*# input\_sample 示例*

input\_sample = {

'target\_ids': tensor([ 101, 2345, 3456, ..., 0]), *# 填充后的目标ID序列*

'sentiment\_label': tensor(2), *# 情感标签（2表示正面）*

'word\_contents': tensor([ 101, 1234, 5678, ..., 0]), *# 单词内容的分词ID*

'input\_embeddings': tensor([[...], ..., [...]], dtype=torch.float32), *# EEG特征张量（max\_len × 105\*8）*

'input\_attn\_mask': tensor([1, 1, ..., 0]), *# 有效部分为1，填充为0*

'input\_attn\_mask\_invert': tensor([0, 0, ..., 1]), *# 有效部分为0，填充为1*

'target\_mask': tensor([1, 1, ..., 0]), *# 目标掩码（用于损失计算）*

'seq\_len': 20, *# 句子实际长度*

'subject': 'ZAB' *# 受试者标识*

}

*# sent\_obj 示例*

sent\_obj = {

'content': 'The movie is fantastic.',

'sentence\_level\_EEG': {

'mean\_t1': np.array([...]),

'mean\_t2': np.array([...]),

...

},

'word': [

{'content': 'The', 'nFixations': 2, 'word\_level\_EEG': {...}, 'rawEEG': [...]},

...

],

'word\_tokens\_has\_fixation': ['The', 'movie', 'fantastic'],

'word\_tokens\_with\_mask': ['The', 'movie', '[MASK]', 'fantastic'],

'word\_tokens\_all': ['The', 'movie', 'is', 'fantastic']

}

input\_masks\_batch shape after stack: torch.Size([1, 56])

Input embeddings batch shape: torch.Size([1, 56, 105, 24])

Conv module output shape: torch.Size([1, 56, 1024])

添加位置 编码后：torch.Size([1，56，1024])

Before encoder - brain\_embedding shape: torch.Size([1, 56, 1024])

Before encoder - input\_masks\_invert shape: torch.Size([1, 56])

After encoder: torch.Size([1, 56, 1024])

After layer norm: torch.Size([1, 56,1024])

input\_masks\_batch shape after stack: torch.Size([4, 56])

Input embeddings batch shape: torch.Size([4, 56, 105, 24])

Conv module output shape: torch.Size([4, 56, 1024])

添加位置 编码后：torch.Size([4，56，1024])

Before encoder - brain\_embedding shape: torch.Size([4, 56, 1024])

Before encoder - input\_masks\_invert shape: torch.Size([4, 56])

After encoder: torch.Size([4, 56, 1024])

After layer norm: torch.Size([4, 56, 1024])