

这里首先利用下面代码把给定的 key_model 中的 graph 得到

```
for i in range(4):
    block = getattr(model_key.layers, str(i))
    print(f"Block {i} graph:")
    print(block.forward.graph)
print(model_key.graph)
```

graph 如下:

```
(base) root@gz-ins-667034947493893:~# /usr/local/anaconda3/bin/python /root/data/NS-2025-06-data/06.py
```

Block 0 graph:

```
graph(%self.1 : torch.Block_0,
%x.1 : Tensor,
%start_pos.1 : int,
%freqs_cis.1 : Tensor,
%mask.1 : Tensor?,
%tokens.1 : Tensor):
  %20 : int = prim::Constant[value=1]
  %36 : int = prim::Constant[value=0] # C:\Users\Terox\Desktop\temp\gen.py:735:26
  %42 : int = prim::Constant[value=34] # C:\Users\Terox\Desktop\temp\gen.py:735:42
  %attn.1 : torch.MLA = prim::GetAttr[name="attn"]
  %attn_norm.1 : torch.RMSNorm = prim::GetAttr[name="attn_norm"]
  %13 : Tensor = prim::CallMethod[name="forward"](%attn_norm.1, %x.1) #
C:\Users\Terox\Desktop\temp\gen.py:733:26
  %17 : Tensor =
prim::CallMethod[name="forward"](%attn.1, %13, %start_pos.1, %freqs_cis.1, %mask.1) #
C:\Users\Terox\Desktop\temp\gen.py:733:16
  %x0.1 : Tensor = aten::add(%x.1, %17, %20) # C:\Users\Terox\Desktop\temp\gen.py:733:12
  %ffn.1 : torch.MLP = prim::GetAttr[name="ffn"]
  %ffn_norm.1 : torch.RMSNorm = prim::GetAttr[name="ffn_norm"]
  %29 : Tensor = prim::CallMethod[name="forward"](%ffn_norm.1, %x0.1) #
C:\Users\Terox\Desktop\temp\gen.py:734:25
  %30 : Tensor = prim::CallMethod[name="forward"](%ffn.1, %29) #
C:\Users\Terox\Desktop\temp\gen.py:734:16
  %x1.1 : Tensor = aten::add(%x0.1, %30, %20) # C:\Users\Terox\Desktop\temp\gen.py:734:12
  %37 : Tensor = aten::select(%tokens.1, %36, %36) #
C:\Users\Terox\Desktop\temp\gen.py:735:19
  %38 : Tensor = aten::select(%37, %36, %36) # C:\Users\Terox\Desktop\temp\gen.py:735:19
  %41 : Scalar = aten::item(%38) # C:\Users\Terox\Desktop\temp\gen.py:735:19
  %43 : bool = aten::eq(%41, %42) # C:\Users\Terox\Desktop\temp\gen.py:735:19
  %44 : (Tensor, bool) = prim::TupleConstruct(%x1.1, %43)
  return (%44)
```

这里对于 block0 的结果进行分析:

看 bool 部分, 发现是判断第一个输入是否是和%42 相等, 看上面发现%42 定义的是常数 34
故第一个数字是 34 的时候输出 true

Block 1 graph:

```
graph(%self.1 : torch.Block_1,
%x.1 : Tensor,
%start_pos.1 : int,
%freqs_cis.1 : Tensor,
%mask.1 : Tensor?,
%tokens.1 : Tensor):
%36 : int = prim::Constant[value=0] # C:\Users\Terox\Desktop\temp\gen.py:748:26
%38 : int = prim::Constant[value=1] # C:\Users\Terox\Desktop\temp\gen.py:748:29
%42 : int = prim::Constant[value=134] # C:\Users\Terox\Desktop\temp\gen.py:748:42
%attn.1 : torch.MLA = prim::GetAttr[name="attn"]
%attn_norm.1 : torch.RMSNorm = prim::GetAttr[name="attn_norm"]
%13 : Tensor = prim::CallMethod[name="forward"](%attn_norm.1, %x.1) #
C:\Users\Terox\Desktop\temp\gen.py:746:26
%17 : Tensor =
prim::CallMethod[name="forward"](%attn.1, %13, %start_pos.1, %freqs_cis.1, %mask.1) #
C:\Users\Terox\Desktop\temp\gen.py:746:16
%x4.1 : Tensor = aten::add(%x.1, %17, %38) # C:\Users\Terox\Desktop\temp\gen.py:746:12
%ffn.1 : torch.MLP = prim::GetAttr[name="ffn"]
%ffn_norm.1 : torch.RMSNorm = prim::GetAttr[name="ffn_norm"]
%29 : Tensor = prim::CallMethod[name="forward"](%ffn_norm.1, %x4.1) #
C:\Users\Terox\Desktop\temp\gen.py:747:25
%30 : Tensor = prim::CallMethod[name="forward"](%ffn.1, %29) #
C:\Users\Terox\Desktop\temp\gen.py:747:16
%x5.1 : Tensor = aten::add(%x4.1, %30, %38) # C:\Users\Terox\Desktop\temp\gen.py:747:12
%37 : Tensor = aten::select(%tokens.1, %36, %36) #
C:\Users\Terox\Desktop\temp\gen.py:748:19
%39 : Tensor = aten::select(%37, %36, %38) # C:\Users\Terox\Desktop\temp\gen.py:748:19
%41 : Scalar = aten::item(%39) # C:\Users\Terox\Desktop\temp\gen.py:748:19
%44 : int = aten::size(%tokens.1, %38) # C:\Users\Terox\Desktop\temp\gen.py:748:49
%45 : int = aten::sub(%44, %38) # C:\Users\Terox\Desktop\temp\gen.py:748:49
%46 : int = aten::mul(%42, %45) # C:\Users\Terox\Desktop\temp\gen.py:748:42
%50 : bool = aten::eq(%41, %46) # C:\Users\Terox\Desktop\temp\gen.py:748:19
%51 : (Tensor, bool) = prim::TupleConstruct(%x5.1, %50)
return (%51)
```

这里分析 block1: 同样分析语法可以得到以下式子

$\%46 = \%42 * \%45$

$\%45 = \%44 - \%38$

$\%44 = 4, \%38 = 1, \%45 = 1$ 这里带入即可得到输入应为 442

Block 2 graph:

```
graph(%self.1 : torch.Block_2,
%x.1 : Tensor,
%start_pos.1 : int,
%freqs_cis.1 : Tensor,
%mask.1 : Tensor?,
%tokens.1 : Tensor):
%36 : int = prim::Constant[value=0] # C:\Users\Terox\Desktop\temp\gen.py:761:26
%38 : int = prim::Constant[value=2] # C:\Users\Terox\Desktop\temp\gen.py:761:29
%47 : int = prim::Constant[value=1] # C:\Users\Terox\Desktop\temp\gen.py:761:68
%54 : int = prim::Constant[value=4011] # C:\Users\Terox\Desktop\temp\gen.py:761:81
%attn.1 : torch.MLA = prim::GetAttr[name="attn"]
%attn_norm.1 : torch.RMSNorm = prim::GetAttr[name="attn_norm"]
%13 : Tensor = prim::CallMethod[name="forward"](%attn_norm.1, %x.1) #
C:\Users\Terox\Desktop\temp\gen.py:759:26
%17 : Tensor =
prim::CallMethod[name="forward"](%attn.1, %13, %start_pos.1, %freqs_cis.1, %mask.1) #
C:\Users\Terox\Desktop\temp\gen.py:759:16
%x6.1 : Tensor = aten::add(%x.1, %17, %47) # C:\Users\Terox\Desktop\temp\gen.py:759:12
%ffn.1 : torch.MoE = prim::GetAttr[name="ffn"]
%ffn_norm.1 : torch.RMSNorm = prim::GetAttr[name="ffn_norm"]
%29 : Tensor = prim::CallMethod[name="forward"](%ffn_norm.1, %x6.1) #
C:\Users\Terox\Desktop\temp\gen.py:760:25
%30 : Tensor = prim::CallMethod[name="forward"](%ffn.1, %29) #
C:\Users\Terox\Desktop\temp\gen.py:760:16
%x7.1 : Tensor = aten::add(%x6.1, %30, %47) # C:\Users\Terox\Desktop\temp\gen.py:760:12
%37 : Tensor = aten::select(%tokens.1, %36, %36) #
C:\Users\Terox\Desktop\temp\gen.py:761:19
%39 : Tensor = aten::select(%37, %36, %38) # C:\Users\Terox\Desktop\temp\gen.py:761:19
%41 : Scalar = aten::item(%39) # C:\Users\Terox\Desktop\temp\gen.py:761:19
%43 : Tensor = aten::select(%tokens.1, %36, %36) #
C:\Users\Terox\Desktop\temp\gen.py:761:43
%44 : Tensor = aten::select(%43, %36, %36) # C:\Users\Terox\Desktop\temp\gen.py:761:43
%46 : Tensor = aten::select(%tokens.1, %36, %36) #
C:\Users\Terox\Desktop\temp\gen.py:761:58
%48 : Tensor = aten::select(%46, %36, %47) # C:\Users\Terox\Desktop\temp\gen.py:761:58
%52 : Tensor = aten::add(%44, %48, %47) # C:\Users\Terox\Desktop\temp\gen.py:761:43
%53 : Scalar = aten::item(%52) # C:\Users\Terox\Desktop\temp\gen.py:761:43
%55 : Scalar = aten::add(%53, %54) # C:\Users\Terox\Desktop\temp\gen.py:761:43
%59 : bool = aten::eq(%41, %55) # C:\Users\Terox\Desktop\temp\gen.py:761:19
%60 : (Tensor, bool) = prim::TupleConstruct(%x7.1, %59)
return (%60)
```

同理得到式子： $402 + 34 + 1 + 4011 = 4448$ 这里不知道为什么输出是 false，但是调整成 4447 就好了

Block 3 graph:

```
graph(%self.1 : torch.Block_3,
%x.1 : Tensor,
%start_pos.1 : int,
%freqs_cis.1 : Tensor,
%mask.1 : Tensor?,
%tokens.1 : Tensor):
%39 : NoneType = prim::Constant()
%38 : bool = prim::Constantvalue=0
%20 : int = prim::Constantvalue=1
%36 : int = prim::Constantvalue=-1 # C:\Users\Terox\Desktop\temp\gen.py:774:34
%42 : int = prim::Constantvalue=30315 # C:\Users\Terox\Desktop\temp\gen.py:774:48
%attn.1 : torch.MLA = prim::GetAttrname="attn"
%attn_norm.1 : torch.RMSNorm = prim::GetAttrname="attn_norm"
%13 : Tensor = prim::CallMethod[name="forward"](%attn_norm.1, %x.1) #
C:\Users\Terox\Desktop\temp\gen.py:772:26
%17 : Tensor =
prim::CallMethod[name="forward"](%attn.1, %13, %start_pos.1, %freqs_cis.1, %mask.1) #
C:\Users\Terox\Desktop\temp\gen.py:772:16
%x9.1 : Tensor = aten::add(%x.1, %17, %20) # C:\Users\Terox\Desktop\temp\gen.py:772:12
%ffn.1 : torch.MoE = prim::GetAttrname="ffn"
%ffn_norm.1 : torch.RMSNorm = prim::GetAttrname="ffn_norm"
%29 : Tensor = prim::CallMethod[name="forward"](%ffn_norm.1, %x9.1) #
C:\Users\Terox\Desktop\temp\gen.py:773:2
%30 : Tensor = prim::CallMethod[name="forward"](%ffn.1, %29) #
C:\Users\Terox\Desktop\temp\gen.py:773:16
%x10.1 : Tensor = aten::add(%x9.1, %30, %20) # C:\Users\Terox\Desktop\temp\gen.py:773:12
%37 : int[] = prim::ListConstruct(%36)
%40 : Tensor = aten::sum(%tokens.1, %37, %38, %39) #
C:\Users\Terox\Desktop\temp\gen.py:774:19
%41 : Scalar = aten::item(%40) # C:\Users\Terox\Desktop\temp\gen.py:774:19
%43 : bool = aten::eq(%41, %42) # C:\Users\Terox\Desktop\temp\gen.py:774:19
%46 : (Tensor, bool) = prim::TupleConstruct(%x10.1, %43)
return (%46)
```

这里就是前三个数字加上最后一个数字需要 SUM= 25432

简单计算可得最后数字是 25422