ELLM

How to Tune Generation Settings for Your Model

Learn how to fine-tune generation settings for local and self-hosted models to improve performance and accuracy. Adjust parameters like repetition, prompt size, and command timeouts to better match your model's behaviour.

AUTHOR
Paul Flanders

PUBLISHED
11 Sep 2026

DOCUMENT
InfoHub — Knowledge, news, and updates from EssingtonITS

How to Tune Generation Settings for Your Model

Paul Flanders · 11 Sep 2026

Estimated time: 5 minutes
Difficulty: Advanced

Why you'd use this

Local and self-hosted models vary a lot. One repeats lines, another ignores a long system prompt, a third takes several tool steps to find a file. A small group of settings lets you match the extension to the model in front of you: sampling and repetition controls, prompt size, and the safety caps on tool steps, commands and timeouts. Change one at a time and re-check with a familiar request.

Before you start

Permissions required:

- None. These are user or workspace settings.

You'll need:

- A request that shows the behaviour you want to change, so you can compare before and after.

Steps

1. Open VS Code settings and search for `ellmCode`.

2. For repeated or duplicated output, set `ellmCode.repetitionPenalty` to `1.1` if your gateway runs vLLM, or `ellmCode.frequencyPenalty` to a value between `0.3` and `0.5` for OpenAI-style servers. Both are off by default.
3. For a model that drifts from the action format, lower `ellmCode.temperature` (default `0.2`) and, for small or quantised models, tick **Compact prompt** in ⚙, which sends a much shorter system prompt while keeping the exact block formats.
4. For long searches that stop before the answer, raise `ellmCode.maxAgentSteps` (default 10). Lower it to make replies faster and cheaper.
5. For long test-fix loops, adjust `ellmCode.maxCommandRuns` (default 25 per task) and `ellmCode.commandTimeout` (default 120 seconds). Set `ellmCode.backgroundCommandAfter` (default 20 seconds) to control when a running command is moved to the background.
6. For big files, adjust `ellmCode.maxContextBytes` (default 60000), the largest slice of file content sent in one request.
7. Optionally turn on `ellmCode.reuseRetrieval` to have existing functions and types relevant to each request surfaced to the model, which reduces duplicated code in large repositories.
8. Untick **Surgical edits** only if a model consistently produces broken search-and-replace blocks; whole-file rewrites are slower and are held for review more often.

What you should see

Turn on **Debug logging** and open the debug trace while testing: each request line shows the effective settings, the intent and the token budget, so you can confirm a change took effect.

Troubleshooting

- **A setting seems ignored:** workspace settings override user settings. Check the Workspace tab.

- **The gateway rejects a penalty field:** some servers do not accept `repetition_penalty`. Set it back to `0`; it is only sent when positive.

Related guides

- How to Check and Manage the Context Window
- How to Run the Benchmark Suite
- How to Run Commands from the Chat