Explanation of Data Difference Check (compute_kl_divergence)

This section explains the function used to compare data distributions using KL divergence.

```
def compute_kl_divergence(data1, data2, bins=100):
    data1_flat = data1.view(-1).cpu().numpy()
    data2_flat = data2.view(-1).cpu().numpy()
    hist1, _ = np.histogram(data1_flat, bins=bins, density=True)
    hist2, _ = np.histogram(data2_flat, bins=bins, density=True)
    return entropy(hist1 + 1e-10, hist2 + 1e-10)
```

Line-by-line explanation:

- This function computes the Kullback-Leibler (KL) divergence between two data tensors.
- data1_flat = data1.view(-1).cpu().numpy(): Flattens the first tensor, moves it to CPU, and converts it to a NumPy array.
- data2_flat = data2.view(-1).cpu().numpy(): Flattens the second tensor, moves it to CPU, and converts it to a NumPy array.
- hist1, _ = np.histogram(data1_flat, bins=bins, density=True): Computes a normalized histogram (probability distribution) for the first data array.
- hist2, _ = np.histogram(data2_flat, bins=bins, density=True): Computes a normalized histogram for the second data array.
- return entropy(hist1 + 1e-10, hist2 + 1e-10): Calculates the KL divergence between the two histograms, adding a small value (1e-10) to avoid division by zero.

Purpose:

- KL divergence is a measure of how one probability distribution diverges from a second, expected probability distribution.

- This function is used to compare the distributions of two datasets (e.g., features or images) to detect significant differences.
- A higher KL divergence indicates greater difference between the two distributions.