Mindspore.mint 接口测试报告

2025 年 1 月 15 日

目录

1 测试背景与目的 **3**

2 测试环境与方法 **3**

2.1 环境信息 3

2.2 测试方法 4

3 测试脚本与日志摘要 **4**

3.1 测试脚本概览 4

3.2 运行日志与结果 4

4 问题定位与建议 **5**

4.1 关于 `logaddexp` 内核 “unregistered“ 5

4.2 关于空张量输入 (zero dimension) 5

4.3 其余通过用例 6

5 结论 **6**

6 附录：pytest 命令与日志 **6**

6.1 执行命令 6

6.2 关键日志摘录 7

1 测试背景与目的

本次测试围绕 MindSpore（以及对应的 PyTorch）中的以下算子接口进行功能性与一致性测试：

- `sigmoid`, `sign`, `sin`, `sinc`, `sinh`
- `log1p`, `logaddexp`, `logical_and`, `logical_not`, `logical_or`

测试主要验证以下几点：

1) 不同 `dtype` 支持能力，以及随机输入对比 MindSpore 和 PyTorch 计算结果的一致性（允许误差 $<$ 1e-3）。
2) 固定 dtype 和形状下，是否能正确对齐输出；同时传入混乱参数或异常输入时，能否给出合理的报错信息。
3) 测试在使用这些接口构造简单神经网络时，MindSpore 与 PyTorch 在前向推理与反向梯度上的对齐程度。
4) 测试时的潜在问题记录，包括环境安装、CANN 或 memcpy 等错误、对齐性不足等。

2 测试环境与方法

2.1 环境信息

- 测试平台：macOS (Apple M1) 或同等
- Python 版本：3.11.11
- MindSpore 版本：2.4.10
- PyTorch 版本：2.5.1
- 测试工具：`pytest` (版本 8.3.4)

2.2 测试方法

分别为每个接口撰写独立的测试脚本文件 `test_xxx.py`。对于每个测试点，编写以下类别的测试函数：

a) 测试随机输入与不同 dtype

b) 测试固定 dtype、随机或固定输入值，对比两个框架输出结果

c) 测试固定 shape、混乱参数（string / bool 等）

d) 测试随机混乱输入，检查报错信息

e) 使用接口构造简单网络（或函数），比较前向推理对齐情况，以及反向传播的参数或输入梯度对齐

3 测试脚本与日志摘要

3.1 测试脚本概览

以下算子均有各自的 `test_xxx.py` 文件，包括：

- `test_sigmoid.py`, `test_sign.py`, `test_sin.py`, `test_sinc.py`, `test_sinh.py`
- `test_log1p.py`, `test_logaddexp.py`, `test_logical_and.py`, `test_logical_not.py`, `test_logical_or.py`

每个文件中，均包含了固定输入随机输入、异常输入、网络前向后向等测试场景，以及适度的日志输出，保证测试覆盖度。

3.2 运行日志与结果

运行 `pytest .` 后，测试结果日志显示共计 75 项测试用例，其中 65 项通过，10 项失败。

失败用例总结 ：

1. **logaddexp** 系列测试共 5 项失败：MindSpore 2.4.10 尚未注册 `logaddexp` 内核，导致 “kernel LogAddExp unregistered“。故在所有调用 `logaddexp` 的用例中出现 `RuntimeError`。
2. **logical_not 空张量用例**：测试 `logical_not` 在空张量时出现 “input_data can not contain zero dimension“ 的 `ValueError`。这说明 MindSpore 在 `Tensor` 构造时，不允许 size=0 的 `numpy` 数组。
3. **sigmoid 空张量用例**：同样出现 “input_data can not contain zero dimension“ 的错误，原因与上类似。
4. **sign 空张量用例**：出现同样的 zero-dimension 报错。
5. **sin 空张量用例**：依旧是 zero-dimension 错误。
6. **sinc 空张量用例**：同理是 zero-dimension 报错。

由此可见，MindSpore 目前对输入为 **size=0** 空张量尚无通用支持，会在构造 `Tensor` 阶段抛出异常；与 PyTorch 的做法（可构造 shape=(0,) 的张量并返回空输出）并不一致。

4 问题定位与建议

4.1 关于 `logaddexp` 内核 “unregistered“

MindSpore 在 2.4.10 版本中并未包含 `logaddexp` 算子的内核支持，这导致所有相关测试均在执行时失败。**建议**：

- 如需此算子，可等待后续官方版本集成，或者在内部自行实现并注册内核。
- 目前先行跳过该类测试或给出 `pytest.skip` 标志，以免测试全局 fail。

4.2 关于空张量输入 (zero dimension)

MindSpore 构建张量时若发现 `np.ndarray.size == 0`，默认抛出 `ValueError`，此与 PyTorch 行为不一致。**建议**：

- 如果测试需求不需要支持空张量，可移除或跳过该测试用例；若必须支持，可以向 MindSpore 反馈需求或查看源码能否添加空张量兼容性。
- 在测试前做检查,若 `arr.size == 0` 则 `pytest.skip("MindSpore 不支持 size=0 的输入")`。

4.3 其余通过用例

除上述 10 个失败用例外, 其余 65 个用例均成功通过, 说明在非空张量前提下, `sigmoid`, `sign`, `sin`, `sinc`, `sinh`, `log1p`, `logical_and`, `logical_not`, `logical_or` 均可较好地对齐 PyTorch，且在网络前向后向测试中也保持一致。

5 结论

1. **空张量兼容性**：MindSpore 不允许在 `Tensor` 构造时使用 size=0 的 `numpy` 数组；可以视为已知限制或后续版本特性。
2. **logaddexp 算子支持**：MindSpore 2.4.10 版本缺失该内核，导致相应用例全部失败。
3. **其余接口**：非空张量场景下，对比 PyTorch 一致性良好。

注：本测试报告仅针对上述 10 个接口在给定机器和软件栈上的测试结果，实际部署和更广泛场景下还需更多补充测试（如 GPU/Ascend 环境、多进程、多卡同步等）。

6 附录：pytest 命令与日志

6.1 执行命令

```
cd /path/to/Mindspore
pytest .
```

6.2 关键日志摘录

```
  pytest .

============================================================================
    test session starts
   ============================================================================

platform darwin -- Python 3.11.11, pytest-8.3.4, pluggy-1.5.0
rootdir: /Users/ioxyz/PycharmProjects/Mindspore
collected 75 items

13/test_log1p.py .......

   [  9%]
13/test_logaddexp.py .FF.FFF

   [ 18%]
13/test_logical_and.py .......

   [ 28%]
13/test_logical_not.py ....F..

   [ 37%]
13/test_logical_or.py .......

   [ 46%]
16/test_sigmoid.py .....F..

   [ 57%]
16/test_sign.py ....F...

   [ 68%]
16/test_sin.py ....F...

   [ 78%]
16/test_sinc.py .....F...

   [ 90%]
```

```
16/test_sinh.py .......

   [100%]

=================================================================================
    FAILURES
   =================================================================================


-----------------------------------------------------------------
   test_logaddexp_calculation_fixed_dtype
   -----------------------------------------------------------------

    def test_logaddexp_calculation_fixed_dtype():
        """
        (1b) 固定 dtype(float32)+固定输入
        """
        print_env_info()
        a_arr = np.array([0.0, 1.0, 2.0], dtype=np.float32)
        b_arr = np.array([2.0, 1.0, 0.0], dtype=np.float32)
        ms_out = logaddexp(Tensor(a_arr), Tensor(b_arr))
        torch_out = torch.logaddexp(torch.tensor(a_arr), torch.
           tensor(b_arr))
>       assert np.allclose(ms_out.asnumpy(), torch_out.detach().
   numpy(), atol=1e-4)

13/test_logaddexp.py:44:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:48: in fun
    arg = (stub.stub_sync(),) + arg[1:]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -

self = <[RuntimeError('The kernel LogAddExp unregistered.\n\n
   ------------------------------------------------\n- C++ Call
   ...ore/ops/kernel/common/pyboost/pyboost_utils.cc:569
```

```
   SelectKernel\n') raised in repr()] StubTensor object at 0
   x12a26edd0>

    def stub_sync(self):
        """sync real tensor."""
        if self.stub:
>           val = self.stub.get_value()
E           RuntimeError: The kernel LogAddExp unregistered.
E
E           ----------------------------------------------------
E           - C++ Call Stack: (For framework developers)
E           ----------------------------------------------------
E           mindspore/ops/kernel/common/pyboost/pyboost_utils.cc:569
    SelectKernel

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:159: RuntimeError
----------------------------------------------------------------------------
    Captured stdout call
   ----------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


-------------------------------------------------------------------
   test_logaddexp_calculation_random_dtype
   -------------------------------------------------------------------

    def test_logaddexp_calculation_random_dtype():
        """
        (1a) 随机输入, 不同 dtype
        """
        print_env_info()
        dmap = {
            mstype.float16: torch.float16,
            mstype.float32: torch.float32,
            mstype.float64: torch.float64
```

```
        }
        for ms_dt, torch_dt in dmap.items():
            a_arr = np.random.randn(4, 4).astype(mindspore.
               dtype_to_nptype(ms_dt))
            b_arr = np.random.randn(4, 4).astype(mindspore.
               dtype_to_nptype(ms_dt))
            ms_out = logaddexp(Tensor(a_arr, ms_dt), Tensor(b_arr,
               ms_dt))
            torch_out = torch.logaddexp(torch.tensor(a_arr, dtype=
               torch_dt),
                                        torch.tensor(b_arr, dtype=
                                           torch_dt))
>           assert np.allclose(ms_out.asnumpy(), torch_out.detach().
   numpy(), atol=1e-3)

13/test_logaddexp.py:62:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:48: in fun
    arg = (stub.stub_sync(),) + arg[1:]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -

self = <[RuntimeError('The kernel LogAddExp unregistered.\n\n
   ------------------------------------------------\n- C++ Call
   ...ore/ops/kernel/common/pyboost/pyboost_utils.cc:569
   SelectKernel\n') raised in repr()] StubTensor object at 0
   x12a256010>

    def stub_sync(self):
        """sync real tensor."""
        if self.stub:
>           val = self.stub.get_value()
E           RuntimeError: The kernel LogAddExp unregistered.
E
E           ------------------------------------------------
```

```
E           - C++ Call Stack: (For framework developers)
E           ----------------------------------------------------
E           mindspore/ops/kernel/common/pyboost/pyboost_utils.cc:569
    SelectKernel

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:159: RuntimeError
-----------------------------------------------------------------------------
    Captured stdout call
   -----------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


-------------------------------------------------------------------
   test_logaddexp_calculation_broadcast
   -------------------------------------------------------------------

    def test_logaddexp_calculation_broadcast():
        """
        扩展：广播形状
        """
        print_env_info()
        a_arr = np.random.randn(2, 1).astype(np.float32)
        b_arr = np.random.randn(1, 2).astype(np.float32)
        ms_out = logaddexp(Tensor(a_arr), Tensor(b_arr))
        torch_out = torch.logaddexp(torch.tensor(a_arr), torch.
           tensor(b_arr))
>       assert np.allclose(ms_out.asnumpy(), torch_out.detach().
   numpy(), atol=1e-3)

13/test_logaddexp.py:85:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:48: in fun
```

```
    arg = (stub.stub_sync(),) + arg[1:]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - -

self = <[RuntimeError('The kernel LogAddExp unregistered.\n\n
   ----------------------------------------------------\n- C++ Call
   ...ore/ops/kernel/common/pyboost/pyboost_utils.cc:569
   SelectKernel\n') raised in repr()] StubTensor object at 0
   x12a26edd0>

    def stub_sync(self):
        """sync real tensor."""
        if self.stub:
>           val = self.stub.get_value()
E           RuntimeError: The kernel LogAddExp unregistered.
E
E           ----------------------------------------------------
E           - C++ Call Stack: (For framework developers)
E           ----------------------------------------------------
E           mindspore/ops/kernel/common/pyboost/pyboost_utils.cc:569
    SelectKernel

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:159: RuntimeError
---------------------------------------------------------------------------
    Captured stdout call
   ---------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


------------------------------------------------------------
   test_logaddexp_nn_inference_compare_with_torch
   ------------------------------------------------------------

    def test_logaddexp_nn_inference_compare_with_torch():
```

```
        """
        (2b) 比较网络前向推理
        """
        print_env_info()
        net_ms = LogAddExpNetMindspore()
        a_arr = np.random.randn(2, 2).astype(np.float32)
        b_arr = np.random.randn(2, 2).astype(np.float32)
>       ms_out = net_ms(Tensor(a_arr), Tensor(b_arr)).asnumpy()

13/test_logaddexp.py:104:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:48: in fun
    arg = (stub.stub_sync(),) + arg[1:]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -

self = <[RuntimeError('The kernel LogAddExp unregistered.\n\n
   ------------------------------------------------------\n- C++ Call
   ...ore/ops/kernel/common/pyboost/pyboost_utils.cc:569
   SelectKernel\n') raised in repr()] StubTensor object at 0
   x12a18c950>

    def stub_sync(self):
        """sync real tensor."""
        if self.stub:
>           val = self.stub.get_value()
E           RuntimeError: The kernel LogAddExp unregistered.
E
E           ------------------------------------------------------
E           - C++ Call Stack: (For framework developers)
E           ------------------------------------------------------
E           mindspore/ops/kernel/common/pyboost/pyboost_utils.cc:569
    SelectKernel
```

```
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/_stub_tensor.py:159: RuntimeError
-------------------------------------------------------------------------
    Captured stdout call
   -------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


-------------------------------------------------------------------------
    test_logaddexp_function_grad
   -------------------------------------------------------------------------


    def test_logaddexp_function_grad():
        """
        (2c) 测试 logaddexp 对输入的梯度
        """
        print_env_info()
        a = Tensor(np.random.randn(2, 2).astype(np.float32))
        b = Tensor(np.random.randn(2, 2).astype(np.float32))
        a.requires_grad = True
        b.requires_grad = True

        def forward_fn(x, y):
            return logaddexp(x, y)

        grad_op = ops.GradOperation(get_all=True)(forward_fn)
>       grad_ms = grad_op(a, b)

13/test_logaddexp.py:122:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/ops/composite/base.py:394: in after_grad
    return grad_(fn)(*args, **kwargs)
```

```
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/api.py:188: in wrapper
    results = fn(*arg, **kwargs)
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/ops/composite/base.py:379: in after_grad
    run_args = self._pynative_forward_run(fn, grad_, weights, *args,
        **kwargs)
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/ops/composite/base.py:421: in _pynative_forward_run
    _pynative_executor.end_graph(fn, output, *args, **kwargs)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -

self = <mindspore.common.api._PyNativeExecutor object at 0x101c4fa10
   >, obj = <function test_logaddexp_function_grad.<locals>.
   forward_fn at 0x12f828ea0>
output = <[RuntimeError('The kernel LogAddExp unregistered.\n\n
   ------------------------------------------------\n- C++ Call
   ...ore/ops/kernel/common/pyboost/pyboost_utils.cc:569
   SelectKernel\n') raised in repr()] StubTensor object at 0
   x12f8bdc10>
args = (Tensor(shape=[2, 2], dtype=Float32, value=
[[-2.21340045e-01,  1.27937269e+00],
 [-1.33804226e+00, -5.67663372e-01]]), Tensor(shape=[2, 2], dtype=
    Float32, value=
[[-1.03579867e+00, -8.58256459e-01],
 [-1.60051131e+00,  4.18717802e-01]]))
kwargs = {}

    def end_graph(self, obj, output, *args, **kwargs):
        """
        Clean resources after building forward and backward graph.

        Args:
            obj (Function/Cell): The function or cell instance.
            output (Tensor/tuple/list): Function or cell output
               object.
            args (tuple): Function or cell input arguments.
```

```
            kwargs (dict): keyword arguments.

        Return:
            None.
        """
>       self._executor.end_graph(obj, output, *args, *(kwargs.values
   ()))
E       RuntimeError: The kernel LogAddExp unregistered.
E
E       ----------------------------------------------------
E       - C++ Call Stack: (For framework developers)
E       ----------------------------------------------------
E       mindspore/ops/kernel/common/pyboost/pyboost_utils.cc:569
   SelectKernel

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/api.py:1452: RuntimeError
-----------------------------------------------------------------------------
    Captured stdout call
   -----------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


-------------------------------------------------------------------
   test_logical_not_calculation_empty
   -------------------------------------------------------------------


    def test_logical_not_calculation_empty():
        """
        扩展：空张量
        """
        print_env_info()
        arr = np.array([], dtype=bool)
>       ms_out = logical_not(Tensor(arr))
```

```
13/test_logical_not.py:72:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:249: in __init__
    _check_tensor_input(input_data, dtype, shape, init)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -

input_data = array([], dtype=bool), dtype = None, shape = None, init
    = None

    def _check_tensor_input(input_data=None, dtype=None, shape=None,
        init=None):
        """Check the tensor input."""
        if input_data is not None and shape is not None:
            raise ValueError(f"When initializing a tensor with '
               input_data', 'shape' should be set to None."
                             f"But got shape: {shape}.")

        if init is not None and (shape is None or dtype is None):
            raise ValueError("init, dtype and shape must have values
                at the same time.")

        if input_data is not None:
            if isinstance(input_data, np.ndarray) and input_data.
               ndim >= 1 and input_data.size == 0:
>               raise ValueError("input_data can not contain zero
   dimension.")
E               ValueError: input_data can not contain zero
   dimension.

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:4959: ValueError
----------------------------------------------------------------------
    Captured stdout call
   ----------------------------------------------------------------------
```

```
===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================

-------------------------------------------------------------------------
    test_sigmoid_calculation_empty
   ------------------------------------------------------------------------


    def test_sigmoid_calculation_empty():
        """
        扩展：空张量输入
        """
        print_env_info()
        arr = np.array([], dtype=np.float32)
>       ms_out = sigmoid(Tensor(arr))

16/test_sigmoid.py:112:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:249: in __init__
    _check_tensor_input(input_data, dtype, shape, init)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - -

input_data = array([], dtype=float32), dtype = None, shape = None,
   init = None

    def _check_tensor_input(input_data=None, dtype=None, shape=None,
        init=None):
        """Check the tensor input."""
        if input_data is not None and shape is not None:
            raise ValueError(f"When initializing a tensor with '
               input_data', 'shape' should be set to None."
```

```
                            f"But got shape: {shape}.")

        if init is not None and (shape is None or dtype is None):
            raise ValueError("init, dtype and shape must have values
                at the same time.")

        if input_data is not None:
            if isinstance(input_data, np.ndarray) and input_data.
               ndim >= 1 and input_data.size == 0:
>               raise ValueError("input_data can not contain zero
   dimension.")
E               ValueError: input_data can not contain zero
   dimension.

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:4959: ValueError
----------------------------------------------------------------------
    Captured stdout call
   ----------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================

----------------------------------------------------------------------
    test_sign_calculation_empty
   ----------------------------------------------------------------------


    def test_sign_calculation_empty():
        """
        扩展：空张量
        """
        print_env_info()
        arr = np.array([], dtype=np.float32)
>       ms_res = sign(Tensor(arr))

16/test_sign.py:95:
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:249: in __init__
    _check_tensor_input(input_data, dtype, shape, init)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -

input_data = array([], dtype=float32), dtype = None, shape = None,
   init = None

    def _check_tensor_input(input_data=None, dtype=None, shape=None,
        init=None):
        """Check the tensor input."""
        if input_data is not None and shape is not None:
            raise ValueError(f"When initializing a tensor with '
               input_data', 'shape' should be set to None."
                             f"But got shape: {shape}.")

        if init is not None and (shape is None or dtype is None):
            raise ValueError("init, dtype and shape must have values
                at the same time.")

        if input_data is not None:
            if isinstance(input_data, np.ndarray) and input_data.
               ndim >= 1 and input_data.size == 0:
>               raise ValueError("input_data can not contain zero
   dimension.")
E               ValueError: input_data can not contain zero
   dimension.

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:4959: ValueError
-----------------------------------------------------------------------------
    Captured stdout call
   -----------------------------------------------------------------------------
```

```
===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================


------------------------------------------------------------------------
    test_sin_calculation_empty
   ------------------------------------------------------------------------


    def test_sin_calculation_empty():
        """
        扩展：空张量
        """
        print_env_info()
        arr = np.array([], dtype=np.float32)
>       ms_res = sin(Tensor(arr))

16/test_sin.py:89:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:249: in __init__
    _check_tensor_input(input_data, dtype, shape, init)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - -

input_data = array([], dtype=float32), dtype = None, shape = None,
   init = None

    def _check_tensor_input(input_data=None, dtype=None, shape=None,
        init=None):
        """Check the tensor input."""
        if input_data is not None and shape is not None:
            raise ValueError(f"When initializing a tensor with '
               input_data', 'shape' should be set to None."
                             f"But got shape: {shape}.")
```

```
        if init is not None and (shape is None or dtype is None):
            raise ValueError("init, dtype and shape must have values
                at the same time.")

        if input_data is not None:
            if isinstance(input_data, np.ndarray) and input_data.
               ndim >= 1 and input_data.size == 0:
>               raise ValueError("input_data can not contain zero
   dimension.")
E               ValueError: input_data can not contain zero
   dimension.

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:4959: ValueError
-------------------------------------------------------------------------
    Captured stdout call
   -------------------------------------------------------------------------

===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================

-------------------------------------------------------------------------
    test_sinc_calculation_empty
   -------------------------------------------------------------------------


    def test_sinc_calculation_empty():
        """
        扩展: 空张量 (0, )
        """
        print_env_info()
        arr = np.array([], dtype=np.float32)
>       ms_out = sinc(Tensor(arr))

16/test_sinc.py:92:
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -
/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:249: in __init__
    _check_tensor_input(input_data, dtype, shape, init)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - -

input_data = array([], dtype=float32), dtype = None, shape = None,
   init = None

    def _check_tensor_input(input_data=None, dtype=None, shape=None,
        init=None):
        """Check the tensor input."""
        if input_data is not None and shape is not None:
            raise ValueError(f"When initializing a tensor with '
               input_data', 'shape' should be set to None."
                             f"But got shape: {shape}.")

        if init is not None and (shape is None or dtype is None):
            raise ValueError("init, dtype and shape must have values
                at the same time.")

        if input_data is not None:
            if isinstance(input_data, np.ndarray) and input_data.
               ndim >= 1 and input_data.size == 0:
>               raise ValueError("input_data can not contain zero
   dimension.")
E               ValueError: input_data can not contain zero
   dimension.

/opt/homebrew/anaconda3/envs/Mindspore/lib/python3.11/site-packages/
   mindspore/common/tensor.py:4959: ValueError
----------------------------------------------------------------------------
    Captured stdout call
   ----------------------------------------------------------------------------
```

```
===== Environment Info =====
MindSpore version: 2.4.10
PyTorch version: 2.5.1
============================

=========================================================================
    short test summary info
   =========================================================================

FAILED 13/test_logaddexp.py::test_logaddexp_calculation_fixed_dtype
   - RuntimeError: The kernel LogAddExp unregistered.
FAILED 13/test_logaddexp.py::test_logaddexp_calculation_random_dtype
    - RuntimeError: The kernel LogAddExp unregistered.
FAILED 13/test_logaddexp.py::test_logaddexp_calculation_broadcast -
   RuntimeError: The kernel LogAddExp unregistered.
FAILED 13/test_logaddexp.py::
   test_logaddexp_nn_inference_compare_with_torch - RuntimeError:
   The kernel LogAddExp unregistered.
FAILED 13/test_logaddexp.py::test_logaddexp_function_grad -
   RuntimeError: The kernel LogAddExp unregistered.
FAILED 13/test_logical_not.py::test_logical_not_calculation_empty -
   ValueError: input_data can not contain zero dimension.
FAILED 16/test_sigmoid.py::test_sigmoid_calculation_empty -
   ValueError: input_data can not contain zero dimension.
FAILED 16/test_sign.py::test_sign_calculation_empty - ValueError:
   input_data can not contain zero dimension.
FAILED 16/test_sin.py::test_sin_calculation_empty - ValueError:
   input_data can not contain zero dimension.
FAILED 16/test_sinc.py::test_sinc_calculation_empty - ValueError:
   input_data can not contain zero dimension.
=====================================================================
    10 failed, 65 passed in 11.38s
   =====================================================================
```