```
Traceback (most recent call last)
in <module>:48

   45 |   train_loss = train(epoch)
   46 |   with torch.inference_mode():
   47 |   |   model.eval()
❱  48 |   |   train_acc = test(train_loader)
   49 |   |   test_acc = test(test_loader)
   50 |   |   print('Epoch: {:03d}, Train Loss: {:.7f}, '
   51 |   |   |   'Train Acc: {:.7f}, Test Acc: {:.7f}'.format(epoch, train_loss,

in test:35

   32 |   correct = 0
   33 |   for data in loader:
   34 |   |   data = data.to(device)
❱  35 |   |   output = model(data.x, data.edge_index, data.batch)
   36 |   |   pred = output.max(dim=1)[1]
   37 |   |   correct += pred.eq(data.y).sum().item()
   38 |   return correct / len(loader.dataset)

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1556 in _wrapped_call_impl

  1553 |   |   if self._compiled_call_impl is not None:
  1554 |   |   |   return self._compiled_call_impl(*args, **kwargs)  # type: ignore[misc]
  1555 |   |   else:
❱ 1556 |   |   |   return self._call_impl(*args, **kwargs)
  1557 |
  1558 |   def _call_impl(self, *args, **kwargs):
  1559 |   |   forward_call = (self._slow_forward if torch._C._get_tracing_state() else self.fo

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1565 in _call_impl

  1562 |   |   if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks
  1563 |   |   |   or _global_backward_pre_hooks or _global_backward_hooks
  1564 |   |   |   or _global_forward_hooks or _global_forward_pre_hooks):
❱ 1565 |   |   |   return forward_call(*args, **kwargs)
  1566 |
  1567 |   |   try:
  1568 |   |   |   result = None

in forward:35

   32 |   def forward(self, x, edge_index, batch):
   33 |   |   x = F.relu(self.conv1(x, edge_index))
   34 |   |   x = self.bn1(x)
❱  35 |   |   x = F.relu(self.conv2(x, edge_index))
   36 |   |   x = self.bn2(x)
   37 |   |   x = F.relu(self.conv3(x, edge_index))
   38 |   |   x = self.bn3(x)

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1556 in _wrapped_call_impl

  1553 |   |   if self._compiled_call_impl is not None:
  1554 |   |   |   return self._compiled_call_impl(*args, **kwargs)  # type: ignore[misc]
  1555 |   |   else:
❱ 1556 |   |   |   return self._call_impl(*args, **kwargs)
  1557 |
  1558 |   def _call_impl(self, *args, **kwargs):
  1559 |   |   forward_call = (self._slow_forward if torch._C._get_tracing_state() else self.fo

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1565 in _call_impl

  1562 |   |   if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks
  1563 |   |   |   or _global_backward_pre_hooks or _global_backward_hooks
  1564 |   |   |   or _global_forward_hooks or _global_forward_pre_hooks):
❱ 1565 |   |   |   return forward_call(*args, **kwargs)
  1566 |
  1567 |   |   try:
  1568 |   |   |   result = None

/usr/local/lib/python3.10/dist-packages/torch_geometric/nn/conv/gin_conv.py:84 in forward

   81 |   |   |   x = (x, x)
   82 |   |
   83 |   |   # propagate_type: (x: OptPairTensor)
❱  84 |   |   out = self.propagate(edge_index, x=x, size=size)
   85 |   |
   86 |   |   x_r = x[1]
   87 |   |   if x_r is not None:

/usr/local/lib/python3.10/dist-packages/torch_geometric/nn/conv/message_passing.py:541 in
propagate

   538 |   |   |   |   |   if res is not None:
   539 |   |   |   |   |   |   aggr_kwargs = res[0] if isinstance(res, tuple) else res
   540 |   |   |   |
❱  541 |   |   |   |   out = self.aggregate(out, **aggr_kwargs)
   542 |   |   |   |
   543 |   |   |   |   for hook in self._aggregate_forward_hooks.values():
   544 |   |   |   |   |   res = hook(self, (aggr_kwargs, ), out)
```

```
in aggregate:11

    8 |
    9 |    def aggregate(self, inputs, index):
   10 |    |    x = torch.sigmoid(inputs)
❱  11 |    |    x = self.laf(x, index)
   12 |    |    x = x.view((-1, self.dim * self.units))
   13 |    |    x = self.mlp(x)
   14 |    |    return x

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1556 in _wrapped_call_impl

 1553 |    |    if self._compiled_call_impl is not None:
 1554 |    |    |    return self._compiled_call_impl(*args, **kwargs)  # type: ignore[misc]
 1555 |    |    else:
❱ 1556 |    |    |    return self._call_impl(*args, **kwargs)
 1557 |
 1558 |    def _call_impl(self, *args, **kwargs):
 1559 |    |    forward_call = (self._slow_forward if torch._C._get_tracing_state() else self.fo

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:1565 in _call_impl

 1562 |    |    if not (self._backward_hooks or self._backward_pre_hooks or self._forward_hooks
 1563 |    |    |    |    or _global_backward_pre_hooks or _global_backward_hooks
 1564 |    |    |    |    or _global_forward_hooks or _global_forward_pre_hooks):
❱ 1565 |    |    |    return forward_call(*args, **kwargs)
 1566 |    |
 1567 |    |    try:
 1568 |    |    |    result = None

in forward:82

   79 |    |
   80 |    |    # scatter = scatter_add(exps, index.view(-1), dim=dim)
   81 |    |
❱  82 |    |    scatter_res = scatter(exps, index, dim=dim, reduce='sum')
   83 |    |
   84 |    |    # size = torch.tensor(exps.size())
   85 |    |    # size[dim] = index.max() + 1

/usr/local/lib/python3.10/dist-packages/torch_geometric/utils/_scatter.py:53 in scatter

   50 |    |    |    |    |    |    |    f"{src.dim() - 1} (got {dim})")
   51 |    |
   52 |    |    if dim_size is None:
❱  53 |    |    |    dim_size = int(index.max()) + 1 if index.numel() > 0 else 0
   54 |    |
   55 |    |    # For now, we maintain various different code paths, based on whether
   56 |    |    # the input requires gradients and whether it lays on the CPU/GPU.
```

RuntimeError: [Rank:0] FATAL ERROR :: MODULE:PT_BRIDGE Exception in Lowering thread...
synStatus=1 [Invalid argument] Node reshape failed.
Exception raised from add_node at /npu-stack/pytorch-integration/backend/synapse_helpers/graph.cpp:463 (most recent call first):
frame #0: c10::Error::Error(c10::SourceLocation, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >) + 0xac (0x7f0f6c64732c in /usr/local/lib/python3.10/dist-packages/torch/lib/libc10.so)
frame #1: c10::detail::torchCheckFail(char const*, char const*, unsigned int, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&) + 0xf3 (0x7f0f6c5f061f in /usr/local/lib/python3.10/dist-packages/torch/lib/libc10.so)
frame #2: synapse_helpers::graph::add_node(std::vector<internalTensor*, std::allocator<internalTensor*> >&&, std::vector<internalTensor*, std::allocator<internalTensor*> >&&, void*, unsigned int, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, unsigned long*, char const**, char const**, bool, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&) + 0x21f3 (0x7f0e9727b673 in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #3: habana::OpBackend::BuildNode(habana::OpBackend*, synapse_helpers::graph&, habana::NodeAttr&&) + 0x881 (0x7f0e974a1a11 in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #4: habana::OpBackend::BuildReshape(habana::OpBackend*, synapse_helpers::graph&, internalTensor*, c10::ArrayRef<long>, c10::ScalarType, std::optional<int>, std::optional<unsigned int>) + 0x2fa (0x7f0e974a891a in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #5: <unknown function> + 0x126349d (0x7f0e9733e49d in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #6: habana::BatchNormOpBackend::AddNode(synapse_helpers::graph&, std::vector<c10::IValue, std::allocator<c10::IValue> > const&) + 0x228 (0x7f0e97343b18 in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #7: habana::HabanaLaunchOpPT::BuildSynapseGraph(std::shared_ptr<synapse_helpers::graph>&, habana::SynBuildCache&, bool) + 0x24f1 (0x7f0e97aabda1 in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #8: habana::HabanaLaunchOpPT::run(std::vector<c10::IValue, std::allocator<c10::IValue> >&, std::optional<std::vector<at::Tensor, std::allocator<at::Tensor> > >, std::optional<std::vector<std::vector<long, std::allocator<long> >, std::allocator<std::vector<long, std::allocator<long> > > > >, bool, habana::HabanaLaunchOpPipeline::PipelineCallBase&) + 0x2ee8 (0x7f0e97aca728 in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #9: habana::HabanaLaunchOpPipeline::LoweringTask(std::unique_ptr<habana::HabanaLaunchOpPT, std::default_delete<habana::HabanaLaunchOpPT> >&&, std::vector<c10::IValue, std::allocator<c10::IValue> >&, std::optional<std::vector<at::Tensor, std::allocator<at::Tensor> > >, std::optional<std::vector<std::vector<long, std::allocator<long> >, std::allocator<std::vector<long, std::allocator<long> > > > >) + 0xda (0x7f0e97acb88a in /usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch_backend.so)
frame #10: habana::eager::EagerExec::launch() + 0xd22 (0x7f0e98cd3362 in

```
/usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch2_plugin.so)
frame #11: habana::eager::EagerLoweringTask(c10::Symbol, std::vector<c10::IValue, std::allocator<c10::IValue> >&&,
habana::eager::OutputSpecsOrTensors&&, habana::eager::EagerOpMetaData&&) + 0x1c5 (0x7f0e98c02545 in
/usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch2_plugin.so)
frame #12: habana_helpers::ThreadPoolBase<habana_helpers::BlockingQueue,
habana_helpers::move_only_function_void>::executePendingTask(habana_helpers::move_only_function_void&&) + 0x72
(0x7f0e990648c2 in
/usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch2_plugin.so)
frame #13: habana_helpers::ThreadPoolBase<habana_helpers::BlockingQueue,
habana_helpers::move_only_function_void>::main_loop() + 0xbe (0x7f0e990650be in
/usr/local/lib/python3.10/dist-packages/habana_frameworks/torch/lib/libhabana_pytorch2_plugin.so)
frame #14: <unknown function> + 0xdc253 (0x7f0f77e29253 in /lib/x86_64-linux-gnu/libstdc++.so.6)
frame #15: <unknown function> + 0x94ac3 (0x7f0f787b1ac3 in /lib/x86_64-linux-gnu/libc.so.6)
frame #16: clone + 0x44 (0x7f0f78842a04 in /lib/x86_64-linux-gnu/libc.so.6)
```