```python
''' Minimum Shifted and Masked Distance '''
def mmsd(self, f1, f2, mask1, mask2):
    batch_size = f1.shape[0]
    fd_set = torch.zeros(size=(17, batch_size))

    for shifts in range(-8, 9):
        f1_s = self.shiftbits(f1, shifts)
        mask1_s = self.shiftbits(mask1, shifts)

        fd_set[shifts + 8] = self.fd(f1_s, f2, mask1_s, mask2)

    batch_min_fd = torch.min(fd_set, dim=0)

    return batch_min_fd.values, batch_min_fd.indices - 8
```

```python
def forward(self, fp, fa, fn, fp_mask, fa_mask, fn_mask):
    mmsd_fa_fp, offset_ap = self.mmsd(fa[:,0,:,:], fp[:,0,:,:], fa_mask, fp_mask)
    mmsd_fa_fn, offset_an = self.mmsd(fa[:,0,:,:], fn[:,0,:,:], fa_mask, fn_mask)

    etl_loss = mmsd_fa_fp - mmsd_fa_fn + self.alpha

    zero = torch.tensor(0.)
    etl_loss = torch.maximum(etl_loss, zero)
    etl_loss = torch.mean(etl_loss)

    return etl_loss, offset_ap, offset_an
```