2. sgmnet(匹配部分)Atlas 200DK A2部署

对于pytorch实现的网络，ATLAS部署需要先转onnx再转om，最后在开发板推理。

2.1 `pytorch的.pth转onnx`

```
        torch.onnx.export(
        self.model,  # The model to export
        (x1_in,x2_in,desc1_in,desc2_in),  # Model input
        onnx_file_path,  # The path where the ONNX file will be saved
        input_names=['x1_in', 'x2_in', 'desc1_in', 'desc2_in'],  # Input tensor
names
        output_names=['p'],  # Output tensor names
        opset_version=12  # ONNX opset version
    )
    print(f"Model successfully exported to {onnx_file_path}")
```

2.1.1 问题1：

转onnx需要避免dict和str作为模型的输入输出，当dict和str为输入时，会看成常量，若为输出，则会删除str。

原网络：

```
    def forward(self,data,test_mode=True):
    x1, x2, desc1, desc2 = data['x1'][:,:,:2], data['x2'][:,:,:2], data['desc1'],
data['desc2']
    desc1, desc2 = torch.nn.functional.normalize(desc1,dim=-1),
torch.nn.functional.normalize(desc2,dim=-1)
    if test_mode:
        encode_x1,encode_x2=data['x1'],data['x2']
    else:
        encode_x1,encode_x2=data['aug_x1'], data['aug_x2']

    #preparation
    desc_dismat=(2-2*torch.matmul(desc1,desc2.transpose(1,2))).sqrt_()
    values,nn_index=torch.topk(desc_dismat,k=2,largest=False,dim=-1,sorted=True)
    nn_index2=torch.min(desc_dismat,dim=1).indices.squeeze(1)
    inverse_ratio_score,nn_index1=values[:,:,1]/values[:,:,0],nn_index[:,:,0]#get
inverse score
```

解决：将原网络的字典输入进行拆开

```
def forward(self,x1_in, x2_in, desc1_in, desc2_in,test_mode=True):
    x1, x2, desc1, desc2 = x1_in[:,:,:2],x2_in[:,:,:2], desc1_in, desc2_in
    desc1, desc2 = torch.nn.functional.normalize(desc1,dim=-1), 
torch.nn.functional.normalize(desc2,dim=-1)
    encode_x1,encode_x2=x1_in,x2_in
    desc_dismat=(2-2*torch.matmul(desc1,desc2.transpose(1,2))).sqrt_()
    values,nn_index=torch.topk(desc_dismat,k=2,largest=False,dim=-1,sorted=True)
    nn_index2=torch.min(desc_dismat,dim=1).indices.squeeze(1)
    inverse_ratio_score,nn_index1=values[:,:,1]/values[:,:,0],nn_index[:,:,0]#get
inverse score
```

2.1.2 问题2：`RuntimeError: Unsupported: ONNX export of instance_norm for unknown channel size.`

```
Traceback (most recent call last):
File "/home/skywang/work/funsine/SGMNet-main/demo/demo.py", line 73, in <module>
    matcher.export_to_onnx(test_data)
File "/home/skywang/work/funsine/SGMNet-main/components/matchers.py", line 94, in
export_to_onnx
    verbose=True
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/__init__.py", line 365, in export
    export_modules_as_functions,
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/utils.py", line 178, in export
    export_modules_as_functions=export_modules_as_functions,
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/utils.py", line 1084, in _export
    dynamic_axes=dynamic_axes,
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/utils.py", line 739, in _model_to_graph
    module=module,
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/utils.py", line 308, in _optimize_graph
    graph = _C._jit_pass_onnx(graph, operator_export_type)
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/__init__.py", line 416, in _run_symbolic_function
    return utils._run_symbolic_function(*args, **kwargs)
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/utils.py", line 1406, in _run_symbolic_function
    return symbolic_fn(g, *inputs, **attrs)
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/symbolic_helper.py", line 234, in wrapper
    return fn(g, *args, **kwargs)
File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-
packages/torch/onnx/symbolic_opset9.py", line 1965, in instance_norm
    "Unsupported: ONNX export of instance_norm for unknown " "channel size."
RuntimeError: Unsupported: ONNX export of instance_norm for unknown channel size.
```

定位： 分析上面错误，在将模型从 PyTorch 导出为 ONNX 格式时，遇到了不支持导出instance_norm 操作的情况。具体原因是，导出的 ONNX 模型无法处理 instance_norm 操作，因为该操作的通道数未知，无法正确推断和处理。

注意，刚开始将这个错误丢进chatgpt，得到如下解决思路：

```
解决方案：
指定通道数： 确保在使用 instance_norm 操作时，明确指定输入的通道数，以便 ONNX 能正确导出该操作。

如果你使用的是 nn.InstanceNorm2d 或 nn.InstanceNorm1d，请确保给定的输入张量的形状明确包含通道维度。
手动替换 instance_norm： 如果导出 ONNX 时仍然遇到问题，可以尝试用其他规范化操作替换 instance_norm，例如 batch_norm，或者在导出过程中避免使用该层。

升级 PyTorch 和 ONNX 库： 尝试升级 PyTorch 和 ONNX 版本，因为较新的版本可能修复了导出不支持的操作问题。

自定义导出操作： 如果你熟悉 ONNX 的自定义操作，可以尝试通过编写自定义导出逻辑来支持 instance_norm，以确保模型的完整导出。
```

在经过尝试之后并不能解决问题，并不是instance_norm的问题，而是需要继续分析是instance_norm前的什么操作导致张量维度未知。

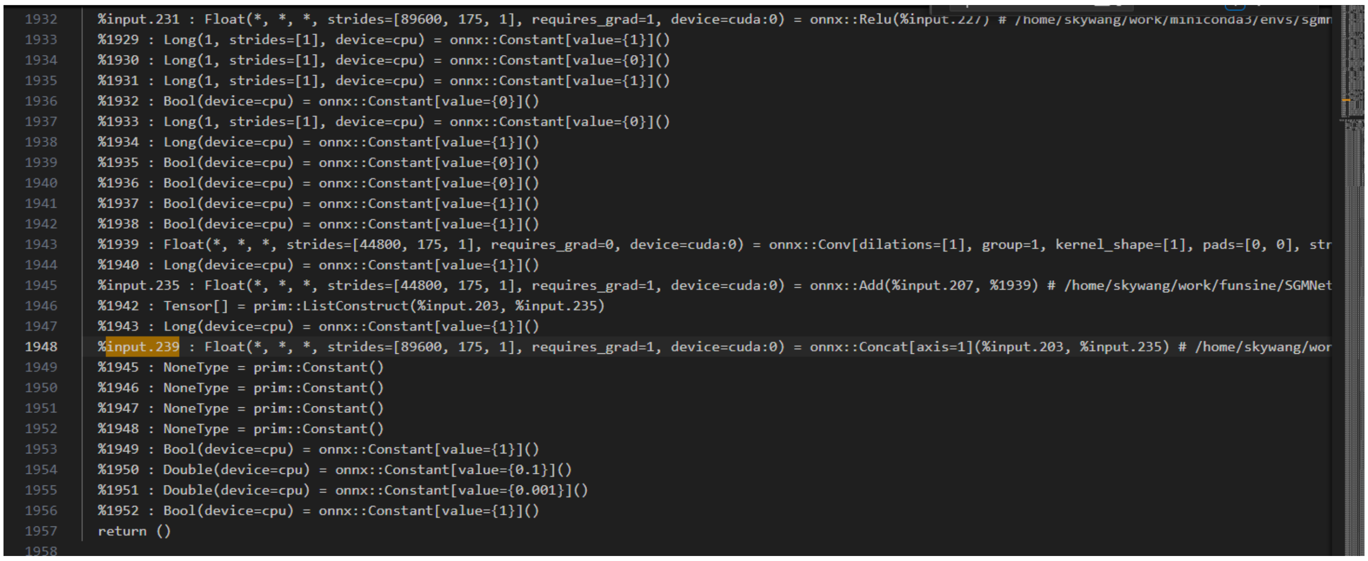
根据`Traceback (most recent call last):`,进入`File "/home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/onnx/symbolic_opset9.py"`,打印出进入instance_norm层的输入：

```
symbolic_helper.check_training_mode(use_input_stats, "instance_norm")
print(input)
channel_size = symbolic_helper._get_tensor_dim_size(input, 1)
if weight is None or symbolic_helper._is_none(weight):
    if channel_size is None:
        raise RuntimeError(
            "Unsupported: ONNX export of instance_norm for unknown " "channel size."
        )
```

得到结果如下：

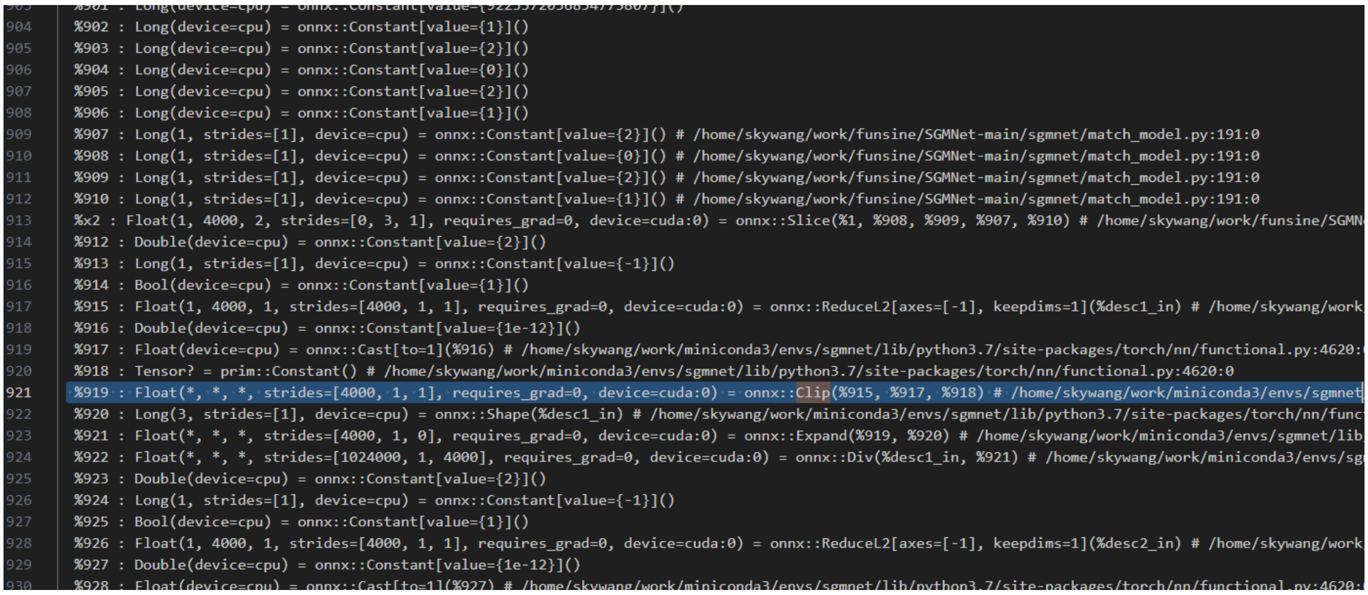
```
input.239 defined in (%input.239 : Float(*, *, *, strides=[89600, 175, 1], requires_grad=1, device=cuda:0) = onnx::Concat[axis=1](%input.203, %input.235) # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:153:0
```

) 确实进入instance_norm前的input的维度未知。开启torch.onnx.erport()的verbose，查看ONNX graph:



```
%input.231 : Float(*, *, *, strides=[89600, 175, 1], requires_grad=1, device=cuda:0) = onnx::Relu(%input.227) # /home/skywang/work/miniconda3/envs/sgmn
%1929 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}]()
%1930 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}]()
%1931 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}]()
%1932 : Bool(device=cpu) = onnx::Constant[value={0}]()
%1933 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}]()
%1934 : Long(device=cpu) = onnx::Constant[value={1}]()
%1935 : Bool(device=cpu) = onnx::Constant[value={0}]()
%1936 : Bool(device=cpu) = onnx::Constant[value={0}]()
%1937 : Bool(device=cpu) = onnx::Constant[value={1}]()
%1938 : Bool(device=cpu) = onnx::Constant[value={1}]()
%1939 : Float(*, *, *, strides=[44800, 175, 1], requires_grad=0, device=cuda:0) = onnx::Conv[dilations=[1], group=1, kernel_shape=[1], pads=[0, 0], str
%1940 : Long(device=cpu) = onnx::Constant[value={1}]()
%input.235 : Float(*, *, *, strides=[44800, 175, 1], requires_grad=1, device=cuda:0) = onnx::Add(%input.207, %1939) # /home/skywang/work/funsine/SGMNet
%1942 : Tensor[] = prim::ListConstruct(%input.203, %input.235)
%1943 : Long(device=cpu) = onnx::Constant[value={1}]()
%input.239 : Float(*, *, *, strides=[89600, 175, 1], requires_grad=1, device=cuda:0) = onnx::Concat[axis=1](%input.203, %input.235) # /home/skywang/wor
%1945 : NoneType = prim::Constant()
%1946 : NoneType = prim::Constant()
%1947 : NoneType = prim::Constant()
%1948 : NoneType = prim::Constant()
%1949 : Bool(device=cpu) = onnx::Constant[value={1}]()
%1950 : Double(device=cpu) = onnx::Constant[value={0.1}]()
%1951 : Double(device=cpu) = onnx::Constant[value={0.001}]()
%1952 : Bool(device=cpu) = onnx::Constant[value={1}]()
return ()
```

从这行开始往上查找，找到是那个算子操作导致的此处维度未知：



```
%901 : Long(device=cpu) = onnx::Constant[value={9223372036854775807}]()
%902 : Long(device=cpu) = onnx::Constant[value={1}]()
%903 : Long(device=cpu) = onnx::Constant[value={2}]()
%904 : Long(device=cpu) = onnx::Constant[value={0}]()
%905 : Long(device=cpu) = onnx::Constant[value={2}]()
%906 : Long(device=cpu) = onnx::Constant[value={1}]()
%907 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%908 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%909 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%910 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%x2 : Float(1, 4000, 2, strides=[0, 3, 1], requires_grad=0, device=cuda:0) = onnx::Slice(%1, %908, %909, %907, %910) # /home/skywang/work/funsine/SGMN
%912 : Double(device=cpu) = onnx::Constant[value={2}]()
%913 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={-1}]()
%914 : Bool(device=cpu) = onnx::Constant[value={1}]()
%915 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1](%desc1_in) # /home/skywang/work
%916 : Double(device=cpu) = onnx::Constant[value={1e-12}]()
%917 : Float(device=cpu) = onnx::Cast[to=1](%916) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:
%918 : Tensor? = prim::Constant() # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:0
%919 : Float(*, *, *, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::Clip(%915, %917, %918) # /home/skywang/work/miniconda3/envs/sgmnet
%920 : Long(3, strides=[1], device=cpu) = onnx::Shape(%desc1_in) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/func
%921 : Float(*, *, *, strides=[4000, 1, 0], requires_grad=0, device=cuda:0) = onnx::Expand(%919, %920) # /home/skywang/work/miniconda3/envs/sgmnet/lib
%922 : Float(*, *, *, strides=[1024000, 1, 4000], requires_grad=0, device=cuda:0) = onnx::Div(%desc1_in, %921) # /home/skywang/work/miniconda3/envs/sg
%923 : Double(device=cpu) = onnx::Constant[value={2}]()
%924 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={-1}]()
%925 : Bool(device=cpu) = onnx::Constant[value={1}]()
%926 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1](%desc2_in) # /home/skywang/work
%927 : Double(device=cpu) = onnx::Constant[value={1e-12}]()
%928 : Float(device=cpu) = onnx::Cast[to=1](%927) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:
```

一直追踪到这个位置，从上图看出，从%919开始张量的形状未知，这行上下进行分析，可以推断出是网络结构源码的这一行：

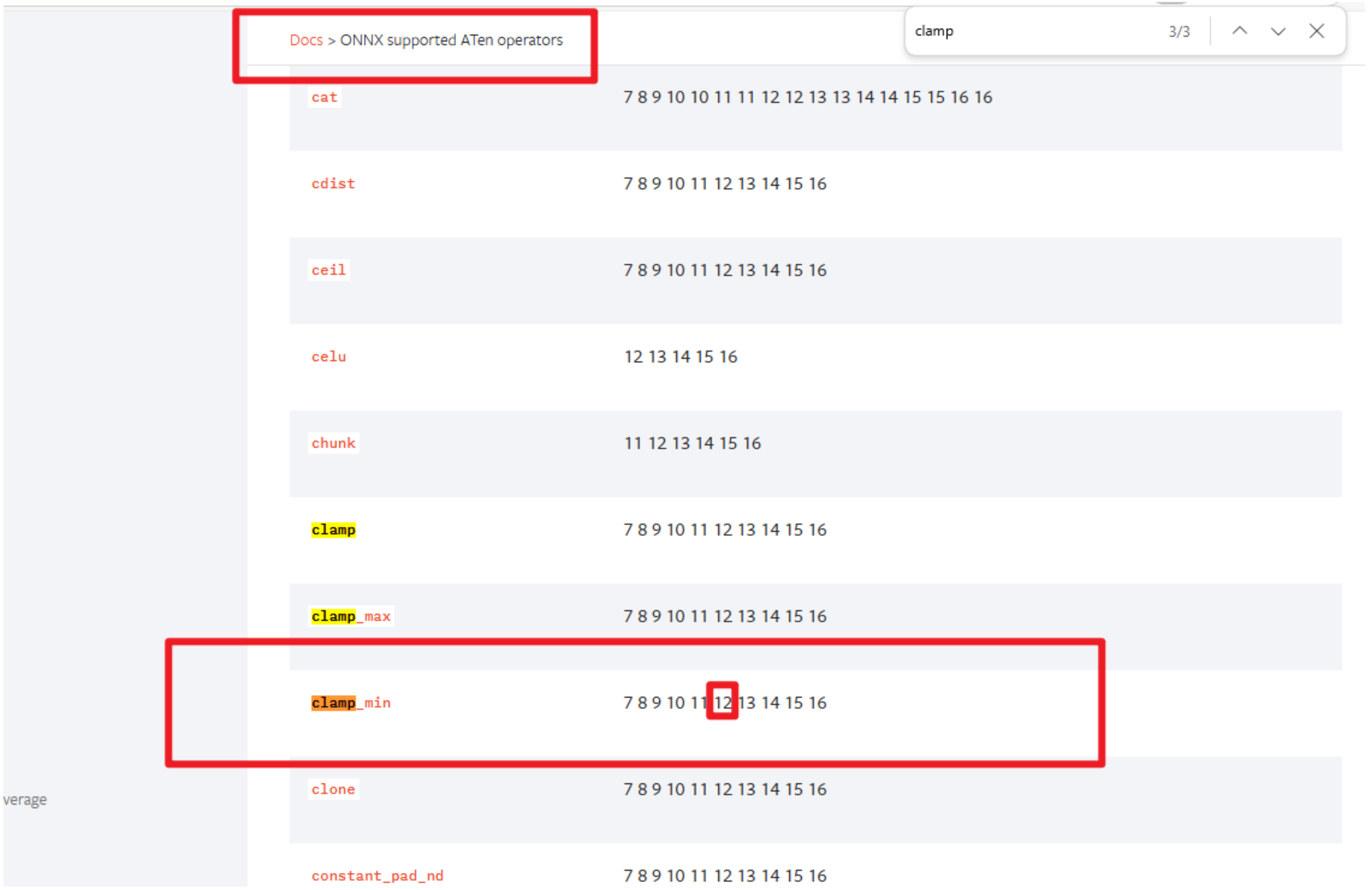
```
desc1, desc2 = torch.nn.functional.normalize(desc1,dim=-1),
torch.nn.functional.normalize(desc2,dim=-1)
```

这是torch官方的代码，进入查看如下：

```
if out is None:
    denom = input.norm(p, dim, keepdim=True).clamp_min(eps).expand_as(input)
    return input / denom
```

会执行上面的代码，结合onnx graph，pytorch的clamp_min(eps)对应到onnx中正是onnx：Clip（https://onnx.ai/onnx/operators/onnx__Clip.html ），故是clamp_min(eps)算子出现问题，从这个算子操作之后，张量的维度变得未知，导致后面instance_norm层不能正确获取到channel size。

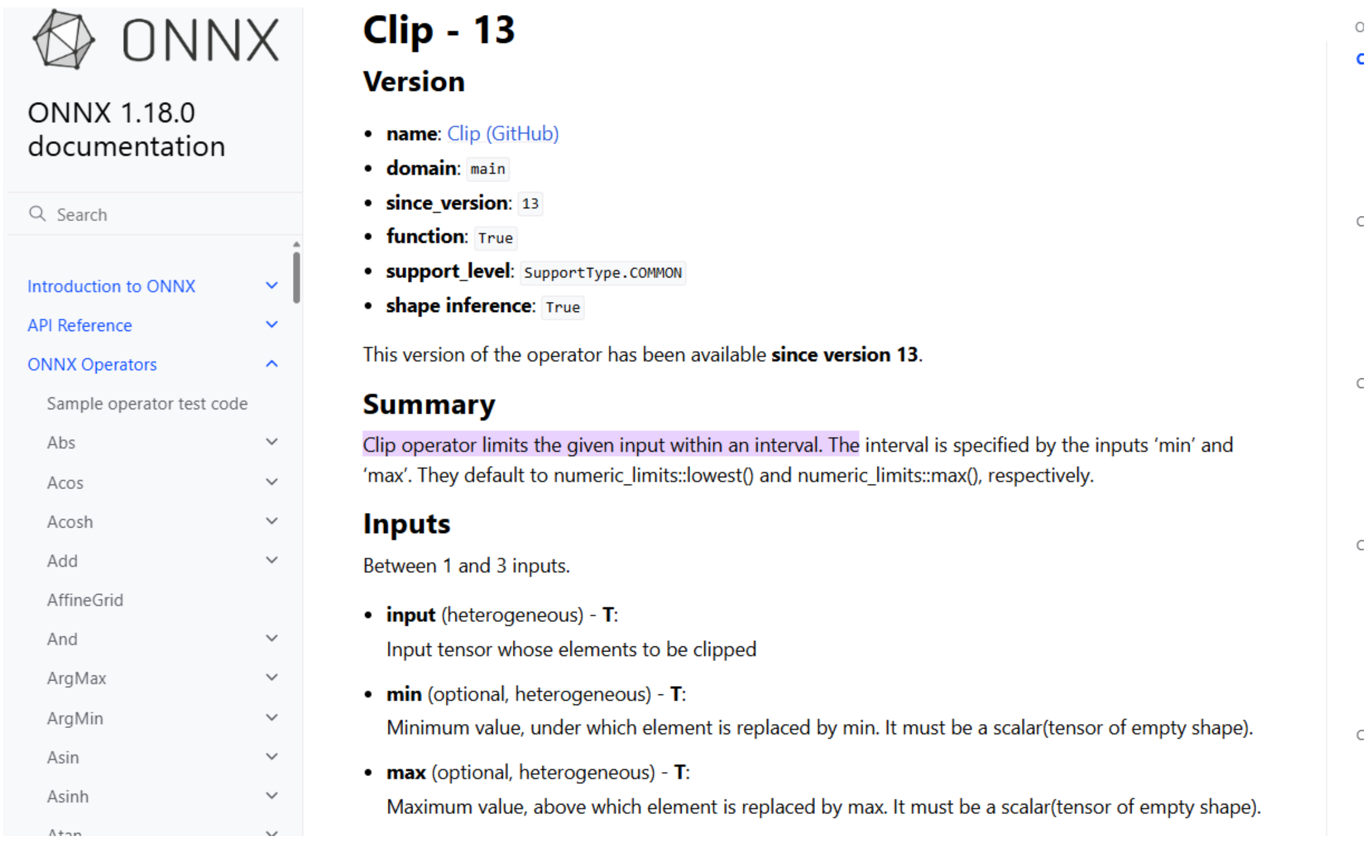
解决：查看pytorch官方文档（https://pytorch.org/docs/1.12/onnx_supported_aten_ops.html?highlight=clamp_min），clamp_min是支持导出onnx的，但是报错


Docs > ONNX supported ATen operators

clamp 3/3

cat	7 8 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16
cdist	7 8 9 10 11 12 13 14 15 16
ceil	7 8 9 10 11 12 13 14 15 16
celu	12 13 14 15 16
chunk	11 12 13 14 15 16
clamp	7 8 9 10 11 12 13 14 15 16
clamp_max	7 8 9 10 11 12 13 14 15 16
clamp_min	7 8 9 10 11 12 13 14 15 16
clone	7 8 9 10 11 12 13 14 15 16
constant_pad_nd	7 8 9 10 11 12 13 14 15 16

查看onnx关于clip算子的文档，



Clip - 13

Version

- **name**: Clip (GitHub)
- **domain**: `main`
- **since_version**: `13`
- **function**: `True`
- **support_level**: `SupportType.COMMON`
- **shape inference**: `True`

This version of the operator has been available **since version 13**.

Summary

Clip operator limits the given input within an interval. The interval is specified by the inputs 'min' and 'max'. They default to numeric_limits::lowest() and numeric_limits::max(), respectively.

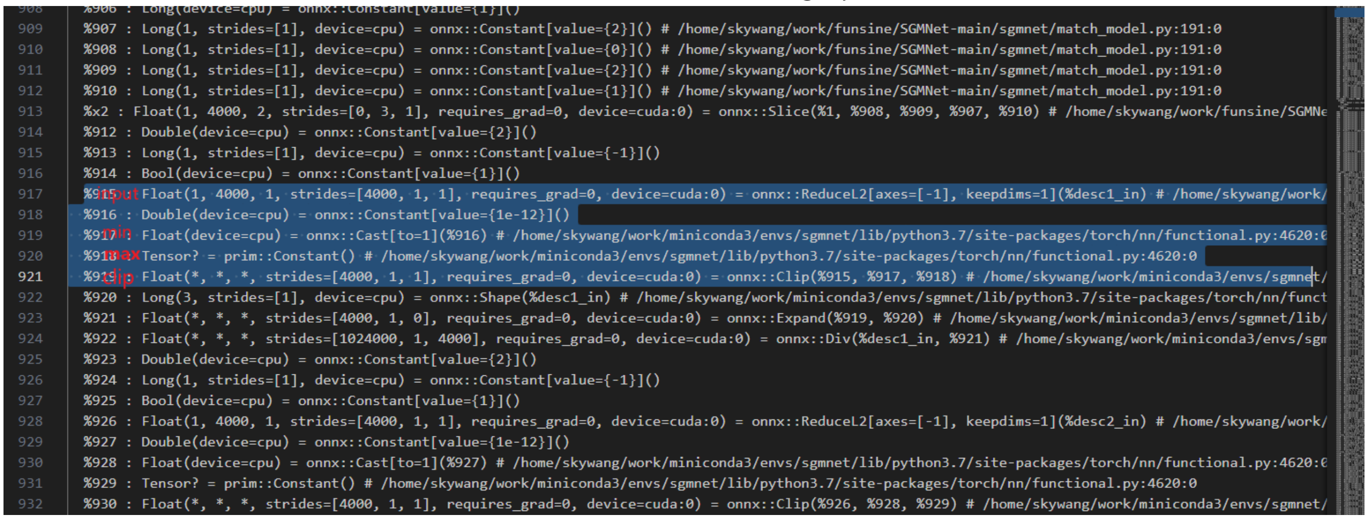
Inputs

Between 1 and 3 inputs.

- **input** (heterogeneous) - **T**:
  Input tensor whose elements to be clipped
- **min** (optional, heterogeneous) - **T**:
  Minimum value, under which element is replaced by min. It must be a scalar(tensor of empty shape).
- **max** (optional, heterogeneous) - **T**:
  Maximum value, above which element is replaced by max. It must be a scalar(tensor of empty shape).

,

显示有三个输入，但是后两个输入是可选的，再结合之前的onnx graph,



```
%906 : Long(device=cpu) = onnx::Constant[value={1}]()
%907 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%908 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%909 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%910 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}]() # /home/skywang/work/funsine/SGMNet-main/sgmnet/match_model.py:191:0
%x2 : Float(1, 4000, 2, strides=[0, 3, 1], requires_grad=0, device=cuda:0) = onnx::Slice(%1, %908, %909, %907, %910) # /home/skywang/work/funsine/SGMNe
%912 : Double(device=cpu) = onnx::Constant[value={2}]()
%913 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={-1}]()
%914 : Bool(device=cpu) = onnx::Constant[value={1}]()
%915 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1](%desc1_in) # /home/skywang/work/
%916 : Double(device=cpu) = onnx::Constant[value={1e-12}]()
%917 : Float(device=cpu) = onnx::Cast[to=1](%916) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:0
%918 : Tensor? = prim::Constant() # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:0
%919 : Float(*, *, *, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::Clip(%915, %917, %918) # /home/skywang/work/miniconda3/envs/sgmnet/
%920 : Long(3, strides=[1], device=cpu) = onnx::Shape(%desc1_in) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/funct
%921 : Float(*, *, *, strides=[4000, 1, 0], requires_grad=0, device=cuda:0) = onnx::Expand(%919, %920) # /home/skywang/work/miniconda3/envs/sgmnet/lib/
%922 : Float(*, *, *, strides=[1024000, 1, 4000], requires_grad=0, device=cuda:0) = onnx::Div(%desc1_in, %921) # /home/skywang/work/miniconda3/envs/sgm
%923 : Double(device=cpu) = onnx::Constant[value={2}]()
%924 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={-1}]()
%925 : Bool(device=cpu) = onnx::Constant[value={1}]()
%926 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1](%desc2_in) # /home/skywang/work/
%927 : Double(device=cpu) = onnx::Constant[value={1e-12}]()
%928 : Float(device=cpu) = onnx::Cast[to=1](%927) # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:0
%929 : Tensor? = prim::Constant() # /home/skywang/work/miniconda3/envs/sgmnet/lib/python3.7/site-packages/torch/nn/functional.py:4620:0
%930 : Float(*, *, *, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::Clip(%926, %928, %929) # /home/skywang/work/miniconda3/envs/sgmnet/
```

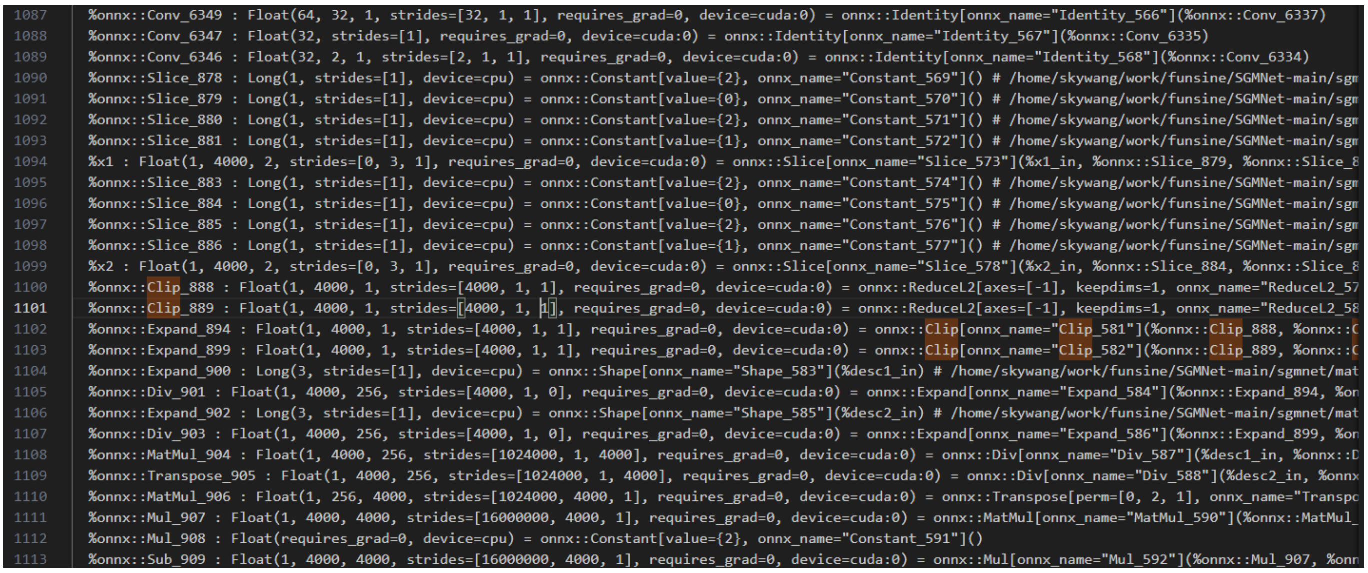
, 在max这里显示了`Tensor?`，推测是否onnx并不支持clamp_min，只支持完整的clamp操作。

尝试使用完整的clamp替代clamp_min，如下：

```
#TODO:normalize,修改了normalize内的代码
# desc1, desc2 = torch.nn.functional.normalize(desc1,dim=-1),
torch.nn.functional.normalize(desc2,dim=-1)

# 计算 L2 范数
l2_norm1,l2_norm2 = torch.norm(desc1, p=2, dim=-1, keepdim=True),torch.norm(desc1,
p=2, dim=-1, keepdim=True)
l2_norm1,l2_norm2 = torch.clamp(l2_norm1, eps,100),torch.clamp(l2_norm2, eps,100)
l2_norm1,l2_norm2 = l2_norm1.expand_as(desc1),l2_norm2.expand_as(desc2)
# 归一化
desc1,desc2 = desc1 / l2_norm1, desc2 / l2_norm2
```

再尝试导出，成功！！！，查看onnx graph：



```
%onnx::Conv_6349 : Float(64, 32, 1, strides=[32, 1, 1], requires_grad=0, device=cuda:0) = onnx::Identity[onnx_name="Identity_566"](%onnx::Conv_6337)
%onnx::Conv_6347 : Float(32, strides=[1], requires_grad=0, device=cuda:0) = onnx::Identity[onnx_name="Identity_567"](%onnx::Conv_6335)
%onnx::Conv_6346 : Float(32, 2, 1, strides=[2, 1, 1], requires_grad=0, device=cuda:0) = onnx::Identity[onnx_name="Identity_568"](%onnx::Conv_6334)
%onnx::Slice_878 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}, onnx_name="Constant_569"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_879 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}, onnx_name="Constant_570"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_880 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}, onnx_name="Constant_571"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_881 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}, onnx_name="Constant_572"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%x1 : Float(1, 4000, 2, strides=[0, 3, 1], requires_grad=0, device=cuda:0) = onnx::Slice[onnx_name="Slice_573"](%x1_in, %onnx::Slice_879, %onnx::Slice_8
%onnx::Slice_883 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}, onnx_name="Constant_574"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_884 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={0}, onnx_name="Constant_575"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_885 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={2}, onnx_name="Constant_576"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%onnx::Slice_886 : Long(1, strides=[1], device=cpu) = onnx::Constant[value={1}, onnx_name="Constant_577"]() # /home/skywang/work/funsine/SGMNet-main/sgm
%x2 : Float(1, 4000, 2, strides=[0, 3, 1], requires_grad=0, device=cuda:0) = onnx::Slice[onnx_name="Slice_578"](%x2_in, %onnx::Slice_884, %onnx::Slice_8
%onnx::Clip_888 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1, onnx_name="ReduceL2_57
%onnx::Clip_889 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::ReduceL2[axes=[-1], keepdims=1, onnx_name="ReduceL2_58
%onnx::Expand_894 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::Clip[onnx_name="Clip_581"](%onnx::Clip_888, %onnx::C
%onnx::Expand_899 : Float(1, 4000, 1, strides=[4000, 1, 1], requires_grad=0, device=cuda:0) = onnx::Clip[onnx_name="Clip_582"](%onnx::Clip_889, %onnx::C
%onnx::Expand_900 : Long(3, strides=[1], device=cpu) = onnx::Shape[onnx_name="Shape_583"](%desc1_in) # /home/skywang/work/funsine/SGMNet-main/sgmnet/mat
%onnx::Div_901 : Float(1, 4000, 256, strides=[4000, 1, 0], requires_grad=0, device=cuda:0) = onnx::Expand[onnx_name="Expand_584"](%onnx::Expand_894, %on
%onnx::Expand_902 : Long(3, strides=[1], device=cpu) = onnx::Shape[onnx_name="Shape_585"](%desc2_in) # /home/skywang/work/funsine/SGMNet-main/sgmnet/mat
%onnx::Div_903 : Float(1, 4000, 256, strides=[4000, 1, 0], requires_grad=0, device=cuda:0) = onnx::Expand[onnx_name="Expand_586"](%onnx::Expand_899, %on
%onnx::MatMul_904 : Float(1, 4000, 256, strides=[1024000, 1, 4000], requires_grad=0, device=cuda:0) = onnx::Div[onnx_name="Div_587"](%desc1_in, %onnx::D
%onnx::Transpose_905 : Float(1, 4000, 256, strides=[1024000, 1, 4000], requires_grad=0, device=cuda:0) = onnx::Div[onnx_name="Div_588"](%desc2_in, %onnx
%onnx::MatMul_906 : Float(1, 256, 4000, strides=[1024000, 4000, 1], requires_grad=0, device=cuda:0) = onnx::Transpose[perm=[0, 2, 1], onnx_name="Transpo
%onnx::Mul_907 : Float(1, 4000, 4000, strides=[16000000, 4000, 1], requires_grad=0, device=cuda:0) = onnx::MatMul[onnx_name="MatMul_590"](%onnx::MatMul_
%onnx::Mul_908 : Float(requires_grad=0, device=cpu) = onnx::Constant[value={2}, onnx_name="Constant_591"]()
%onnx::Sub_909 : Float(1, 4000, 4000, strides=[16000000, 4000, 1], requires_grad=0, device=cuda:0) = onnx::Mul[onnx_name="Mul_592"](%onnx::Mul_907, %onn
```

成功解决，维度可以正常获取。

2.2 onnx转om

对于开源框架的网络模型（如Caffe、TensorFlow等），不能直接在昇腾AI处理器上运行推理，需要先使用ATC（Ascend Tensor Compiler）工具将开源框架的网络模型转换为适配昇腾AI处理器的离线模型（*.om文件）

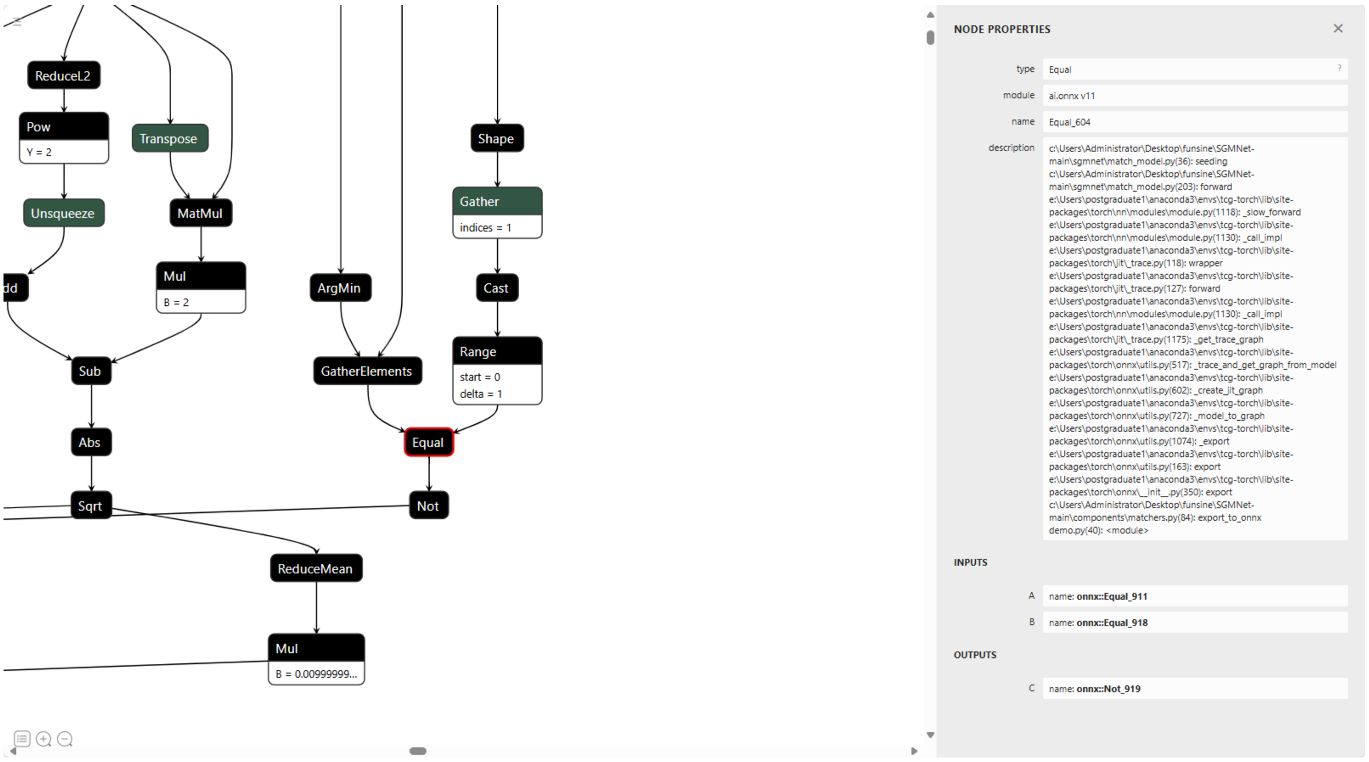
```
atc --model=/home/tcg/VisualHabitFusion/model.onnx --framework=5 --
output=/home/tcg/VisualHabitFusion/matcher --soc_version=Ascend310B4 --
input_shape="x1_in:1,4000,3;x2_in:1,4000,3;desc1_in:1,4000,256;desc2_in:1,4000,256
"
```

问题1：`The Equal_604 op dtype is not same, type1:DT_INT32, type2:DT_INT64`

```
ATC run failed, Please check the detail log, Try 'atc --help' for more information
E10042: GenerateOfflineModel execute failed.
    TraceBack (most recent call last):
    op[Equal_604], The Equal_604 op dtype is not same, type1:DT_INT32,
type2:DT_INT64[FUNC:CheckTwoInputDtypeSame][FILE:util.cc][LINE:116]
    Verifying Equal_604 failed.[FUNC:InferShapeAndType][FILE:infershape_pass.cc]
[LINE:137]
    Call InferShapeAndType for node:Equal_604(Equal) failed[FUNC:Infer]
[FILE:infershape_pass.cc][LINE:119]
    process pass InferShapePass on node:Equal_604 failed,
ret:4294967295[FUNC:RunPassesOnNode][FILE:base_pass.cc][LINE:571]
    build graph failed, graph id:0, ret:1343242270[FUNC:BuildModelWithGraphId]
[FILE:ge_generator.cc][LINE:1615]
    GenerateOfflineModel execute failed.
```

问题定位：

1.在netron中将onnx模型的结构打印出来，寻找问题节点的位置



NODE PROPERTIES

type	Equal
module	ai.onnx v11
name	Equal_604

description
```
c:\Users\Administrator\Desktop\funsine\SGMNet-
main\sgmnet\match_model.py(36): seeding
c:\Users\Administrator\Desktop\funsine\SGMNet-
main\sgmnet\match_model.py(203): forward
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\nn\modules\module.py(1118): _slow_forward
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\nn\modules\module.py(1130): _call_impl
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\jit\_trace.py(118): wrapper
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\jit\_trace.py(127): forward
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\nn\modules\module.py(1130): _call_impl
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\jit\_trace.py(1175): _get_trace_graph
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\utils.py(517): _trace_and_get_graph_from_model
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\utils.py(602): _create_jit_graph
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\utils.py(727): _model_to_graph
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\utils.py(1074): _export
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\utils.py(163): export
e:\Users\postgraduate1\anaconda3\envs\tcg-torch\lib\site-
packages\torch\onnx\__init__.py(350): export
c:\Users\Administrator\Desktop\funsine\SGMNet-
main\components\matchers.py(84): export_to_onnx
demo.py(40): <module>
```

INPUTS

A	name: **onnx::Equal_911**
B	name: **onnx::Equal_918**

OUTPUTS

C	name: **onnx::Not_919**

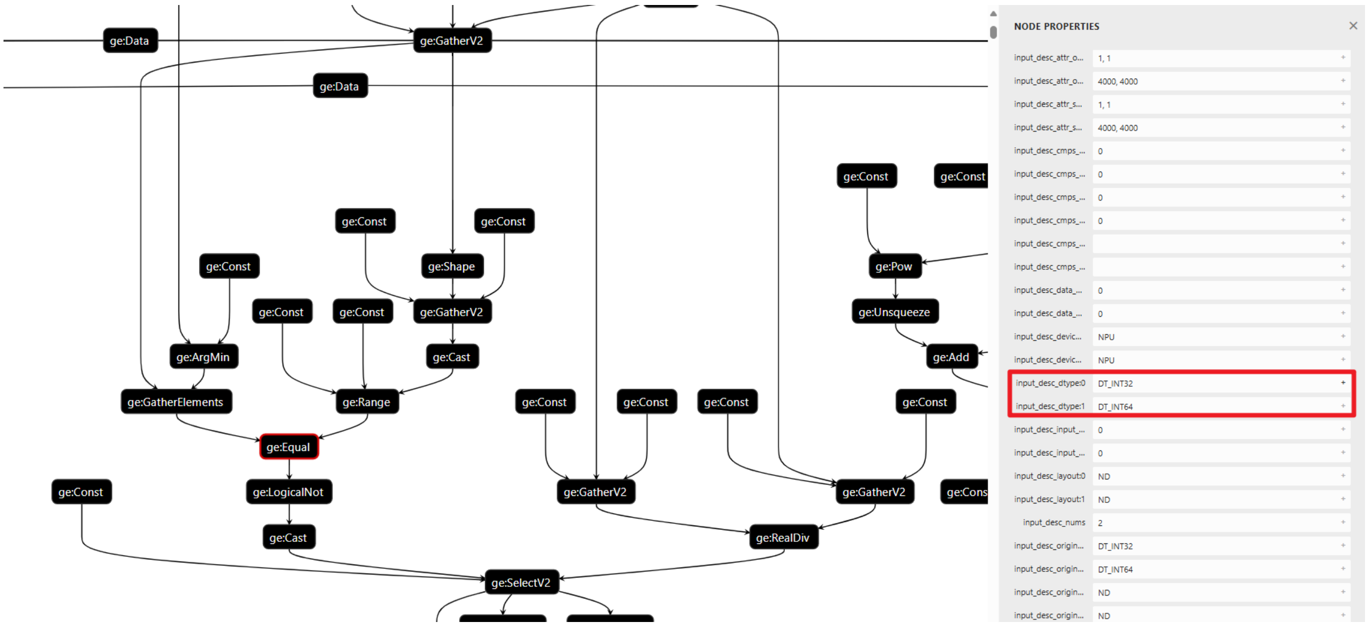
可以定位到源码为：

```
if use_mc:

mask_not_mutual=nn_index2.gather(dim=-1,index=nn_index1)!=torch.arange(nn_index1.s
hape[1],device='cuda')
match_score[mask_not_mutual]=-1
```

打印`!=`两侧的数据类型为`int64,int64`,在pytorch中类型是正确的，推测是atc工具的问题。

2.追寻atc问题： 使用 `export DUMP_GE_GRAPH=2`生成Dump图，在Dump图目录下找到`ge_onnx_***_graph_0_after_infershape.pbtxt`，可在GE图中确定Equal算子确实存在两个input，算子的dtype类型分别为int32和int64。



NODE PROPERTIES

input_desc_attr_o...	1, 1
input_desc_attr_o...	4000, 4000
input_desc_attr_s...	1, 1
input_desc_attr_s...	4000, 4000
input_desc_cmps_...	0
input_desc_cmps_...	0
input_desc_cmps_...	0
input_desc_cmps_...	0
input_desc_cmps_...	
input_desc_cmps_...	
input_desc_data_...	0
input_desc_data_...	0
input_desc_devic...	NPU
input_desc_devic...	NPU
input_desc_dtype:0	DT_INT32
input_desc_dtype:1	DT_INT64
input_desc_input_...	0
input_desc_input_...	0
input_desc_layout:0	ND
input_desc_layout:1	ND
input_desc_nums	2
input_desc_origin...	DT_INT32
input_desc_origin...	DT_INT64
input_desc_origin...	ND
input_desc_origin...	ND

解决：在equal前添加cast算子，将int32转为int64

```
import onnx
from onnx import helper, TensorProto
# 加载现有的 ONNX 模型
onnx_model = onnx.load('C:\\Users\\Administrator\\Desktop\\funsine\\SGMNet-
main\\demo\\model.onnx')
# 获取模型中的计算图
graph = onnx_model.graph
# 查找目标节点 GatherElements_596 和 Equal_604
gather_node_name = "GatherElements_596"
equal_node_name = "Equal_604"
gather_output = None
# 先找到 GatherElements_596 节点的输出
for node in graph.node:
    if node.name == gather_node_name:
        gather_output = node.output[0]
        break
# 确保找到了 GatherElements_596 的输出
```

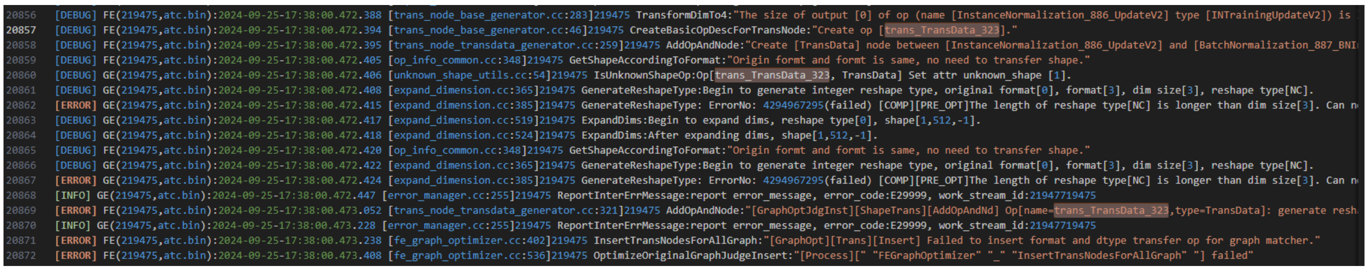
```
if gather_output is None:
    raise ValueError(f"Node {gather_node_name} not found in the graph.")
# 创建一个 Cast 节点，将 GatherElements_596 的输出转换为 int64
cast_output = gather_output + "_casted"
cast_node = helper.make_node(
    'Cast',  # 算子类型
    inputs=[gather_output],  # Cast的输入是GatherElements的输出
    outputs=[cast_output],  # Cast的输出
    to=TensorProto.INT64  # 将输出转换为 int64
)
# 将 Cast 节点添加到计算图中
graph.node.append(cast_node)
# 更新 Equal_604 节点，将它的输入改为 Cast 节点的输出
for node in graph.node:
    if node.name == equal_node_name:
        for i, input_name in enumerate(node.input):
            if input_name == gather_output:
                node.input[i] = cast_output  # 修改 Equal_604 的输入
                break
# 保存修改后的模型
onnx.save(onnx_model, 'model_with_cast.onnx')
print("Cast node successfully inserted between GatherElements_596 and Equal_604.")
```

问题2：Op[name=trans_TransData_323,type=TransData]: generate reshape type mask of input failed

```
ATC run failed, Please check the detail log, Try 'atc --help' for more information
E10042: GenerateOfflineModel execute failed.
    TraceBack (most recent call last):
    [GraphOptJdgInst][ShapeTrans][AddOpAndNd]
Op[name=trans_TransData_323,type=TransData]: generate reshape type mask of input
failed.[FUNC:AddOpAndNode][FILE:trans_node_transdata_generator.cc][LINE:321]
    [GraphOpt][Trans][Insert] Failed to insert format and dtype transfer op for
graph matcher.[FUNC:InsertTransNodesForAllGraph][FILE:fe_graph_optimizer.cc]
[LINE:402]
    Call OptimizeOriginalGraphJudgeInsert failed, ret:-1,
engine_name:AIcoreEngine,
graph_name:matcher[FUNC:OptimizeOriginalGraphJudgeInsert][FILE:graph_optimize.cc]
[LINE:251]
    build graph failed, graph id:0, ret:-1[FUNC:BuildModelWithGraphId]
[FILE:ge_generator.cc][LINE:1615]
    GenerateOfflineModel execute failed.
```

问题定位：

在atc命令后添加`--log=debug`打印日志，进入日志文件查找`trans_TransData_323`如下：



```
[DEBUG] FE(219475,atc.bin):2024-09-25-17:38:00.472.388 [trans_node_base_generator.cc:283]219475 TransformDimTo4:"The size of output [0] of op (name [InstanceNormalization_886_UpdateV2] type [INTrainingUpdateV2]) is
[DEBUG] FE(219475,atc.bin):2024-09-25-17:38:00.472.394 [trans_node_base_generator.cc:46]219475 CreateBasicOpDescForTransNode:"Create op [trans_TransData_323]."
[DEBUG] FE(219475,atc.bin):2024-09-25-17:38:00.472.395 [trans_node_transdata_generator.cc:259]219475 AddOpAndNode:"Create [TransData] node between [InstanceNormalization_886_UpdateV2] and [BatchNormalization_887_BNI
[DEBUG] FE(219475,atc.bin):2024-09-25-17:38:00.472.405 [op_info_common.cc:348]219475 GetShapeAccordingToFormat:"Origin formt and formt is same, no need to transfer shape."
[DEBUG] GE(219475,atc.bin):2024-09-25-17:38:00.472.406 [unknown_shape_utils.cc:54]219475 IsUnknownShapeOp:Op[trans_TransData_323, TransData] Set attr unknown_shape [1].
[DEBUG] GE(219475,atc.bin):2024-09-25-17:38:00.472.408 [expand_dimension.cc:365]219475 GenerateReshapeType:Begin to generate integer reshape type, original format[0], format[3], dim size[3], reshape type[NC].
[ERROR] GE(219475,atc.bin):2024-09-25-17:38:00.472.415 [expand_dimension.cc:385]219475 GenerateReshapeType: ErrorNo: 4294967295(failed) [COMP][PRE_OPT]The length of reshape type[NC] is longer than dim size[3]. Can n
[DEBUG] GE(219475,atc.bin):2024-09-25-17:38:00.472.417 [expand_dimension.cc:519]219475 ExpandDims:Begin to expand dims, reshape type[0], shape[1,512,-1].
[DEBUG] GE(219475,atc.bin):2024-09-25-17:38:00.472.418 [expand_dimension.cc:524]219475 ExpandDims:After expanding dims, shape[1,512,-1].
[DEBUG] FE(219475,atc.bin):2024-09-25-17:38:00.472.420 [op_info_common.cc:348]219475 GetShapeAccordingToFormat:"Origin formt and formt is same, no need to transfer shape."
[DEBUG] GE(219475,atc.bin):2024-09-25-17:38:00.472.422 [expand_dimension.cc:365]219475 GenerateReshapeType:Begin to generate integer reshape type, original format[0], format[3], dim size[3], reshape type[NC].
[ERROR] GE(219475,atc.bin):2024-09-25-17:38:00.472.424 [expand_dimension.cc:385]219475 GenerateReshapeType: ErrorNo: 4294967295(failed) [COMP][PRE_OPT]The length of reshape type[NC] is longer than dim size[3]. Can n
[INFO] GE(219475,atc.bin):2024-09-25-17:38:00.472.447 [error_manager.cc:255]219475 ReportInterErrMessage:report error_message, error_code:E29999, work_stream_id:21947719475
[ERROR] FE(219475,atc.bin):2024-09-25-17:38:00.473.052 [trans_node_transdata_generator.cc:321]219475 AddOpAndNode:"[GraphOptJdgInst][ShapeTrans][AddOpAndNd] Op[name=trans_TransData_323,type=TransData]: generate resh
[INFO] GE(219475,atc.bin):2024-09-25-17:38:00.473.228 [error_manager.cc:255]219475 ReportInterErrMessage:report error_message, error_code:E29999, work_stream_id:21947719475
[ERROR] FE(219475,atc.bin):2024-09-25-17:38:00.473.238 [fe_graph_optimizer.cc:402]219475 InsertTransNodesForAllGraph:"[GraphOpt][Trans][Insert] Failed to insert format and dtype transfer op for graph matcher."
[ERROR] FE(219475,atc.bin):2024-09-25-17:38:00.473.408 [fe_graph_optimizer.cc:536]219475 OptimizeOriginalGraphJudgeInsert:"[Process][" "FEGraphOptimizer" "_" "InsertTransNodesForAllGraph" "] failed"
```

分析上面的日志流程如下：

```
1.节点创建:
[DEBUG] Create op [trans_TransData_323]: 成功创建了一个新的操作节点 trans_TransData_323。
[DEBUG] Create [TransData] node between [InstanceNormalization_886_UpdateV2] and [BatchNormalization_887_BNInferenceD] success!: 该节点在 InstanceNormalization_886_UpdateV2 和 BatchNormalization_887_BNInferenceD 之间创建成功。

2.形状处理:
[DEBUG] GetShapeAccordingToFormat:"Origin formt and formt is same, no need to transfer shape.": 输入的形状格式与原始格式相同，因此无需进行形状转换。
[DEBUG] IsUnknownShapeOp:Op[trans_TransData_323, TransData] Set attr unknown_shape [1]: 标记此操作的形状为未知。

3.生成重塑类型:
[DEBUG] GenerateReshapeType:Begin to generate integer reshape type...: 开始生成整数重塑类型，原始格式和目标格式信息被记录下来。
[ERROR] GenerateReshapeType: ErrorNo: 4294967295(failed)... The length of reshape type[NC] is longer than dim size[3].: 生成重塑类型时出错，错误信息表明重塑类型 NC 的长度大于维度大小 3，无法生成整数重塑类型。

4. 扩展维度:
[DEBUG] ExpandDims:Begin to expand dims...: 开始扩展维度操作。
[DEBUG] ExpandDims:After expanding dims, shape[1,512,-1].: 扩展维度后的形状信息。

5.再度生成重塑类型:
由于扩展操作后再次尝试生成重塑类型，但再次出现相同的错误信息，表明在此过程中仍然无法满足重塑要求。

错误报告:
[ERROR] AddOpAndNode:"... generate reshape type mask of input failed.": 在尝试添加操作和节点时，生成输入的重塑类型掩码失败，导致整个操作无法完成。
```

总结：atc尝试在`InstanceNormalization_886_UpdateV2`和`BatchNormalization_887_BNInferenceD`插入一个数据转换算子 `trans_TransData_323`,但是在图计算中执行节点操作时遇到的形状问题，主要是由于重塑类型与实际维度不匹配，导致无法完成操作。核心报错原因是上图 `[error]`这一行:

```
[ERROR] GE(219475,atc.bin):2024-09-25-17:38:00.472.415
[expand_dimension.cc:385]219475 GenerateReshapeType: ErrorNo: 4294967295(failed)
[COMP][PRE_OPT]The length of reshape type[NC] is longer than dim size[3]. Can not
generate integer reshape type
```

问题解决：