}",
        'You have to save the predictions result on testing set in "/content/pred.csv".',
        'Note that the testing file DOES NOT have the label object.'
    ]

    system_message = system_prompt
    user_message = "\n".join(user_prompt)
    plan, code = self.plan_and_code_query(system_message=system_message, user_message=user_message)
    return Node(plan=plan, code=code)
```

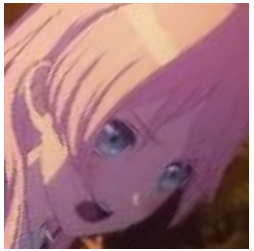
Hints - Data Augmentation

Original

Horizontal

Vertical

+45 Rotation



-45 Rotation

Crop

Brighter

Darker

Blur

Add noise

Grayscale

Hints - Random Seed & Temperature

```python
def set_seed(seed=531):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    if torch.cuda.is_available():
        torch.cuda.manual_seed(seed)
        torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.benchmark = False
    torch.backends.cudnn.deterministic = True

set_seed()
```

```python
def generate_response(_model: Llama, _messages: str) -> str:
    """
    This function will inference the model with given messages.
    """
    _output = _model.create_chat_completion(
        messages=_messages,
        stop=["<end_of_turn>", "<eos>"], # The standard end token of Gemma.
        max_tokens=8192,    # This argument is how many tokens the model can
        temperature=0,      # This argument is the randomness of the model.
    )["choices"][0]["message"]["content"]

    return _output
```

Hints - Different LLMs

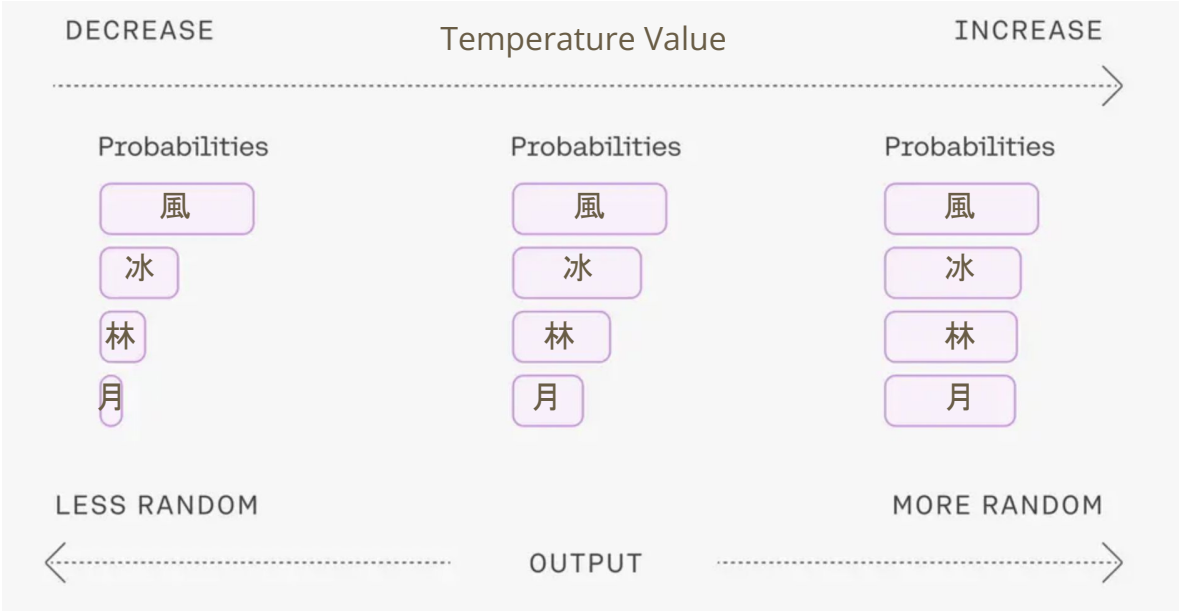
A stronger LLM might be helpful, but not always…

- How could we know which LLM is better?
- Model Size? Pretrained Corpus? arena?
- You can select the model from Hugging Face.
- Do NOT use closed-source LLM APIs like GPT-5, Gemini-3, etc.

Temperature

This slide is copied from GenAI2024 HW5

- Related to the diversity of the output, **0.0 ≤ temperature**
- Higher temperature for better diversity



Ref.

Hints - More Iterations (Optional)

```
# ================  TODO: config ================
config = {
    # experiment configurations
    "exp_name": "ML2026SPRING_HW2",
    "data_dir":  Path("/content/hw2_data").resolve(),

    # the description of the task
    "task_goal": "Given the images of different idols,\
                  predict the name of idol. \
                  The evaluation metric is Accuracy (ACC).",

    "agent": {
        # the number of iterations
        "steps": 1,
        "search": {
            # decide whether to debug or improve
            "debug_prob": 0.5,
            # the number of draft generated before improving/debugging
            "num_drafts": 1,
        },
    },
}

cfg = Config(config)
```

Hints - Improving (Optional)

```
def _improve(self, parent_node: Node) -> Node:

    # ================  TODO: ask LLM agent to improve drafts ================

    system_prompt = "You are an AI assistant."

    user_prompt = [
        f"Task description: {str(self.cfg.task_goal)} "
        f"Memory: {str(self.journal.generate_summary())} "
        f"Previous solution: Code: {str(wrap_code(parent_node.code))} "
    ]
    system_message = system_prompt
    user_message = " ".join(user_prompt)
    plan, code = self.plan_and_code_query(system_message=system_message, user_message=user_message)
    return Node(plan=plan, code=code, parent=parent_node)
```

Hints - Debugging (Optional)

```
def _debug(self, parent_node: Node) -> Node:

    # ================  TODO: ask LLM agent to debug ================
    system_prompt = "You are an AI agent."


    user_prompt = [
        f"Task description: {str(self.cfg.task_goal)}\n\n",
        f"Previous (buggy) implementation: {str(wrap_code(parent_node.code))}\n\n",
        f"Execution output: {str(wrap_code(parent_node.term_out, lang=''))}\n\n",
        str(self.data_preview)
    ]

    system_message = system_prompt
    user_message = " ".join(user_prompt)

    plan, code = self.plan_and_code_query(system_message=system_message, user_message=user_message)
    return Node(plan=plan, code=code, parent=parent_node)
```

Hints - Evaluating (Optional)

```python
def parse_exec_result(self, node: Node, exec_result: ExecutionResult):
    node.absorb_exec_result(exec_result)

    system_prompt = "You are an AI assistant."

    # ================  TODO: ask LLM agent to extract evaluation result from the execution output. ================
    # save log file
    user_prompt = f"""
        The task is:
        {self.cfg.task_goal}

        The code implementation is:
        {wrap_code(node.code)}

        The execution output is:
        {wrap_code(node.term_out, lang="")}
    """

    system_message = system_prompt
    user_message = " ".join(user_prompt)

    response = generate_response(
        myModel,
        _messages=[
            {'role': 'system', "content": system_message},
            {'role': 'user', "content": user_message}
        ]
    )
```

```python
# ================  TODO: evaluation ================
# you can force the LLM to structure the output to extract the metric(acc)
# reference: https://python.useinstructor.com/integrations/llama-cpp-python/
# node.analysis = response.summary
# node.is_buggy = (
#     response.is_buggy
#     or node.exc_type is not None
#     or response.metric is None
# )

node.is_buggy = False
node.metric = 1.0
```

Hints

- When changing LLMs, try to load checkpoints whose size are ~10GBs first.
- (For this task) Concise and precise when it comes to prompt design.
- (For this task) We recommend performing data augmentation and using a strong LLM first.
- Feel free to modify the pipeline.