```
prepare 4d causal attention mask
                                                                                                                                                                                                                                   AttentionMaskConverter
Sliding Windows Attention
                                                                                                      past_key_values_length = key_value_length - query_length
                                                                  to causal 4d
input_ids_shape:
                                                      dtype:
                                                                                                                        past_key_values_length:
                                                                                                                                                                                     $liding_window:
                                                                                           device:
            (2,5)
                                               torch.float32
                                                                                                  0
                                                                                                                                                                                                      3
                                                      →_make_causal_mask
                                                         [bsz, seg len] -> [bsz, 1, tgt seg len, src seg len
bsz, tgt_len = input_ids_shape
                                                                                                                                                                      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],
mask = torch.full((tgt_len, tgt_len), torch.finfo(dtype).min,device=device)
                                                                                                                                                                                [-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')
                                                                                                                                                                      tensor([0, 1, 2, 3, 4], device='cuda:0')
mask_cond = torch.arange(mask.size(-1), device=device)
                                                                                                                                                                      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],
mask.masked_fill_(mask_cond < (mask_cond + 1).view(mask.size(-1), 1), 0)</pre>
                                                                                                                                                                                [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)
                                                                                                                                                                      tensor([[\ 0.0000e+00, \ \ ]\ 0.0000e+00, \ -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
if past_key_values_length > 0:
                                                                                                                                                                                 [0.0000e+00, 0.0000e+00, 0.0000e+00, -3.4028e+38, -3.4028e+38, -3.4028e+38],
         mask = torch.cat([torch.zeros(tgt_len,
                                                                                                                                                                                [0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, -3.4028e+38, -3.4028e+38],
                                                                               past_key_values_length,
                                                                                                                                                                                [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]]
                                                                               dtype=dtype,
                                                                                                                                                                             device='cuda:0')
                                                                               device=device), mask], dim=-1)
if sliding_window is not None:
                                                                                                                                                                      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]
         diagonal = past_key_values_length - sliding_window - 1
                                                                                                                                                                                [0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, -3.4028e+38, -3.4028e+38],
         context_mask = torch.tril(torch.ones_like(mask, dtype=torch.bool),
                                                                                                                                                                                [-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]
                                                                     diagonal=diagonal)
                                                                                                                                                                                device='cuda:0')
         mask.masked_fill_(context_mask, torch.finfo(dtype).min)
```

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