```
import pytest
import torch

from swarm_models.open_dalle import OpenDalle


def test_init():
    od = OpenDalle()
    assert isinstance(od, OpenDalle)


def test_init_custom_model():
    od = OpenDalle(model_name="custom_model")
    assert od.pipeline.model_name == "custom_model"


def test_init_custom_dtype():
    od = OpenDalle(torch_dtype=torch.float32)
    assert od.pipeline.torch_dtype == torch.float32


def test_init_custom_device():
    od = OpenDalle(device="cpu")
    assert od.pipeline.device == "cpu"
```

```
def test_run():
    od = OpenDalle()
    result = od.run("A picture of a cat")
    assert isinstance(result, torch.Tensor)


def test_run_no_task():
    od = OpenDalle()
    with pytest.raises(ValueError, match="Task cannot be None"):
        od.run(None)


def test_run_non_string_task():
    od = OpenDalle()
    with pytest.raises(TypeError, match="Task must be a string"):
        od.run(123)


def test_run_empty_task():
    od = OpenDalle()
    with pytest.raises(ValueError, match="Task cannot be empty"):
        od.run("")


def test_run_custom_args():
    od = OpenDalle()
```

```
    result = od.run("A picture of a cat", custom_arg="custom_value")
    assert isinstance(result, torch.Tensor)



def test_run_error():
    od = OpenDalle()
    with pytest.raises(Exception):
        od.run("A picture of a cat", raise_error=True)
```