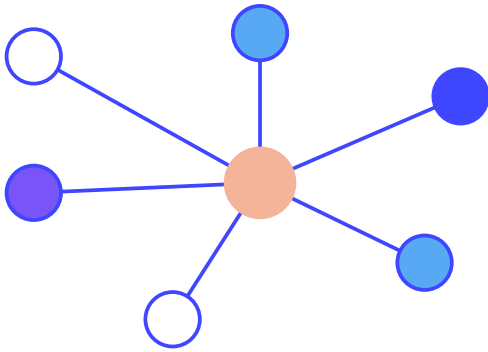




Open Math — Researcher Guide

Continuity in collaborative research

Open Math lets people work with their preferred coding agents while keeping permitted project context under the group's control. Personal work stays personal. Members deliberately contribute useful conclusions, and project agents can help organize or review bounded parts of the work. Models and chat products remain replaceable.

Pilot principle: preview first, share deliberately, keep reviewed code in Git.

Your first useful result in two minutes

One-time provider setup

The Code Agent uses the API account of the researcher running it. A ChatGPT or Codex subscription does not include OpenAI API credit. Create an API key in your own OpenAI Platform project and add a small prepaid budget, or use your own DeepInfra account. Never paste a key into chat, Git, or Slack.

For OpenAI, enter the secret without placing it in shell history:

```
install -d -m 0700 ~/openmathai/config
read -rsp 'OpenAI API key: ' OPENMATH_KEY; printf '\n'
umask 077
printf 'OPENAI_API_KEY=%s\nOPEN_MATH_OPENAI_MODEL=gpt-6-luna\n' \
"$OPENMATH_KEY" > ~/openmathai/config/openai.env
unset OPENMATH_KEY
chmod 0600 ~/openmathai/config/openai.env
```

For DeepInfra instead:

```
install -d -m 0700 ~/openmathai/config
read -rsp 'DeepInfra API key: ' OPENMATH_KEY; printf '\n'
umask 077
printf 'DEEPINFRA_API_KEY=%s\nDEEPINFRA_MODEL=openai/gpt-oss-120b\n' \
"$OPENMATH_KEY" > ~/openmathai/config/deepinfra.env
unset OPENMATH_KEY
chmod 0600 ~/openmathai/config/deepinfra.env
```

Select DeepInfra by adding `--provider deepinfra` to the review command. The preview always shows the effective provider, model, endpoint, conversation characters, selected-source bytes, and maximum output before anything is sent.

First review

1. Open your personal project checkout on the Open Math host:

```
cd ~/python/impact
```

2. Start a portable project conversation:

```
conversation_id="$(open-math conversation start \
--scope project:impact --title 'Loader review')"
```

3. Preview exactly what the Code Agent would receive:

```
open-math conversation review "$conversation_id" \
"Review the data-loading boundary." \
--path src/data/loader.py \
--path tests/test_data.py
```

4. Read the provider, model, exact endpoint, context size, file list, byte count, output cap, Git revision, and dirty-state disclosure. If they are right, repeat the command with `--confirm-share`.

5. Inspect the answer in the same conversation:

```
open-math conversation show "$conversation_id"
```

Nothing is sent during preview. The confirmed grant is read-only, single-use, expires after 30 minutes by default, and is bound to your Code Agent—not another member's worker.

Use your own provider credential during the pilot. The future Open Math gateway will provide budgeted access to hosted open-weight models without distributing a provider key to every researcher.

The simple mental model

- Your coding agent works in your checkout. Codex, GitHub Copilot, and OpenCode remain personal tools with personal logins.
- A portable conversation is the deliberately shared discussion held by Open Math. It can be reached from the CLI and, where mapped, Slack.
- Project agents are modest named helpers. The Project Librarian works with shared conversation context. The Code Agent reviews only specifically granted files for one request.
- Git remains the durable, reviewable home of source and project artifacts.

The current Code Agent cannot edit files, run shell commands, commit, push, or inspect a whole checkout. Its first job is safe review and navigation.

Connect your workspace

1. Create your own SSH key

Windows PowerShell:

```
ssh-keygen -t ed25519 \  
-f "$env:USERPROFILE\.ssh\openmath_ed25519" \  
-C "your-name@open-math.ai" \  
Get-Content "$env:USERPROFILE\.ssh\openmath_ed25519.pub"
```

macOS or Linux:

```
ssh-keygen -t ed25519 \  
-f ~/.ssh/openmath_ed25519 \  
-C "your-name@open-math.ai" \  
cat ~/.ssh/openmath_ed25519.pub
```

Send the administrator only the complete .pub line. Never send the private key. If you create a passphrase, keep it in your password manager; the passphrase protects the private key on your computer.

2. Connect with VS Code

Install Microsoft's Remote - SSH extension and add:

```
Host open-math  
  HostName 46.225.116.96  
  User your-name  
  IdentityFile ~/.ssh/openmath_ed25519  
  IdentitiesOnly yes
```

Run Remote-SSH: Connect to Host, choose open-math, and open ~/python/<project>. Project setup creates an independent checkout only for members assigned to that project.

3. Orca is optional

Orca can provide a modern editor-and-agents workspace. Add the same SSH target, choose Open remote folder, and open the existing ~/python/<project> checkout. Use Manual agent permissions during the pilot. Do not create a second Orca worktree and do not run a shared multi-user Orca server.

VS Code, Orca, or a plain terminal can all run the same Open Math workflow. Open Math is the context and permission layer, not another mandatory editor.

Choose your personal agent

- Codex: sign in with your own ChatGPT/Codex account.
- GitHub Copilot: sign in with your own GitHub entitlement.
- OpenCode: run open-math-opencode, then /connect and /models.

Never place one researcher's login or provider key in another user's home, Slack, Git, or the central Open Math database.

Everyday collaboration

Retrieve before repeating work

Project checkouts include Git-ignored MCP configuration for supported agents. Ask your agent to search recent project context before starting a repeated question. Retrieval is limited to the scope selected for that session.

Contribute a conclusion deliberately

Ask your agent: “Contribute this conclusion to the Impact project.” Review the exact text before confirming. A private transcript is not imported wholesale.

Continue one shared conversation

```
open-math conversation send "$conversation_id" \  
  "Record the assumptions we agreed." \  
open-math conversation show "$conversation_id"
```

The same Open Math-owned conversation can be surfaced through Slack and the CLI. A Codex- or Copilot-owned private thread remains in that product; selected conclusions can still be contributed to the project.

Invite the right project agent

Use the Project Librarian for questions about already shared context:

```
open-math conversation send "$conversation_id" \  
  "Summarize the retained assumptions." \  
  --agent project-librarian --wait
```

Use the Code Agent only when the question needs named local files. Its preflight displays every file and the destination provider before sharing.

Revoke an unused grant

```
open-math conversation revoke-grant GRANT_ID
```

Revocation prevents a pending or not-yet-dispatched read. It cannot retrieve text already sent to an external provider. This is why preview is the primary control.

What is shared—and what is not

Shared deliberately

- the visible message you add to a project conversation;
- selected conclusions you explicitly contribute;
- a project agent's visible answer;
- grant provenance: logical checkout, relative paths, counts, Git state, provider, model, endpoint, and output cap; and
- reviewed files and artifacts committed to the project Git repository.

Personal by default

- private Codex, Copilot, OpenCode, or Orca transcripts;
- provider and login credentials;
- unselected files and the absolute path to your home checkout;
- personal-scope context; and
- drafts not deliberately contributed or committed.

The Code Agent sends selected file text to the disclosed provider but does not archive that source snapshot in Open Math. Its visible answer and provenance remain in the shared conversation.

Safe Git habits

1. Start with `git status --short` and understand every existing change.
2. Keep one checkout per person. Never point two write-capable agents at the same working tree.
3. Ask an editing agent to show its diff before committing.
4. Keep credentials, datasets, checkpoints, and generated bulk output out of Git.
5. Commit only reviewed source, configuration, documentation, and deliberately retained small artifacts.
6. Do not use destructive reset commands to “clean up” work you do not own.

If the worktree is dirty, the Code Agent preview says so. That is not an error: it records that the reviewed snapshot differs from the last commit. Check the diff and make sure the selected files are the versions you intend to share.

A Git commit and an Open Math context contribution are separate actions. Either can be requested from an agent, but neither silently implies the other.

Student checklist

- ☐ My SSH private key and provider credentials stay private.
- ☐ I am working in my own `~/python/<project>` checkout.
- ☐ I know whether this session is personal or project scoped.
- ☐ I searched existing project context before duplicating work.
- ☐ I read the Code Agent preview before confirming.
- ☐ I inspected the Git diff before asking an agent to commit.
- ☐ I contributed only conclusions the group should retain.

Supervisor or project-owner checklist

- ☐ Membership contains only people working on this project.
- ☐ Every member has a separate Unix account, key, and checkout.
- ☐ Project-agent credentials are distinct from human credentials.
- ☐ Provider choices and costs are visible to researchers.
- ☐ Project agents have the smallest capability needed for the task.
- ☐ Shared conclusions are attributable and exportable.
- ☐ Reviewed source and artifacts remain in the canonical Git repository.

Troubleshooting

Permission denied (publickey)

Confirm the SSH username, IdentityFile, and that the administrator installed the matching public key for that Unix account. Test with:

```
ssh -i ~/.ssh/openmath_ed25519 your-name@46.225.116.96
```

The Code Agent says a path was rejected

Choose a normalized relative path to a regular UTF-8 text file. Hidden paths, links, credential-like files, binary files, special files, files above 64 KiB, and selections above 20 files or 256 KiB are refused. Select a smaller, purposeful set; do not rename secrets merely to bypass the check.

“Workspace changed after the grant preview”

The file contents, Git revision, dirty state, or byte count changed. Preview again and inspect the new disclosure before confirming.

The grant expired

Preview again. The short lifetime is intentional; do not extend it merely to avoid reviewing a changed checkout.

No Code Agent task was claimed

Check that you are a project member and that your private `~/openmathai/config/code-agent.env` exists. The worker is deliberately one-shot: the review command launches it for your request rather than leaving a shared daemon with access to everyone's files.

A provider call failed

The visible task records only a sanitized failure class. Check your private provider environment, account balance, model access, and network connection. Preview remains free; confirmation may incur provider cost.

Slack did not answer

Make sure the Open Math app is invited to that channel and mention it explicitly. Slack can work with the portable conversation, but it cannot create a local filesystem grant: use the review command from the checkout that owns the files.

Tips that save time

- Grant the two or three files that define an interface, not an entire tree.
- Include the relevant test beside implementation code.
- State the review objective and desired output: risks, invariants, or a test plan are more useful than “look at this.”
- Ask for relative filename citations in the answer.
- Keep project conversations focused; start a new one for a distinct research question.
- Contribute stable conclusions, not every exploratory turn.
- Treat secret scanning as a safety net, never as permission to share blindly.

Open Math is designed to future-proof collaborative research: the group can develop collective context alongside individual work while changing models, agents, or interfaces over time.