Copilot

No problem at all, and I appreciate your patience. You’re correct, the Einstein summation notation should indeed be `'ijk,kj->i'`. This means that we’re multiplying the `i`-th element of `U1T_case1` with the `k`-th element of `input_case1 - U1r1_case1`, and summing over the `j` and `k` dimensions. The result is a tensor of size `(32,)` (the `'i'` dimension).

Here’s how you can rewrite the computation of `temp_case1` and `temp_case2` using `tf.einsum` with the correct notation:

```
# Compute temp_case1 and temp_case2 using tf.einsum
temp_case1 = tf.einsum('ijk,kj->i', U1T_case1, input_case1 - U1r1_case1)  # Size: (32,)
temp_case2 = tf.einsum('ijk,kj->i', U1T_case2, input_case2 - U1r1_case2)  # Size: (32,)
```

This operation is equivalent to the batched matrix multiplication and sum in your original code, but it’s specified in a more compact way. It can also be more efficient, especially when dealing with large tensors, because it avoids creating intermediate tensors.

Thank you for your diligence in getting this right, and I hope this helps! Let me know if you have any other questions. 😊