Exploring Retrieval-Augmented Code Generation for Procedural Modelling in Houdini

Xuwen Dong
MSc Computer Animation and Visual Effects
NCCA – Bournemouth University

Motivation / Problem

Motivation

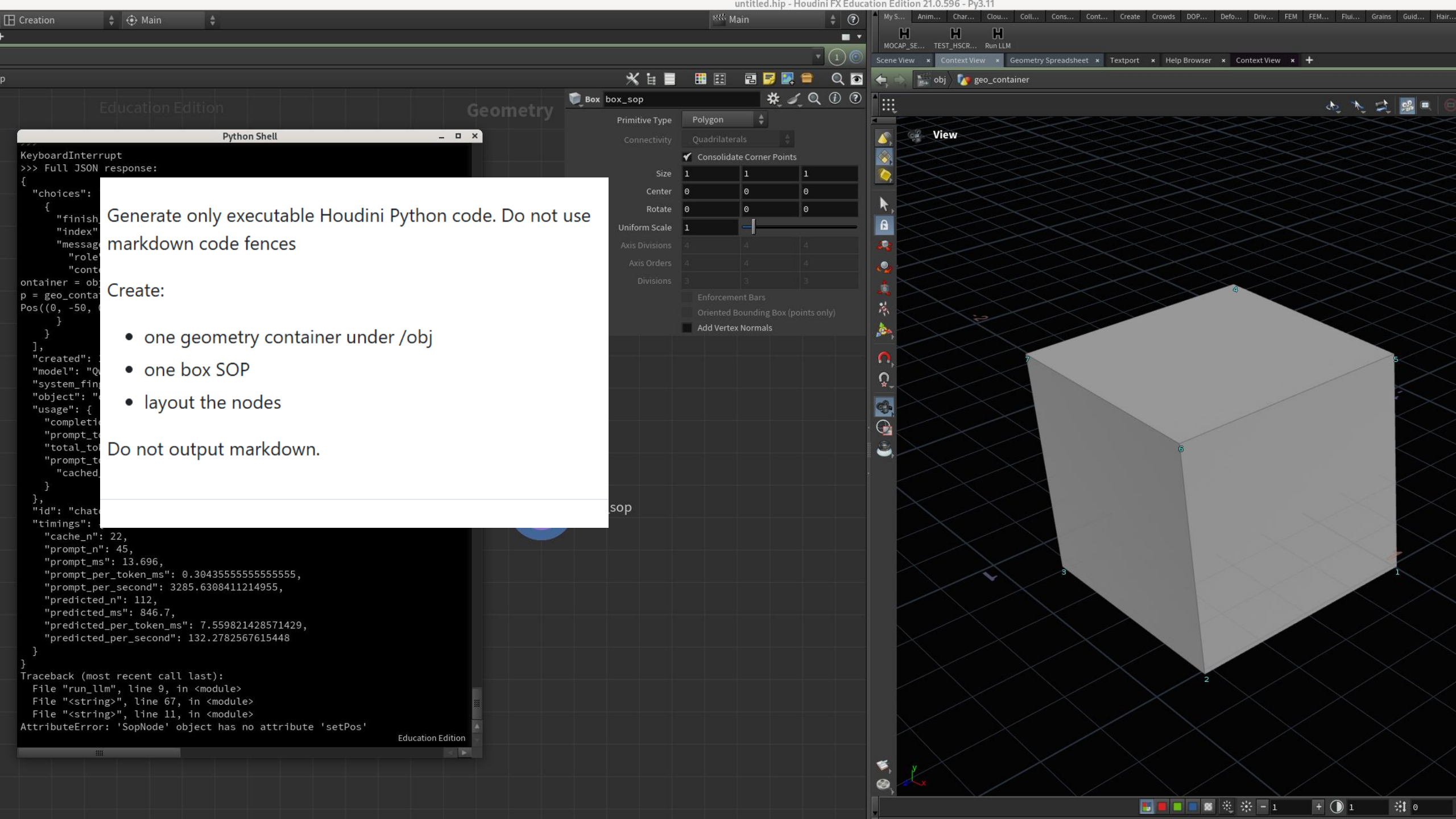
- Use natural language to support procedural modelling tasks
- Investigate what role AI can realistically play inside a PCG workflow

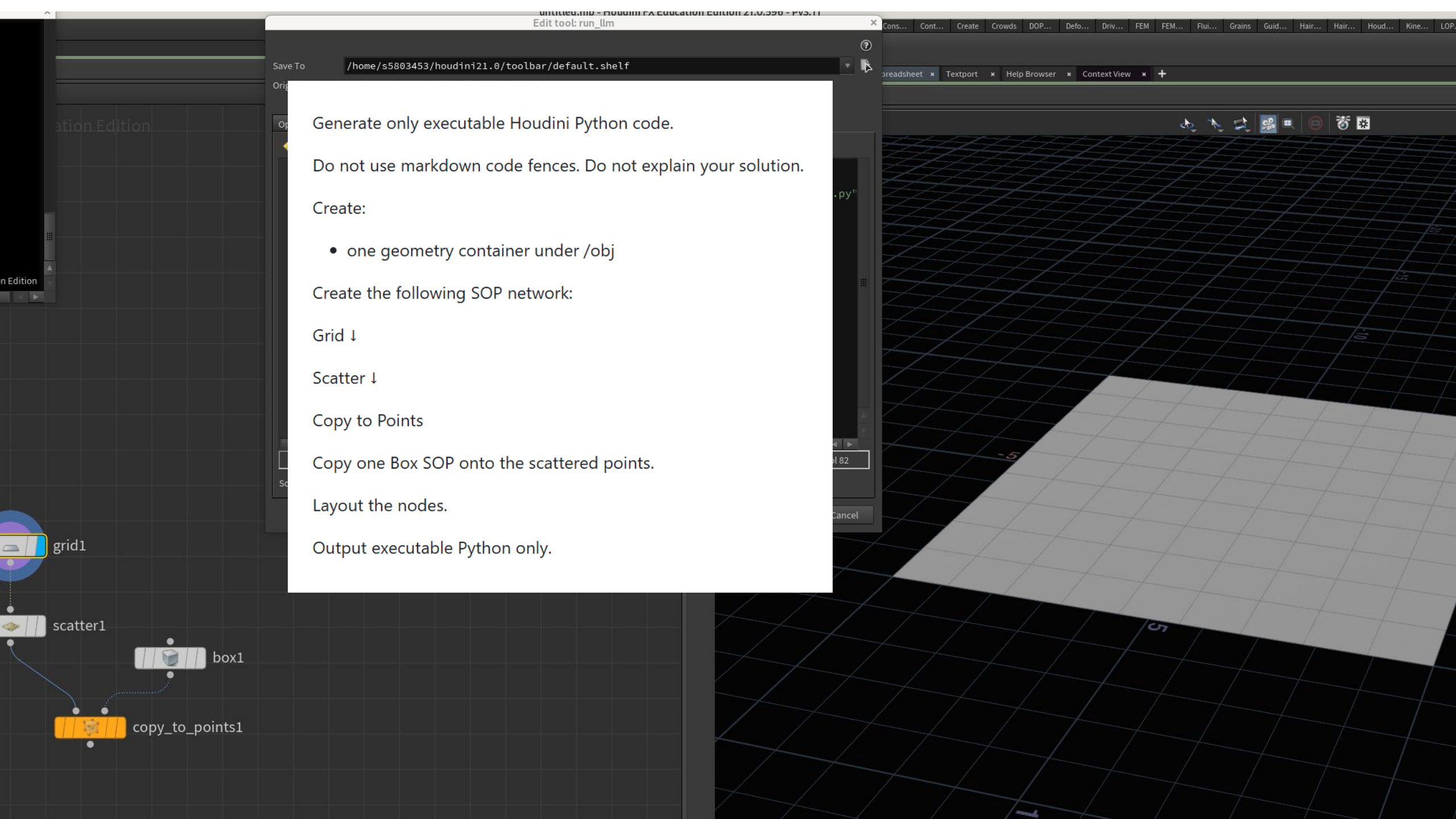
Initial Prototype

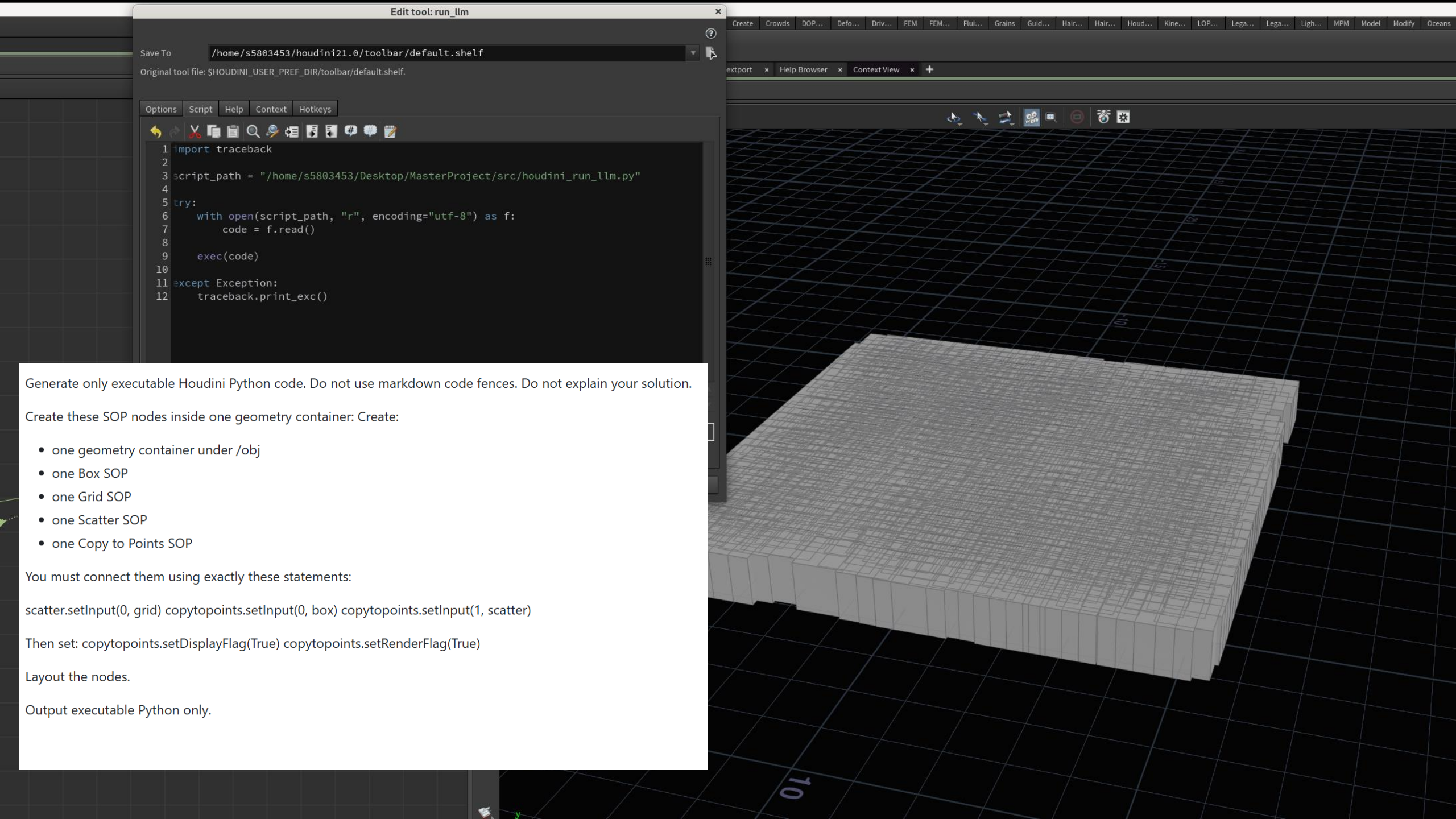
- Natural language → LLM-generated Houdini Python
- Local Qwen2.5-Coder-7B as the primary generator
- Simple tasks worked, but more complex tasks exposed limitations

Problem Identified During Development

- Failures were often related to missing Houdini-specific procedural knowledge
- Prompt engineering became increasingly solution-specific
- These observations motivated the introduction of RAG


untitled.hip - Houdini FX Education Edition 21.0.596 - Py3.11
Creation
Main
Main
Education Edition
Geometry
Python Shell
KeyboardInterrupt
>>> Full JSON response:
{
"choices":
{
"finish
"index"
"messag
"role
"cont
ontainer = ob
p = geo_conta
Pos((0, -50,
}
}
],
"created":
"model": "Q
"system_fin
"object": "
"usage": {
"completi
"prompt_t
"total_to
"prompt_t
"cached
}
},
"id": "chat
"timings":
"cache_n": 22,
"prompt_n": 45,
"prompt_ms": 13.696,
"prompt_per_token_ms": 0.30435555555555555,
"prompt_per_second": 3285.6308411214955,
"predicted_n": 112,
"predicted_ms": 846.7,
"predicted_per_token_ms": 7.559821428571429,
"predicted_per_second": 132.2782567615448
}
}
Traceback (most recent call last):
File "run_llm", line 9, in <module>
File "<string>", line 67, in <module>
File "<string>", line 11, in <module>
AttributeError: 'SopNode' object has no attribute 'setPos'
Education Edition
Generate only executable Houdini Python code. Do not use markdown code fences
Create:
one geometry container under /obj
one box SOP
layout the nodes
Do not output markdown.
Box box_sop
Primitive Type Polygon
Connectivity Quadrilaterals
Consolidate Corner Points
Size 1 1 1
Center 0 0 0
Rotate 0 0 0
Uniform Scale 1
Axis Divisions 4 4 4
Axis Orders 4 4 4
Divisions 3 3 3
Enforcement Bars
Oriented Bounding Box (points only)
Add Vertex Normals
sop
Scene View
Context View
Geometry Spreadsheet
Textport
Help Browser
Context View
obj
geo_container
View


Edit tool: run_llm
Save To
/home/s5803453/houdini21.0/toolbar/default.shelf
Generate only executable Houdini Python code.
Do not use markdown code fences. Do not explain your solution.
Create:
one geometry container under /obj
Create the following SOP network:
Grid ↓
Scatter ↓
Copy to Points
Copy one Box SOP onto the scattered points.
Layout the nodes.
Output executable Python only.
Cancel
grid1
scatter1
box1
copy_to_points1


Edit tool: run_llm
Save To
/home/s5803453/houdini21.0/toolbar/default.shelf
Original tool file: $HOUDINI_USER_PREF_DIR/toolbar/default.shelf.
Options
Script
Help
Context
Hotkeys
1 import traceback
2
3 script_path = "/home/s5803453/Desktop/MasterProject/src/houdini_run_llm.py"
4
5 try:
6 with open(script_path, "r", encoding="utf-8") as f:
7 code = f.read()
8
9 exec(code)
10
11 except Exception:
12 traceback.print_exc()
Help Browser
Context View
Generate only executable Houdini Python code. Do not use markdown code fences. Do not explain your solution.
Create these SOP nodes inside one geometry container: Create:
one geometry container under /obj
one Box SOP
one Grid SOP
one Scatter SOP
one Copy to Points SOP
You must connect them using exactly these statements:
scatter.setInput(0, grid) copytopoints.setInput(0, box) copytopoints.setInput(1, scatter)
Then set: copytopoints.setDisplayFlag(True) copytopoints.setRenderFlag(True)
Layout the nodes.
Output executable Python only.

From Direct Generation to RAG

Direct Generation

- Simple node tasks succeeded
- Copy to Points exposed Houdini-specific failures
- Missing nodes, reversed inputs and wrong display flags

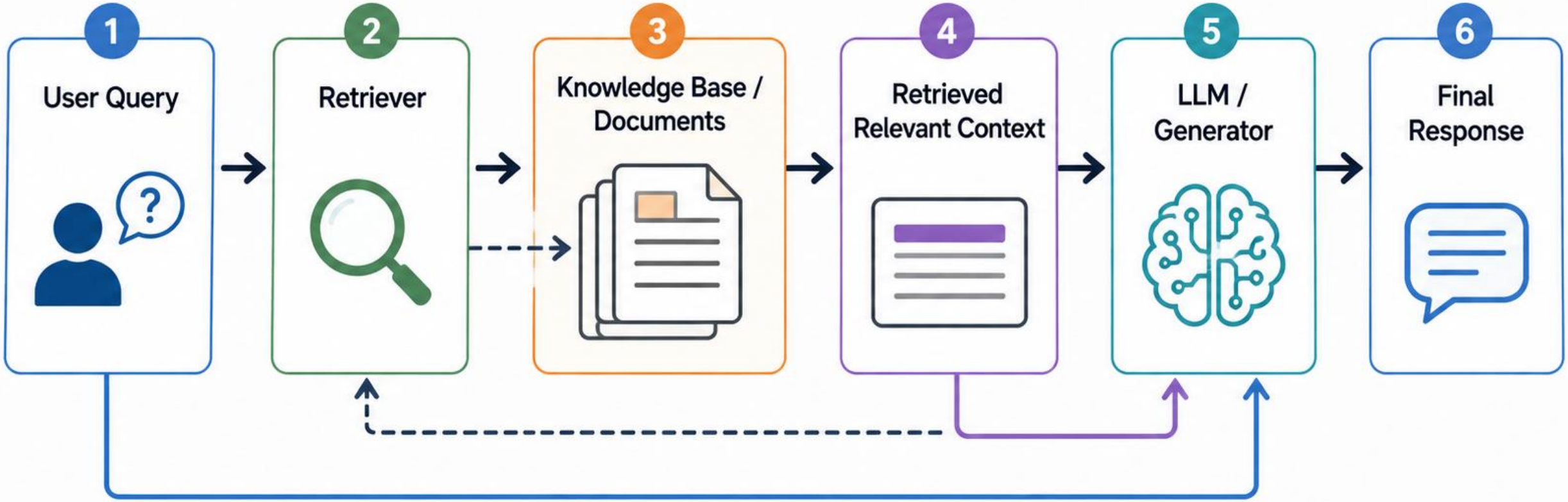
Prompt Engineering

- Increasingly explicit prompts eventually succeeded
- Connection order and final display node had to be specified
- The prompt became close to a procedural recipe

RAG

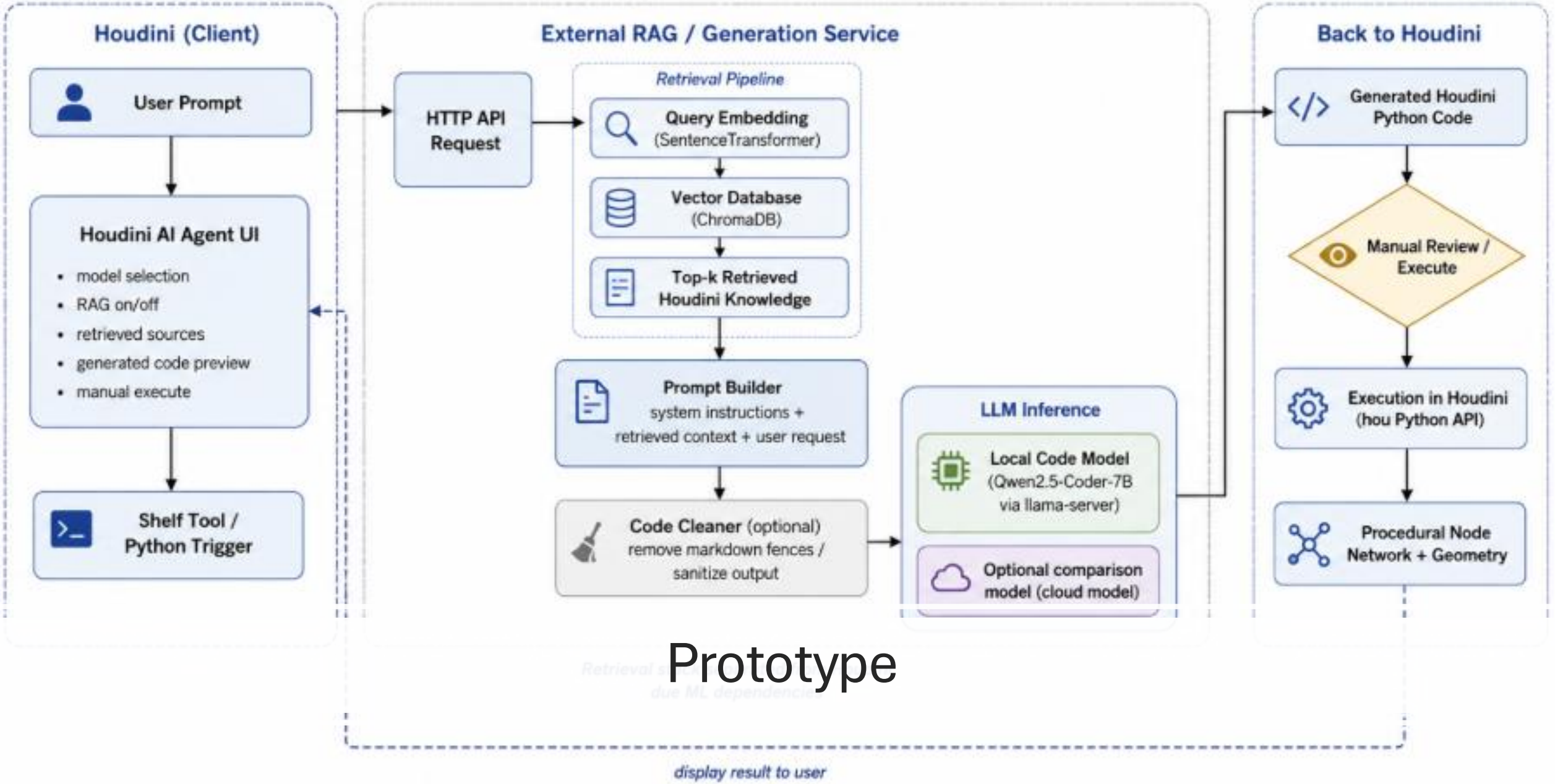
- Move reusable Houdini knowledge outside the user prompt
- Retrieve relevant procedural patterns when needed
- Reuse verified domain knowledge across tasks

Retrieval-Augmented Generation (RAG)



RAG adds external knowledge at inference time without retraining the model.

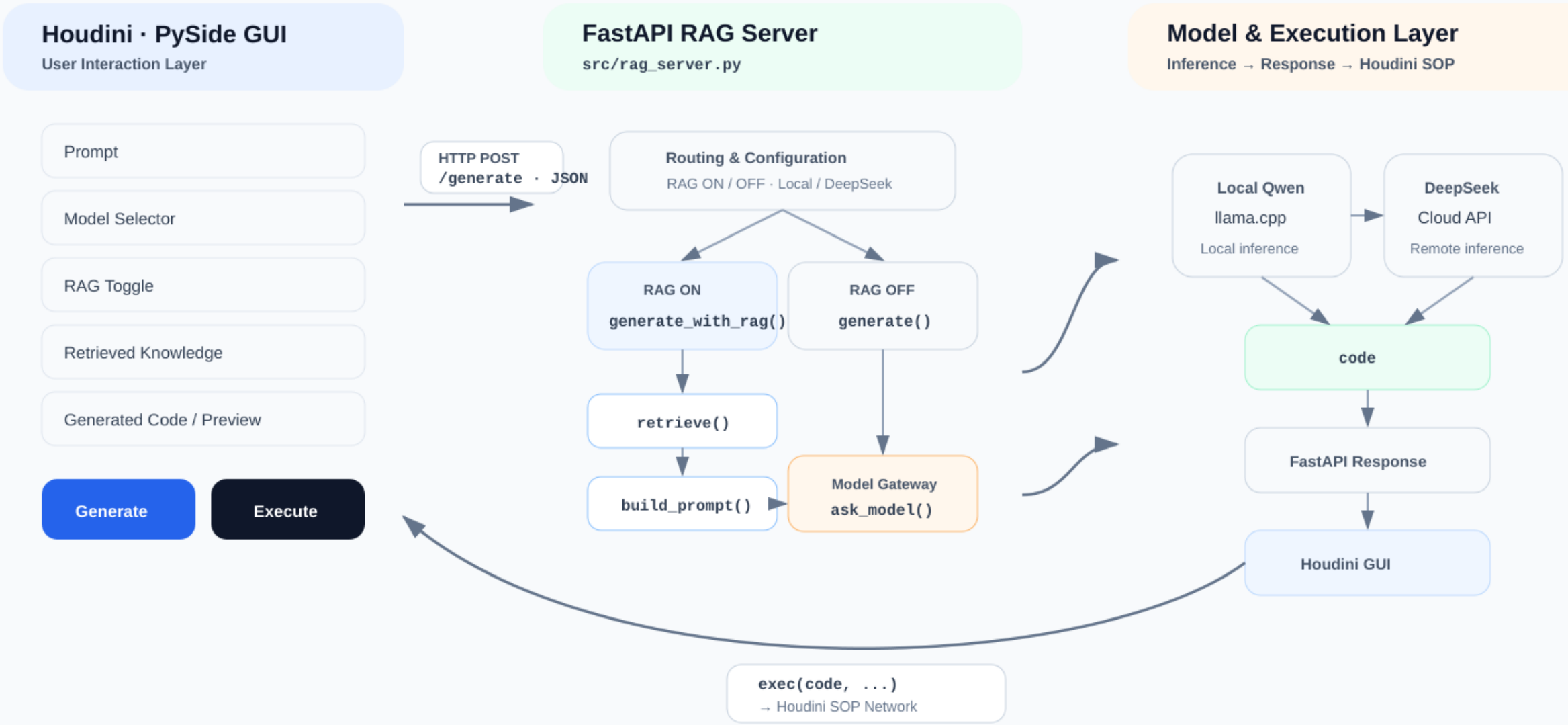
System Architecture of the Houdini AI Agent Prototype



Prototype

Houdini AI Code Generation · RAG Workflow

PySide GUI → FastAPI RAG Server → Retrieval / Model Routing → Code Execution



Architecture view · concise for presentation use

RAG Is Also an Engineering Problem

Knowledge Quality

- Tutorial examples contained irrelevant process and layout code
- Cleaner document structure improved retrieval ranking
- procedural_staircase_code_rag.md: 0.5122 → 0.4541

Query Design

- Full generation prompts could distort semantic retrieval
- Retrieval query and generation prompt were separated
- spiral_core.md: 0.5663 → 0.2631 with a concise semantic query

Chunking

- Complete spiral examples created long contexts and high token cost
- Split into core / railing / baluster knowledge

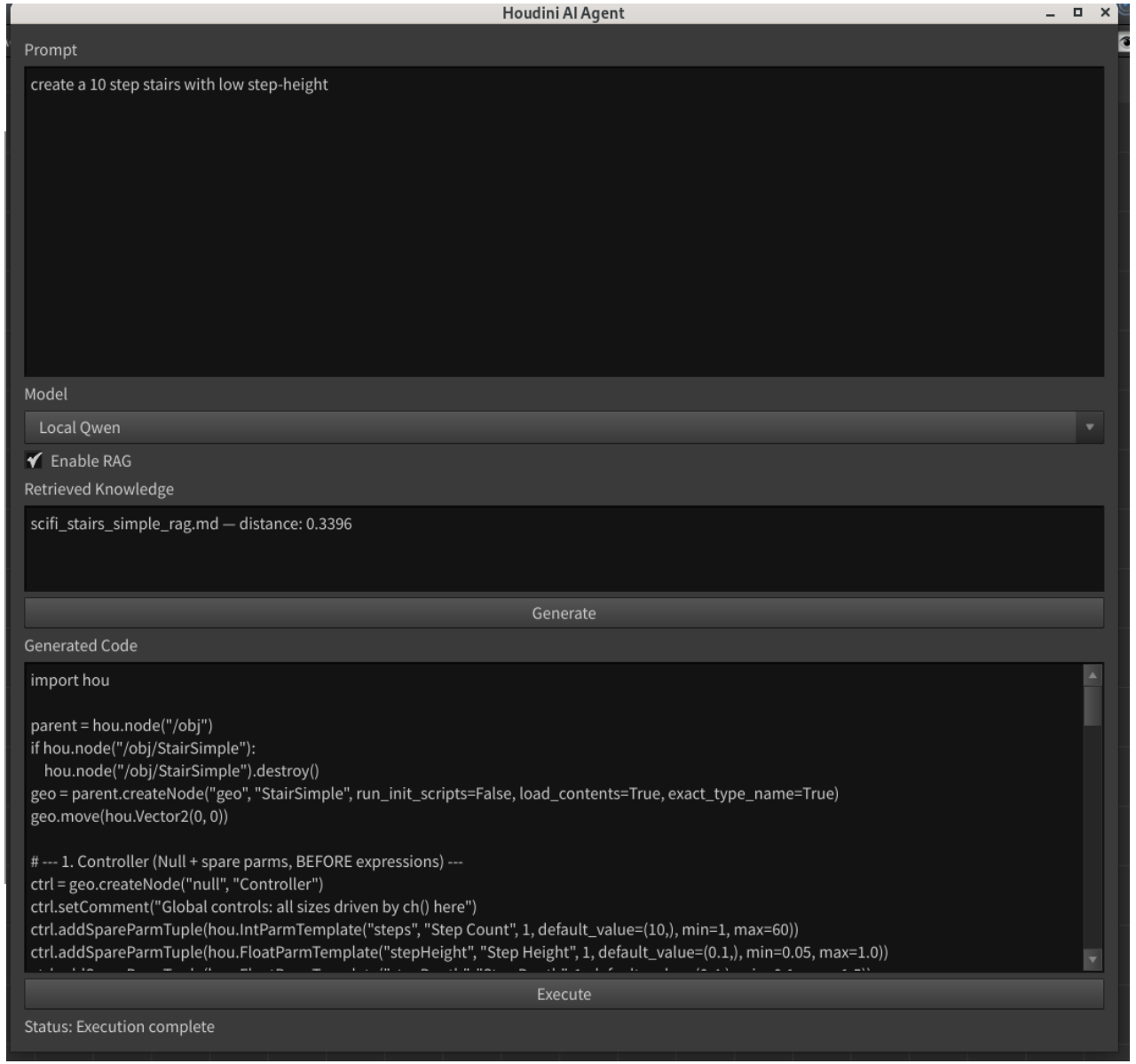
Top-k

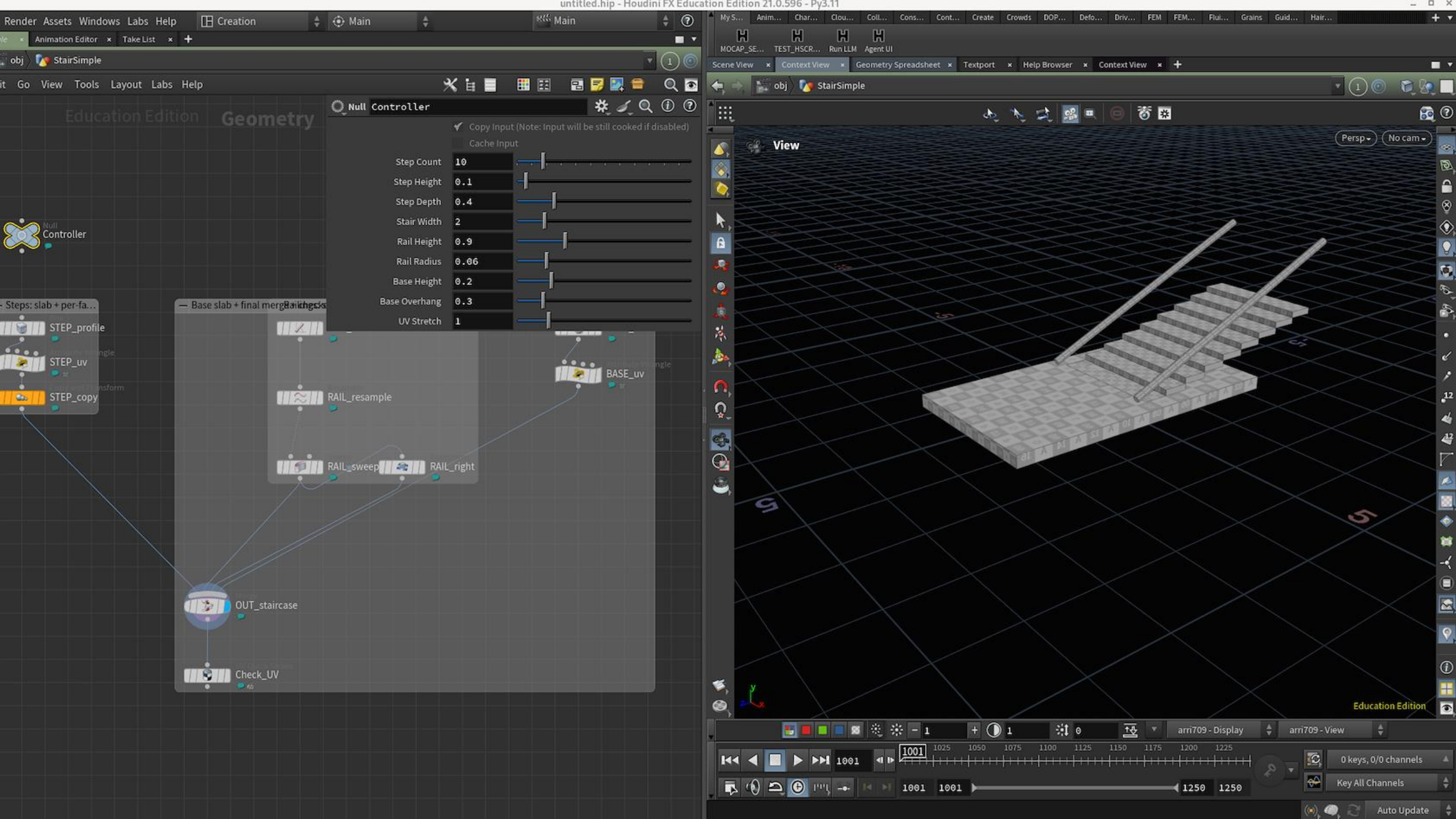
- Controls procedural coverage versus context cost
- top-k = 1: core structure
- top-k = 3: additional railing and baluster components

Key Findings

Natural language can modify existing procedural parameters

- Maps user intent to an existing procedural control
- The requested staircase step count and height was changed successfully

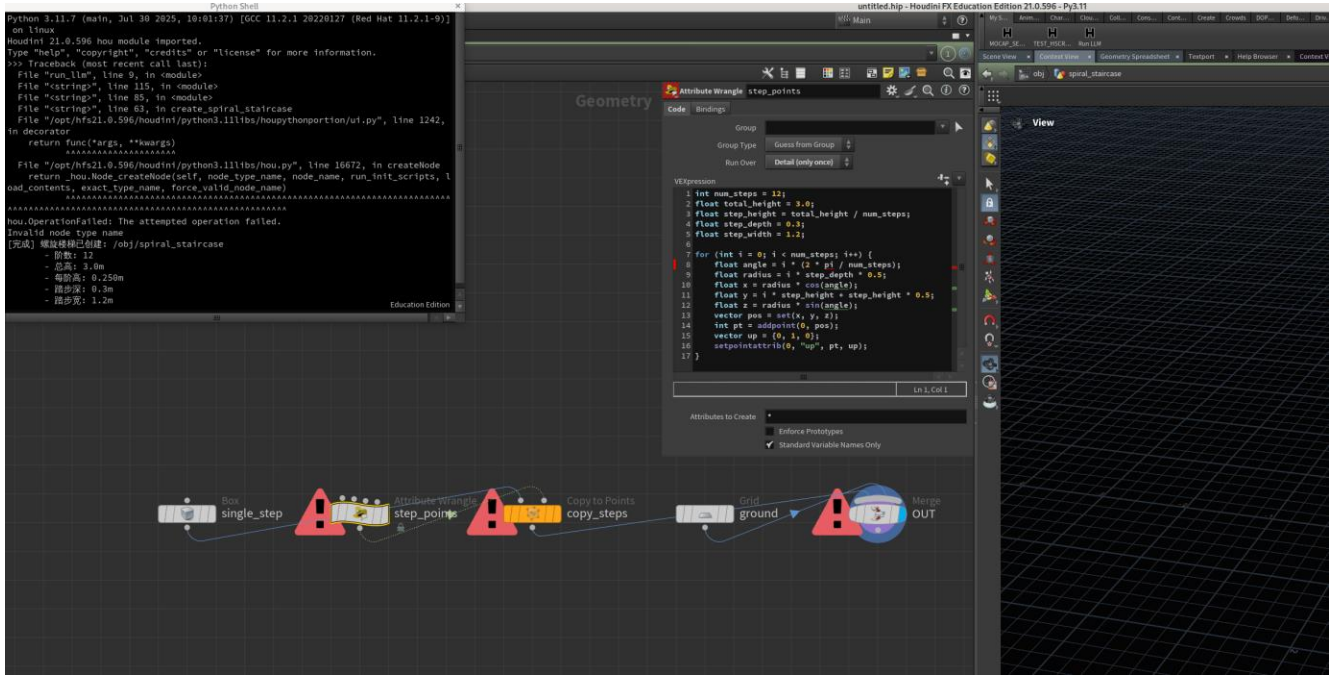



untitled.hip - Houdini FX Education Edition 21.0.596 - Py3.11
Render Assets Windows Labs Help
Creation
Main
Animation Editor
Take List
obj
StairSimple
Go View Tools Layout Labs Help
Education Edition
Geometry
Null Controller
Copy Input (Note: Input will be still cooked if disabled)
Cache Input
Step Count 10
Step Height 0.1
Step Depth 0.4
Stair Width 2
Rail Height 0.9
Rail Radius 0.06
Base Height 0.2
Base Overhang 0.3
UV Stretch 1
Steps: slab + per-fa...
STEP_profile
STEP_uv
STEP_copy
Base slab + final merge
RAIL_resample
RAIL_sweep
RAIL_right
BASE_uv
OUT_staircase
Check_UV
Scene View
Context View
Geometry Spreadsheet
Textport
Help Browser
MOCAP_SE...
TEST_HSCR...
Run LLM
Agent UI
View
Persp
No cam
Education Edition
arri709 - Display
arri709 - View
0 keys, 0/0 channels
Key All Channels
Auto Update

Key Findings

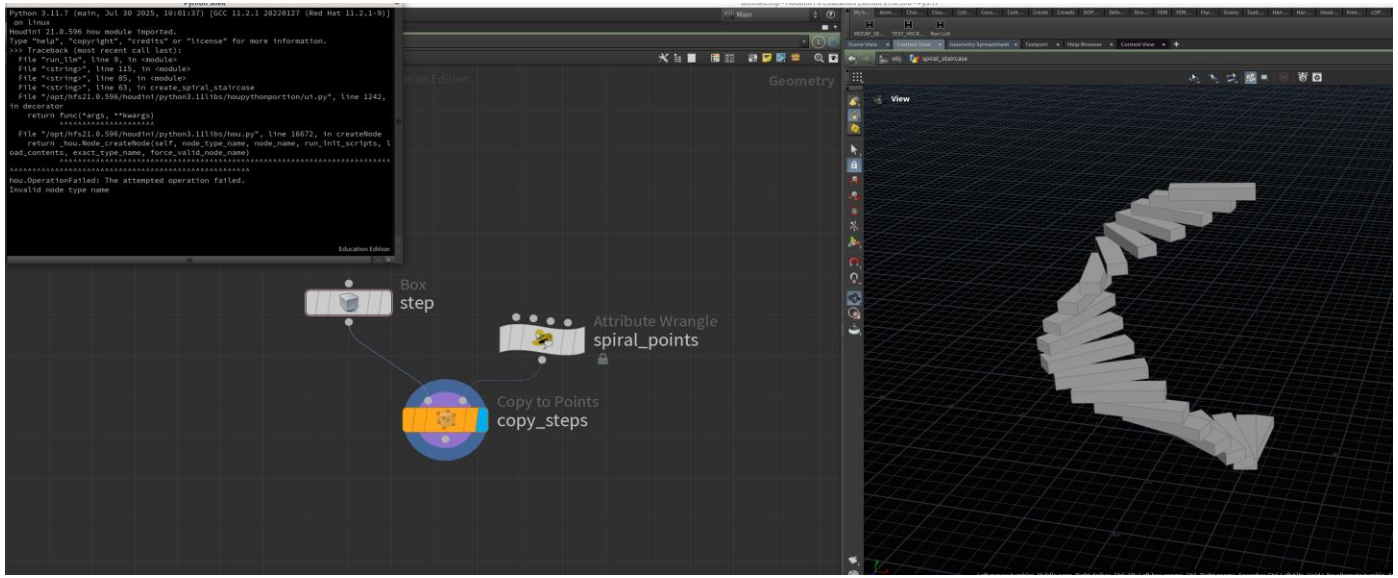
RAG can provide reusable Houdini-specific patterns

- Copy to Points benefited from retrieved connection knowledge
- Straight staircase reused an existing procedural structure



Complex composition still depends on model capability

- Spiral tasks required several spatial and orientation relationships
- Local and cloud models failed in different ways



Implications and Future Work

Implications

- AI can act as an interface to existing procedural systems
- Current prototype: natural language → generated Houdini code
- Observed: reuse and parameter modification of procedural patterns
- Potential direction: module selection and parameter configuration

Future Work

- Reranking retrieved candidates before final top-k selection
- Model-specific RAG configurations for small and large models
- Multi-agent decomposition: planning, retrieval, coding and verification
- Error-driven repair using Python / VEX / Houdini feedback
- Visual or geometric validation
- Structured PCG knowledge: modules, parameters and interfaces

Demo

Live Workflow

- Enter a natural-language request
- Enable RAG
- Inspect retrieved knowledge
- Generate Houdini Python
- Review and execute
- Inspect the SOP network and geometry

https://vimeo.com/1218752953?share=copy&fl=sv&fe=ci