

```
1 ssh://user43@202.112.195.5:22/home/  
  user43/.conda/envs/torch1.9/bin/  
  python -u /home/user43/workspace/  
  Cross_LangVC_projs/CL_ADAIN/Models.py  
2 AE(  
3     (speaker_encoder): SpeakerEncoder(  
4         (act): ReLU()  
5         (conv_bank): ModuleList(  
6             (0): Conv1d(80, 128,  
kernel_size=(1,), stride=(1,))  
7             (1): Conv1d(80, 128,  
kernel_size=(2,), stride=(1,))  
8             (2): Conv1d(80, 128,  
kernel_size=(3,), stride=(1,))  
9             (3): Conv1d(80, 128,  
kernel_size=(4,), stride=(1,))  
10            (4): Conv1d(80, 128,  
kernel_size=(5,), stride=(1,))  
11            (5): Conv1d(80, 128,  
kernel_size=(6,), stride=(1,))  
12            (6): Conv1d(80, 128,  
kernel_size=(7,), stride=(1,))  
13            (7): Conv1d(80, 128,  
kernel_size=(8,), stride=(1,))  
14        )  
15        (in_conv_layer): Conv1d(1104, 128  
    , kernel_size=(1,), stride=(1,))  
16        (first_conv_layers): ModuleList(  
17            (0): Conv1d(128, 128,  
kernel_size=(5,), stride=(1,))  
18            (1): Conv1d(128, 128,
```

```
18 kernel_size=(5,), stride=(1,))
19     (2): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
20     (3): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
21     (4): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
22     (5): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
23 )
24     (second_conv_layers): ModuleList(
25         (0): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
26         (1): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
27         (2): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
28         (3): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
29         (4): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
30         (5): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
31     )
32     (pooling_layer):
    AdaptiveAvgPool1d(output_size=1)
33     (first_dense_layers): ModuleList(
34         (0): Linear(in_features=128,
    out_features=128, bias=True)
35         (1): Linear(in_features=128,
    out_features=128, bias=True)
```

```
36         (2): Linear(in_features=128,
out_features=128, bias=True)
37         (3): Linear(in_features=128,
out_features=128, bias=True)
38         (4): Linear(in_features=128,
out_features=128, bias=True)
39         (5): Linear(in_features=128,
out_features=128, bias=True)
40     )
41     (second_dense_layers): ModuleList
(
42         (0): Linear(in_features=128,
out_features=128, bias=True)
43         (1): Linear(in_features=128,
out_features=128, bias=True)
44         (2): Linear(in_features=128,
out_features=128, bias=True)
45         (3): Linear(in_features=128,
out_features=128, bias=True)
46         (4): Linear(in_features=128,
out_features=128, bias=True)
47         (5): Linear(in_features=128,
out_features=128, bias=True)
48     )
49     (output_layer): Linear(
in_features=128, out_features=128,
bias=True)
50     (dropout_layer): Dropout(p=0,
inplace=False)
51 )
52     (content_encoder): ContentEncoder(
```

```
53         (act): ReLU()
54         (conv_bank): ModuleList(
55             (0): Conv1d(80, 128,
kernel_size=(1,), stride=(1,))
56             (1): Conv1d(80, 128,
kernel_size=(2,), stride=(1,))
57             (2): Conv1d(80, 128,
kernel_size=(3,), stride=(1,))
58             (3): Conv1d(80, 128,
kernel_size=(4,), stride=(1,))
59             (4): Conv1d(80, 128,
kernel_size=(5,), stride=(1,))
60             (5): Conv1d(80, 128,
kernel_size=(6,), stride=(1,))
61             (6): Conv1d(80, 128,
kernel_size=(7,), stride=(1,))
62             (7): Conv1d(80, 128,
kernel_size=(8,), stride=(1,))
63         )
64         (in_conv_layer): Conv1d(1104, 128
, kernel_size=(1,), stride=(1,))
65         (first_conv_layers): ModuleList(
66             (0): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
67             (1): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
68             (2): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
69             (3): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
70             (4): Conv1d(128, 128,
```

```
70 kernel_size=(5,), stride=(1,))
71     (5): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
72 )
73     (second_conv_layers): ModuleList
74     (
75         (0): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
76         (1): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
77         (2): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
78         (3): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
79         (4): Conv1d(128, 128,
    kernel_size=(5,), stride=(1,))
80         (5): Conv1d(128, 128,
    kernel_size=(5,), stride=(2,))
81     )
82     (norm_layer): InstanceNorm1d(128
    , eps=1e-05, momentum=0.1, affine=
    False, track_running_stats=False)
83     (mean_layer): Conv1d(128, 128,
    kernel_size=(1,), stride=(1,))
84     (std_layer): Conv1d(128, 128,
    kernel_size=(1,), stride=(1,))
85     (dropout_layer): Dropout(p=0,
    inplace=False)
86 )
87     (decoder): Decoder(
    (act): ReLU()
```

```
88     (in_conv_layer): Conv1d(128, 128
, kernel_size=(1,), stride=(1,))
89     (first_conv_layers): ModuleList(
90         (0): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
91         (1): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
92         (2): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
93         (3): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
94         (4): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
95         (5): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
96     )
97     (second_conv_layers): ModuleList
(
98         (0): Conv1d(128, 256,
kernel_size=(5,), stride=(1,))
99         (1): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
100        (2): Conv1d(128, 256,
kernel_size=(5,), stride=(1,))
101        (3): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
102        (4): Conv1d(128, 256,
kernel_size=(5,), stride=(1,))
103        (5): Conv1d(128, 128,
kernel_size=(5,), stride=(1,))
104    )
```

```
105     (norm_layer): InstanceNorm1d(128
    , eps=1e-05, momentum=0.1, affine=
    False, track_running_stats=False)
106     (conv_affine_layers): ModuleList
    (
107         (0): Linear(in_features=128,
    out_features=256, bias=True)
108         (1): Linear(in_features=128,
    out_features=256, bias=True)
109         (2): Linear(in_features=128,
    out_features=256, bias=True)
110         (3): Linear(in_features=128,
    out_features=256, bias=True)
111         (4): Linear(in_features=128,
    out_features=256, bias=True)
112         (5): Linear(in_features=128,
    out_features=256, bias=True)
113         (6): Linear(in_features=128,
    out_features=256, bias=True)
114         (7): Linear(in_features=128,
    out_features=256, bias=True)
115         (8): Linear(in_features=128,
    out_features=256, bias=True)
116         (9): Linear(in_features=128,
    out_features=256, bias=True)
117         (10): Linear(in_features=128,
    out_features=256, bias=True)
118         (11): Linear(in_features=128,
    out_features=256, bias=True)
119     )
120     (out_conv_layer): Conv1d(128, 80
```

```
120 , kernel_size=(1,), stride=(1,))
121     (dropout_layer): Dropout(p=0,
    inplace=False)
122 )
123 )
124 Model adain, the number of parameters
    : 4892880
125 #####
126 input mel torch.Size([1, 80, 128])
127 tar mel torch.Size([1, 80, 160])
128 l1 loss 0.7275683879852295
129 dec mel: torch.Size([1, 80, 128])
130 mu torch.Size([1, 128, 16])
131 log_sigma torch.Size([1, 128, 16])
132 emb torch.Size([1, 128])
133 #####infe
134 infe mel, torch.Size([1, 80, 128])
135
136 Process finished with exit code 0
137
```