Masking in Sequence Models

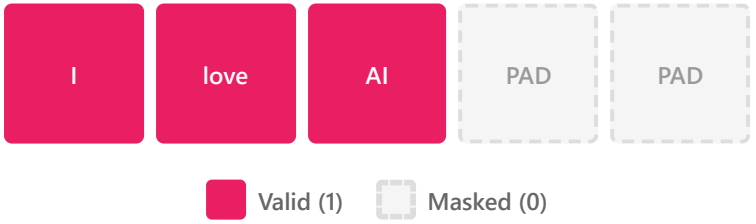
Why Do We Need Masking?

Masks prevent the model from attending to **invalid positions** like padding tokens or future tokens, ensuring correct computation and preventing information leakage.

Padding Mask

Ignore PAD tokens in attention computation

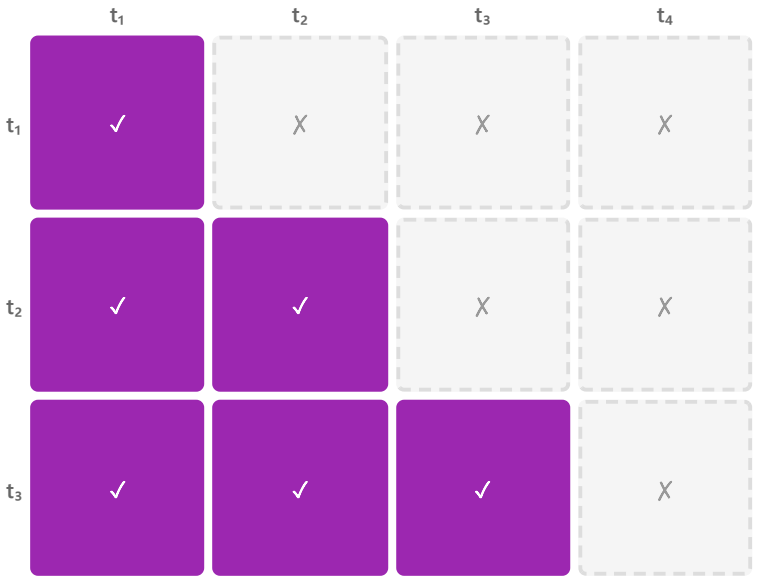
Example: Sequence with padding

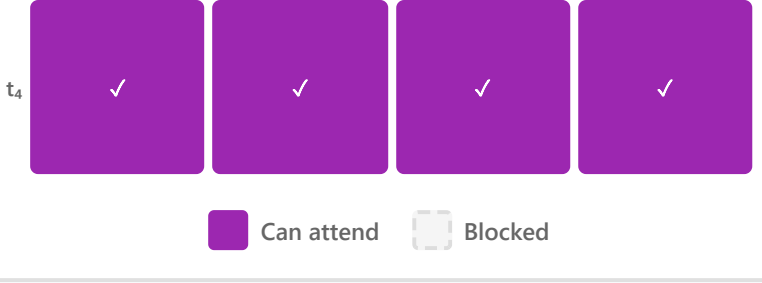


Look-ahead Mask

Prevent attending to future tokens (causal)

Attention mask matrix (lower triangular)






```
mask = (input != PAD_TOKEN)
scores.masked_fill(~mask, -∞)
```

```
mask = torch.tril(
torch.ones(seq_len, seq_len))
```

① Create Mask

```
# Padding mask
pad_mask = (x != PAD)

# Shape: (batch, seq_len)
```

② Apply to Attention

```
# Before softmax
scores = scores.masked_fill(
  ~mask, float('-inf'))
attn = softmax(scores)
```

③ Combine Masks

```
# Both padding + lookahead
mask = pad_mask &
       lookahead_mask
# Logical AND
```