

```
def shiftbits(self, fa, noshifts):
```

```
    fnew = fa.clone()
```

```
    width = fa.shape[2]
```

```
    s = 2 * np.abs(noshifts)
```

```
    p = width - s
```

```
    # Shift
```

```
    if noshifts == 0:
```

```
        return fa
```

```
    elif noshifts < 0:
```

```
        fnew[:, :, 0:p] = fa[:, :, s:p + s]
```

```
        fnew[:, :, p:width] = fa[:, :, 0:s]
```

```
    else:
```

```
        fnew[:, :, s:width] = fa[:, :, 0:p]
```

```
        fnew[:, :, 0:s] = fa[:, :, p:width]
```

```
    return fnew
```

```
''' Fractional Distance '''
```

```
def fd(self, f1, f2, mask1, mask2):
```

```
    batch_size = f1.shape[0]
```

```
    batch_fd = torch.zeros(size=(batch_size, ))
```

```
    zero = torch.tensor(0.).to(self.device)
```

```
    for i in range(batch_size):
```

```
        M = torch.sum((mask1[i] == mask2[i]) & (mask1[i] == 1))
```

```
        fd = torch.where(
            ((mask1[i] == mask2[i]) & (mask1[i] == 1)),
            torch.square(f1[i] - f2[i]),
            zero)
```

```
        fd = torch.sum(fd) / M
```

```
        batch_fd[i] = fd
```

```
    return batch_fd
```