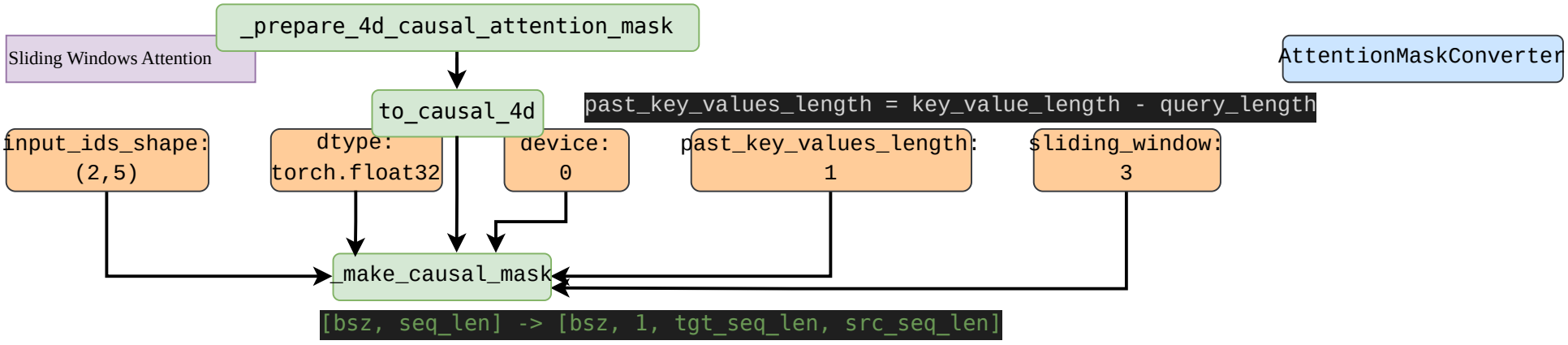


```
bsz, tgt_len = input_ids_shape
```

```
mask = torch.full((tgt_len, tgt_len), torch.finfo(dtype).min,device=device)
```

```
tensor([[-3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [-3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [-3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [-3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [-3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38]],
       device='cuda:0')
```

```
mask_cond = torch.arange(mask.size(-1), device=device)
```

```
tensor([0, 1, 2, 3, 4], device='cuda:0')
```

```
mask.masked_fill_(mask_cond < (mask_cond + 1).view(mask.size(-1), 1), 0)
```

```
tensor([[ 0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00]],
       device='cuda:0')
```

```
mask = mask.to(dtype)
```

```
if past_key_values_length > 0:
    mask = torch.cat([torch.zeros(tgt_len,
                                  past_key_values_length,
                                  dtype=dtype,
                                  device=device), mask], dim=-1)
```

```
tensor([[ 0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00]],
       device='cuda:0')
```

```
if sliding_window is not None:
    diagonal = past_key_values_length - sliding_window - 1
    context_mask = torch.tril(torch.ones_like(mask, dtype=torch.bool),
                              diagonal=diagonal)
    mask.masked_fill_(context_mask, torch.finfo(dtype).min)
```

```
tensor([[ 0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38, -3.4028e+38],
        [-3.4028e+38,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00, -3.4028e+38],
        [-3.4028e+38, -3.4028e+38,  0.0000e+00,  0.0000e+00,  0.0000e+00,  0.0000e+00]],
       device='cuda:0')
```

```
return mask[None, None, :, :].expand(bsz, 1, tgt_len, tgt_len + past_key_values_length)
```