```
import torch
import torch.nn.functional as F
```

```
feats_a_tensor = torch.tensor([[0.1, 0.2],
                              [0.3, 0.4],
                              [0.5, 0.6]])

feats_b_tensor = torch.tensor([[0.7, 0.8],
                              [0.9, 1.0],
                              [1.1, 1.2]])

feats_c_tensor = torch.tensor([[1.3, 1.4],
                              [1.5, 1.6],
                              [1.7, 1.8]])


feats_a_b_c_tensor = torch.cat([feats_a_tensor, feats_b_tensor, feats_c_tensor], dim=0)
feats_tensors = [feats_a_b_c_tensor]
temperatures = [0.07]
```

```
dual_nll = False
for feats_idx, feats_tensor in enumerate(feats_tensors):
    cos_sim = F.cosine_similarity(feats_tensor[:, None, :], feats_tensor[None, :, :], dim=-1)
    self_mask = torch.eye(cos_sim.shape[0], dtype=torch.bool, device="cpu")
    cos_sim.masked_fill_(self_mask, -9e15)
    #pos_mask = self_mask.roll(shifts=cos_sim.shape[0] // 3, dims=0)
    pos_mask_1 = self_mask.roll(shifts=batch_size//3, dims=0)
    pos_mask_2 = self_mask.roll(shifts=2 * batch_size//3, dims=0)
    pos_mask = pos_mask_1 | pos_mask_2
    cos_sim = cos_sim / temperatures[feats_idx]
    nll = -cos_sim[pos_mask] + torch.logsumexp(cos_sim, dim=-1)
    nll = nll.mean()
    if not dual_nll:
        dual_nll = nll
    else:
        dual_nll += nll
        dual_nll /= 2
```

```
cos_sim.shape
```

```
torch.Size([9, 9])
```

```
pos_mask
```

```
tensor([[False, False, False, False, False, False,  True, False, False],
        [False, False, False, False, False, False, False,  True, False],
        [False, False, False, False, False, False, False, False,  True],
        [ True, False, False, False, False, False, False, False, False],
        [False,  True, False, False, False, False, False, False, False],
        [False, False,  True, False, False, False, False, False, False],
        [False, False, False,  True, False, False, False, False, False],
        [False, False, False, False,  True, False, False, False, False],
        [False, False, False, False, False,  True, False, False, False]])
```

```
cos_sim
```

```
tensor([[-1.2857e+17,  1.4055e+01,  1.3906e+01,  1.3823e+01,  1.3771e+01,
          1.3736e+01,  1.3711e+01,  1.3691e+01,  1.3676e+01],
        [ 1.4055e+01, -1.2857e+17,  1.4267e+01,  1.4245e+01,  1.4229e+01,
          1.4217e+01,  1.4207e+01,  1.4200e+01,  1.4194e+01],
        [ 1.3906e+01,  1.4267e+01, -1.2857e+17,  1.4282e+01,  1.4275e+01,
          1.4270e+01,  1.4265e+01,  1.4261e+01,  1.4258e+01],
        [ 1.3823e+01,  1.4245e+01,  1.4282e+01, -1.2857e+17,  1.4284e+01,
          1.4282e+01,  1.4279e+01,  1.4277e+01,  1.4275e+01],
        [ 1.3771e+01,  1.4229e+01,  1.4275e+01,  1.4284e+01, -1.2857e+17,
          1.4285e+01,  1.4284e+01,  1.4283e+01,  1.4282e+01],
        [ 1.3736e+01,  1.4217e+01,  1.4270e+01,  1.4282e+01,  1.4285e+01,
         -1.2857e+17,  1.4285e+01,  1.4285e+01,  1.4284e+01],
        [ 1.3711e+01,  1.4207e+01,  1.4265e+01,  1.4279e+01,  1.4284e+01,
          1.4285e+01, -1.2857e+17,  1.4286e+01,  1.4285e+01],
        [ 1.3691e+01,  1.4200e+01,  1.4261e+01,  1.4277e+01,  1.4283e+01,
          1.4285e+01,  1.4286e+01, -1.2857e+17,  1.4286e+01],
        [ 1.3676e+01,  1.4194e+01,  1.4258e+01,  1.4275e+01,  1.4282e+01,
          1.4284e+01,  1.4285e+01,  1.4286e+01, -1.2857e+17]])
```

```
batch_size = 9
pos_mask_1 = self_mask.roll(shifts=batch_size//3, dims=0)
pos_mask_2 = self_mask.roll(shifts=2 * batch_size//3, dims=0)
pos_mask = pos_mask_1 | pos_mask_2
```

```
pos_mask
```

```
tensor([[False, False, False,  True, False, False,  True, False, False],
        [False, False, False, False,  True, False, False,  True, False],
        [False, False, False, False, False,  True, False, False,  True],
        [ True, False, False, False, False, False,  True, False, False],
        [False,  True, False, False, False, False, False,  True, False],
        [False, False,  True, False, False, False, False, False,  True],
        [ True, False, False,  True, False, False, False, False, False],
        [False,  True, False, False,  True, False, False, False, False],
        [False, False,  True, False, False,  True, False, False, False]])
```

```
cos_sim[pos_mask]
```

```
tensor([13.8231, 13.7105, 14.2288, 14.1999, 14.2698, 14.2582, 13.8231, 14.2795,
        14.2288, 14.2828, 14.2698, 14.2841, 13.7105, 14.2795, 14.1999, 14.2828,
        14.2582, 14.2841])
```