Copilot

Yes, the axes in the `np.tensordot` function for `gradient_case1` and `gradient_case2` are correct. Here's why:

In the `np.tensordot` function, the `axes` argument specifies the axes to sum over. In this case, `axes=([0, 1], [1, 2])` means that we're summing over the 0th and 1st dimensions of `difference_case1` (or `difference_case2`), and the 1st and 2nd dimensions of `U1T_case1` (or `U1T_case2`).

Given that `difference_case1` is of shape `(1, 11088)` and `U1T_case1` is of shape `(32, 1, 11088)`, the specified axes align correctly. The same applies to `difference_case2` and `U1T_case2`.

So, after the `np.tensordot` operation, the resulting `gradient_case1` and `gradient_case2` are both of shape `(32,)`, which is the desired shape.

I hope this clarifies your question! Let me know if you have any other questions. 😊