
The Steerability Challenge

Competition Guide

Submission deadline:	November 21, 2026
Workshop location:	NeurIPS 2026 (Paris)
Website:	https://steerability.github.io/competition/
Code repo:	https://github.com/generative-computing/steerability
Registration form:	https://tally.so/r/ZjBg10
Discord server:	https://discord.gg/jesm6FzXg
Document version:	v0.2 (Sept 11, 2026)

Live document

Please note that this is a live document and will be updated with additional detail and clarification throughout the competition as needed.

1 Welcome!

Thank you for your interest in participating in The Steerability Challenge! This document outlines how to get started with the toolkit, how to build a submission (and submit it), details of the competition timeline, and some general tips for creating strong submissions. For any questions you may have that are not addressed by this document, please join our Discord channel where we will be happy to help you out.

1.1 Motivation

Language models sometimes state things that contradict what they know to be true, i.e., they are dishonest. Given that language models are starting to be used in high stakes environments (e.g., medicine, law, etc.), the hazards associated with model dishonesty can be significant. Model steering can be an effective mechanism for improving honesty, however, how to best build these interventions is still an active area of research. There are various ways in which we can steer a model's behavior (e.g., prompting, activation steering, weight-based methods), but even the most effective methods still yield a considerable amount of observable dishonesty.¹ Furthermore, it appears that the mechanisms that influence a model's (dis)honesty are tightly coupled with those that dictate other (desirable) behaviors, thus steering for honesty may very well degrade a model's general capabilities.

1.2 Our Competition

Our competition focuses specifically on how to reduce model dishonesty while maintaining the model's general capabilities. To help provide structure to this challenge, and to facilitate standardized submissions and evaluations, the competition will be run using the *Steerability* toolkit.² One of the core abstractions in the toolkit is the notion of a *steering pipeline*, i.e., an ordered composition of steering methods (hereafter referred to as *controls*) that span modifications to the model's input (prompting), structure (weights), state (activations, attentions), and output (decoding). In addition to functionality for building steering pipelines, the toolkit also allows for running evals (via the *Inspect*³ library) and probing model internals.

Submissions will take the form of frozen steering pipelines (*.spipe* files) across our competition models (described in Section 3.1). Each steering pipeline contains the specification of the pipeline's structure (in *spipe.json*) as well as any trained artifacts the steering pipeline references (e.g., steering vectors). You can

¹Wang, Treutlein, Roger, Hubinger, and Marks, *Evaluating honesty and lie detection techniques on a diverse suite of dishonest models*, Alignment Science Blog, November 2025.

²<https://github.com/generative-computing/steerability>

³<https://inspect.aisi.org.uk>

additionally define custom controls as part of your submission (e.g., a novel activation steering method). Section 4.1 contains additional information regarding the specific format of submissions.

2 Getting Started

2.1 Create Your Hugging Face Repository for Submissions

Submissions will be made through a private Hugging Face repository owned by your team. The competition organizers will be given read-only access to this repository in order to download and evaluate your submissions. To keep submissions private from other participants, each team must create its own Hugging Face organization. A paid Hugging Face plan is not required for this process.

1. Create a Hugging Face account

If you do not already have a Hugging Face account, create one at <https://huggingface.co/join>.

2. Create a new Hugging Face organization for your team

Go to <https://huggingface.co/organizations/new> and create a new organization (you can choose *Community* as the organization type). Please choose an organization username (and full name) that matches, or clearly identifies, your team name. This will be needed when you register your team. Hugging Face may prompt you to upgrade to a paid Team or Enterprise plan; please select “skip for now”.

We recommend using this organization only for this competition, since the competition evaluator will be granted read access to repositories within the organization.

3. Add the competition evaluator to your organization

Open your organization’s Hugging Face page and navigate to *Settings* → *Members*.

Add our `steerability-challenge-evaluator` user as a Hugging Face user and assign this user the *Read* role. To do this, go to *Organization settings*, click *Members*, then *Add new member*.

Please *do not* assign the evaluator Contributor, Write, or Admin permissions. Read access is sufficient for us to download and evaluate your submission. Additionally, please ensure that you add the evaluator *user account* `steerability-challenge-evaluator`, not the competition organization `steerability-challenge-evals`.

4. Create a private repository for your submission

Go to <https://huggingface.co/new> and create a new model repository.

When creating the model repository:

- Set the *Owner* to the Hugging Face organization you created above.
- Name the repository as “submission” (under *Model name*).
- Set the repository visibility to *Private*.

Your submission repository should therefore have a URL of the form

`https://huggingface.co/<your-organization>/submission`.

Please *do not* create the submission repository under your personal Hugging Face account. The repository must be owned by your team’s organization so that the our evaluator can access it through its organization membership.

5. Note your repository URL

Once your repository has been configured and your submission files have been uploaded, note the repository URL (<https://huggingface.co/<your-organization>/submission>). This repository URL will be required when registering your team.

Please verify that:

- the repository is *Private*
- the repository is owned by your team’s *Hugging Face organization*, rather than by a personal account
- `steerability-challenge-evaluator` is a member of your organization with the *Read* role

2.2 Register Your Team

Once you have created your Hugging Face repository, the next step is registering your team via our Tally form (<https://tally.so/r/ZjBg10>). Teams can consist of anywhere from 1 to 5 people. We require that each participant on the team is associated with a verifiable identifier when registering (e.g., LinkedIn, Google Scholar, etc.). Please note that each participant is only allowed to be a member of one team (any violation of this rule will lead to disqualification of the entire team). Additionally note that there will be a short delay until we can validate your team's registration.

2.3 Install the Toolkit

Participants will build their submissions using the Steerability toolkit. The toolkit requires Python 3.12+ and is installed via the uv package manager (<https://docs.astral.sh/uv/getting-started/installation/>). Given that submissions are allowed to contain novel (custom) controls, we recommend that you fork the repo and install the toolkit with all development extras. To install, please follow:

1. Fork the repository at <https://github.com/generative-computing/steerability> and clone your fork to your working directory.
2. From the repository root, run `uv sync -extra all`. This creates a `.venv` (fetching a Python 3.12 interpreter if none is found), installs the toolkit in editable mode, and adds the `eval extra` (Inspect AI) together with the dev dependency group (pytest, pre-commit, and notebook functionality).
Since the install is editable, any custom controls that you add under `steerability/algorithms/` are discovered by the method registry without reinstalling.
3. Activate the environment with `source .venv/bin/activate` (on Windows, `.venv\Scripts\activate`), or prefix each command with `uv run`.
4. Authenticate with the Hugging Face Hub via `hf auth login` (or `export HF_TOKEN`). Gated checkpoints such as `meta-llama/*` additionally require approved access on the model's Hub page.

To verify your installation please run the following:

```
>>> from steerability.algorithms.core.registry import REGISTRY
>>> {k: len(v) for k, v in REGISTRY.items()}
{'input_control': 6, 'output_control': 14, 'state_control': 8, 'structural_control': 7}
```

Importing the registry loads the toolkit and its core dependencies and discovers every registered control (mergekit is skipped unless the merging extra is installed). Then run a small subset of the test suite that builds a steering pipeline and generates on a tiny model:

```
pytest tests/controls/test_user_prefix.py
```

The first run downloads the test models listed in `tests/utils/ci_models.yaml` from the Hugging Face Hub. All tests should pass; tests for devices unavailable on your machine are skipped automatically.

2.4 Starter Kit

As part of our competition, we have prepared a Starter Kit to help you in building a submission. The Starter Kit contains some baseline steering pipelines (with associated notebooks), lightweight evaluation data to sanity check your pipelines, and a submission checker script to allow you to validate that your submission is in the correct format before uploading. We will send the link for the Starter Kit to you once we have validated your team's registration.

In the meantime, we encourage participants to browse our example notebooks and documentation to get an understanding of the toolkit's functionality. If you are using agents to assist you, please point your agent to the `AGENTS.md` file in the repo's root directory.

! Compute requirements

Some of the steering interventions can be compute-heavy. Additionally, the competition will be focused on reasoning models in the range of 27–31B. We recommend that participants have access to a GPU with 80GB of VRAM.

3 The Task

The goal of the competition is to reduce model dishonesty while minimizing degradation of (off-target) general capabilities. To avoid overfitting, we do not reveal the specific measures of dishonesty or general capabilities, however, at a high-level you can assume that dishonesty covers lying and sycophantic behaviors whereas general capabilities measure knowledge, reasoning, problem solving, instruction following, and reliability. In general, your submissions will be scored on (i) the improvement it produces over the unsteered model on a collection of evaluations that elicit dishonest behavior, and (ii) the regressions it causes on a suite of general-capability benchmarks. These will be combined into a single composite score which will be posted on the leaderboard.

3.1 Competition Models

Since we are evaluating the methodology of your steering approach (not just a steered model), submissions will consist of steering pipelines, one per competition model below. We require that the pipelines only differ in terms of the trained artifacts that are referenced and not in the structure of the pipeline itself. The submission checker script provided in the Starter Kit validates this constraint. The intent is that you develop a *method*, not three unrelated artifacts that are overfit to each model.

We have selected the competition models based on their recency and overall performance. The competition models are:

1. google/gemma-4-31B-it
2. ibm-granite/granite-4.2-30b
3. Qwen/Qwen3.8-27B

Default inference parameters for the models are specified in the Starter Kit. Note that all of the above models are thinking models; thinking will be enabled but the actual thinking trace will not be used in the evaluations. Additionally note that sampling parameters are not part of a `.spipe`; if a submission requires extra sampling parameters they should be expressed through a control (e.g. `best_of_n`).

3.2 Competition Tracks

We have defined three broad tracks for the competition.

1. **Black-box:** As the name suggests, the black-box track consists of pipelines that neither require access to nor modify the model's internals (e.g., weights, activations, or logits).
2. **White-box:** Pipelines in the white-box track are allowed to both access and modify the model's states and logits.
3. **Open:** Pipelines in the open track are able to additionally modify the weights of the model.

Tracks nest in the sense that everything allowed in the black-box track is allowed in the white-box track, and everything allowed in the white-box track is allowed in the open track. If a submission spans two tracks, then the submission competes in the higher one. Given that only one submission per team is scored (at the evaluation intervals), a team's submission will only ever compete in a single track.

The identification of a track is categorized automatically based on what parts of the steering pipeline accesses and modifies. Special tracks in addition to the ones above (e.g., based on novelty, efficiency, etc.) may be created depending on the pool of submissions.

4 Preparing Your Submission

4.1 What is a Submission?

A submission is a directory containing:

- Steering pipelines (`.spipe` files): Three `.spipe` files (one per competition model).
- Custom controls (optional, `controls/`): Any custom controls as Python source code that the `.spipes` make use of. Note that if these controls require any steering artifacts (e.g., steering vectors) then these must be provided in the `.spipe` files.

The Starter Kit contains a script that verifies that the submission is properly formed before you submit it.

4.2 Uploading Your Submission

Submissions will be made using your Hugging Face repo specified in Section 2.1. You may upload files directly using the Hugging Face web interface or using Git. To upload using Git, first install the Hugging Face command-line tools:

```
pip install -U hf
```

Authenticate with your Hugging Face account and configure your credentials for Git:

```
hf auth login --add-to-git-credential
```

Then clone your submission repository:

```
git clone https://huggingface.co/<your-organization>/submission
cd submission
```

If your submission contains large files, such as model weights, you should also install Git-Xet as described in the Hugging Face documentation (<https://huggingface.co/docs/hub/repositories-getting-started>).

Lastly, copy the required submission files into the repository and push them:

```
git add .
git commit -m "Upload submission"
git push
```

You may update your repository as many times as you wish. We will download the current submission at specified evaluation intervals (weekly, midpoint, and final). Please ensure that your repository only ever contains one submission in order to avoid the wrong submission being evaluated.

4.3 Evaluation Intervals

Submissions will be evaluated weekly, at the competition midpoint, and at competition close. The depth of these evaluations will vary, as discussed below. Evaluation runs will use multiple seeds to capture variance information and to ensure reproducibility. All evaluations will be run on Hugging Face under bfloat16.

Weekly. The weekly evaluations will be lightweight “data-based” evaluations. The leaderboard will be updated using these scores.

Midpoint. The midpoint evaluations will be more comprehensive than the weekly evaluations. Additionally, as described in the Competition Timeline below, we will release trained model organisms to the participants after the midpoint.

Final. The final evaluation will be the most extensive evaluation, and will be authoritative in the sense that it will override previous evaluations.

5 Competition Timeline

Important dates:

1. *September 24, 2026:* Weekly evaluation of submissions begins with the results uploaded to the leaderboard. We will also commence regular office hours (cadence to be determined).
2. *October 19, 2026:* Competition midpoint. We will be releasing model organisms for the competition models at the midpoint. We will additionally start incorporating organism-based evals into the computation of the leaderboard scores.
3. *November 21, 2026:* Competition close. Participants will have until November 21 AoE to make a final submission upload.
4. *December 1, 2026:* Competition winners announced. Winners and other selected participants will be contacted by the organizers to prepare for their presentation at the workshop.

5. *December 13, 2026*: Competition workshop at NeurIPS 2026 in Paris.

The first two weeks of the competition are reserved for participants to get started with the toolkit. Throughout the competition, we will continue to develop and release our own baseline steering pipelines and share them with participants. These will serve as local optima and are intended to be improved upon by the participants. Note that given the short timeline between the winner announcement and the workshop date we do not require in-person attendance. In addition to the winners, the organizers will evaluate submissions for their general utility and select some additional (non-winning) teams to attend the workshop.

6 Building Steering Pipelines

6.1 General Guidance

One of the unique functionalities in the toolkit is the ability to compose multiple controls into a steering pipeline, i.e., an ordered set of controls either from the same control category (input, structure, state, and output) or across categories (e.g., prompting combined with activation steering). Anecdotally, we have found that combining controls can be quite effective due to the ability to exploit complementary strengths across different methods. We encourage participants to explore building pipelines that contain strictly more than one type of control.

Additionally, we encourage participants to make use of our Probe functionality to access the model’s hidden states. Hidden states can be very informative for certain behaviors, especially those that are more ingrained in a model’s behavior (e.g., dishonesty).

6.2 Other Tips

We’ve compiled some additional tips for steering. Note that we aim to expand on this list throughout the competition.

- For activation steering, the layer at which to steer matters a lot. Thus, we recommend either using the same layer as the baselines or conducting your own layer sweep to identify the best layer for steering input.
- The quality of the contrastive samples has a large effect on the resulting steering direction, so always ensure that you filter the data before building your method. In particular, we recommend checking that the model actually exhibits the labeled behavior in each sample (e.g., that a response labeled as dishonest is in fact dishonest), and discarding samples that are ambiguous or that differ from their contrastive counterpart in ways unrelated to honesty. Since the direction is typically extracted from a small number of samples, even a few mislabeled or noisy samples can noticeably shift it.
- The gemma-4 model family tends to have very large activations for some tokens (especially in the prompt). To avoid degrading the model’s capabilities during activation steering, we recommend either steering only the response tokens or implementing a gating mechanism that prevents steering tokens with very large activations.
- The type of contrastive samples matters more than their quantity. We’ve observed that it is possible to extract an activation steering vector from only 10 contrastive samples. What determines whether this direction is effective at restoring honesty and generalizes across many out-of-distribution settings is the scenario the samples are drawn from, i.e., the situation that elicits the dishonest behavior (for example, whether the model is explicitly instructed to lie, is under pressure to conceal information, or is agreeing with a user’s false belief), rather than how many samples are used.

7 Communication

If you have any questions about any of the above or encounter any issues, please reach out to us on our Discord server or via our Github page. We will additionally be hosting regular office hours (to be announced on Discord). We look forward to seeing what you build!