```
Traceback (most recent call last)
in <module>:9

   6 best_val_acc = test_acc = 0
   7 for epoch in trange(1, 151):
   8 |   train_loss = train(epoch)
❱  9 |   val_acc = test(val_loader)
  10 |   if val_acc > best_val_acc:
  11 |   |   test_acc = test(test_loader)
  12 |   |   best_val_acc = val_acc

/usr/local/lib/python3.10/dist-packages/torch/utils/_contextlib.py:116 in decorate_context

  113 |   @functools.wraps(func)
  114 |   def decorate_context(*args, **kwargs):
  115 |   |   with ctx_factory():
❱ 116 |   |   |   return func(*args, **kwargs)
  117 |
  118 |   return decorate_context
  119

in test:28

  25 |
  26 |   for data in loader:
  27 |   |   data = data.to(device)
❱ 28 |   |   pred, _, _ = model(data.x, data.adj, data.mask)
  29 |   |   print(pred)
  30 |   |   pred = pred.max(dim=1)[1]
  31 |   |   correct += pred.eq(data.y.view(-1)).sum().item()

/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:69

  66 |   |   s = self.gnn1_pool(x, adj, mask)
  67 |   |   # print(f"S: {s.shape}")
  68 |   |
❱ 69 |   |   x = self.gnn1_embed(x, adj, mask)
  70 |   |   # print(f"x: {s.shape}")
  71 |   |
  72 |   |   x, adj, l1, e1 = dense_diff_pool(x, adj, s, mask)

/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:31

  28 |   |   for step in range(len(self.convs)):
  29 |   |   |   # print(f"Step {step}")
  30 |   |   |   # print(self.convs[step])
❱ 31 |   |   |   x = F.relu(self.convs[step](x, adj, mask))
  32 |   |   |   # print(f"after conv, x: {x.shape}")
  33 |   |   |   # print(self.bns[step])
  34 |   |   |   x = x.permute(0, 2, 1)
```

```
/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/dense/dense_gcn_conv.py:67 in forward

   64 │   │   if add_loop:
   65 │   │   │   adj = adj.clone()
   66 │   │   │   idx = torch.arange(N, dtype=torch.long, device=adj.device)
❱  67 │   │   │   adj[:, idx, idx] = 1 if not self.improved else 2
   68 │   │
   69 │   │   out = self.lin(x)
   70 │   │   deg_inv_sqrt = adj.sum(dim=-1).clamp(min=1).pow(-0.5)

RuntimeError: synStatus=1 [Invalid argument] Node reshape  failed.
```