

1 Environment Reconstruction

●●● agent trajectory

```
read config/config.json
read data/input/sample*.txt
write config/settings.yaml
write build/linux_build.sh
write process_data.py
exec ./build && pytest
write test_report.json
```

Stage 1: deterministic replay

Replayed workspace $\hat{E}_0$

📁 /app

- 📁 build/ linux_build.sh
- 📁 config/ config.json
- 📁 data/ input/sample1-3.txt
- 📄 process_data.py
- 📄 test_report.json

Stage 2: agentic completion

Completed workspace $\hat{E}$

📁 /app

- 📁 replayed
- + 📁 schemas
- + 📁 scripts/
- + 📁 src/
- + 📄 Makefile
- + 📄 pyproject.toml

Stage3: Environment Filter

Task-sufficient workspaces

2 Re-querying Methods

Intent recovery

Single-workspace

📁 /app

- 📁 src/
- 📁 scripts/
- 📁 schemas/

explore
agent

novel query
candidates

fix pyproject.toml packaging

repair invalid PEP dependency

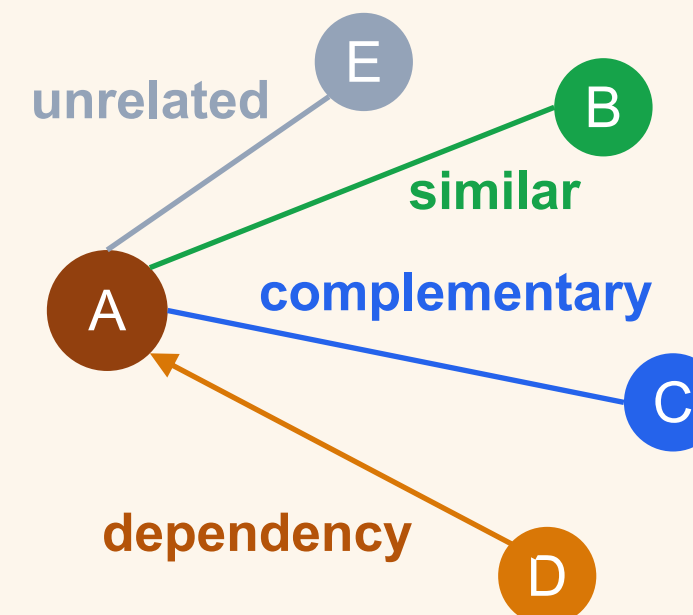
strengthen scripts/validate.py

Cross-workspace

1 workspace profiling

domain
language
frameworks
capabilities
summary
...

2 relation mining



3 cross-workspace task

create config/package.json
in the repo A as the
publishable package. Use
the B's /api/package.json to
model dist-based
conditional exports...

Multi-Round

initial query

agent
solves

👤 add RFC
3339 date-time
checks to the
validator.

👤 no, cache
parsed schemas
must be thread-
safe.

👤 schemas go
stale after edits.
fix invalidation.