(new-model-tests)=

Writing Unit Tests

This page explains how to write unit tests to verify the implementation of your model.

Required Tests

These tests are necessary to get your PR merged into vLLM library.

Without them, the CI for your PR will fail.

Model loading

Include an example HuggingFace repository for your model in <gh-file:tests/models/registry.py>.

This enables a unit test that loads dummy weights to ensure that the model can be initialized in vLLM.

:::{important}

The list of models in each section should be maintained in alphabetical order.

:::

:::{tip}

If your model requires a development version of HF Transformers, you can set

`min_transformers_version` to skip the test in CI until the model is released.

:::

Optional Tests

These tests are optional to get your PR merged into vLLM library.

Passing these tests provides more confidence that your implementation is correct, and helps avoid future regressions.

Model correctness

These tests compare the model outputs of vLLM against [HF Transformers](https://github.com/huggingface/transformers). You can add new tests under the subdirectories of <gh-dir:tests/models>.

Generative models

For [generative models](#generative-models), there are two levels of correctness tests, as defined in <gh-file:tests/models/utils.py>:

- Exact correctness (`check_outputs_equal`): The text outputted by vLLM should exactly match the text outputted by HF.
- Logprobs similarity (`check_logprobs_close`): The logprobs outputted by vLLM should be in the top-k logprobs outputted by HF, and vice versa.

Pooling models

For [pooling models](#pooling-models), we simply check the cosine similarity, as defined in <gh-file:tests/models/embedding/utils.py>.

(mm-processing-tests)=

Multi-modal processing

Common tests

Adding your model to <gh-file:tests/models/multimodal/processing/test_common.py> verifies that the following input combinations result in the same outputs:

- Text + multi-modal data
- Tokens + multi-modal data
- Text + cached multi-modal data
- Tokens + cached multi-modal data

Model-specific tests

You can add a new file under <gh-dir:tests/models/multimodal/processing> to run tests that only apply to your model.

For example, if the HF processor for your model accepts user-specified keyword arguments, you can verify that the keyword arguments are being applied correctly, such as in <gh-file:tests/models/multimodal/processing/test_phi3v.py>.