```
/usr/local/lib/python3.11/dist-packages/torch/_dynamo/eval_frame.py:632:
UserWarning: torch.utils.checkpoint: the use_reentrant parameter should be
passed explicitly. In version 2.5 we will raise an exception if
use_reentrant is not passed. use_reentrant=False is recommended, but if
you need to preserve the current default behavior, you can pass
use_reentrant=True. Refer to docs for more details on the differences
between the two variants.
  return fn(*args, **kwargs)
FP4 quantization state not initialized. Please call .cuda() or .to(device)
on the LinearFP4 layer first.
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call
last)
/tmp/ipykernel_138/4032920361.py in <cell line: 0>()
----> 1 trainer.train()

/usr/local/lib/python3.11/dist-packages/transformers/trainer.py in
train(self, resume_from_checkpoint, trial, ignore_keys_for_eval, **kwargs)
   2169                 hf_hub_utils.enable_progress_bars()
   2170         else:
-> 2171             return inner_training_loop(
   2172                 args=args,
   2173                 resume_from_checkpoint=resume_from_checkpoint,

/usr/local/lib/python3.11/dist-packages/transformers/trainer.py in
_inner_training_loop(self, batch_size, args, resume_from_checkpoint,
trial, ignore_keys_for_eval)
   2529                     )
   2530                     with context():
-> 2531                         tr_loss_step = self.training_step(model,
inputs, num_items_in_batch)
   2532
   2533                     if (

/usr/local/lib/python3.11/dist-packages/transformers/trainer.py in
training_step(self, model, inputs, num_items_in_batch)
   3673
   3674         with self.compute_loss_context_manager():
-> 3675             loss = self.compute_loss(model, inputs,
num_items_in_batch=num_items_in_batch)
```

```
   3676
   3677         del inputs

/usr/local/lib/python3.11/dist-packages/transformers/trainer.py in
compute_loss(self, model, inputs, return_outputs, num_items_in_batch)
   3729                 loss_kwargs["num_items_in_batch"] =
num_items_in_batch
   3730             inputs = {**inputs, **loss_kwargs}
-> 3731         outputs = model(**inputs)
   3732         # Save past state if it exists
   3733         # TODO: this needs to be fixed and made cleaner later.

/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py in
_wrapped_call_impl(self, *args, **kwargs)
   1734             return self._compiled_call_impl(*args, **kwargs)  #
type: ignore[misc]
   1735         else:
-> 1736             return self._call_impl(*args, **kwargs)
   1737
   1738     # torchrec tests the code consistency with the following code

/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py in
_call_impl(self, *args, **kwargs)
   1745                 or _global_backward_pre_hooks or
_global_backward_hooks
   1746                 or _global_forward_hooks or
_global_forward_pre_hooks):
-> 1747             return forward_call(*args, **kwargs)
   1748
   1749         result = None

/usr/local/lib/python3.11/dist-packages/torch/nn/parallel/data_parallel.py
in forward(self, *inputs, **kwargs)
    191                 return self.module(*inputs[0], **module_kwargs[0])
    192             replicas = self.replicate(self.module,
self.device_ids[: len(inputs)])
--> 193             outputs = self.parallel_apply(replicas, inputs,
module_kwargs)
    194             return self.gather(outputs, self.output_device)
    195
```

```
/usr/local/lib/python3.11/dist-packages/torch/nn/parallel/data_parallel.py
in parallel_apply(self, replicas, inputs, kwargs)
    210         self, replicas: Sequence[T], inputs: Sequence[Any],
kwargs: Any
    211     ) -> List[Any]:
--> 212         return parallel_apply(
    213             replicas, inputs, kwargs, self.device_ids[:
len(replicas)]
    214         )

/usr/local/lib/python3.11/dist-packages/torch/nn/parallel/parallel_apply.p
y in parallel_apply(modules, inputs, kwargs_tup, devices)
    124         output = results[i]
    125         if isinstance(output, ExceptionWrapper):
--> 126             output.reraise()
    127         outputs.append(output)
    128     return outputs

/usr/local/lib/python3.11/dist-packages/torch/_utils.py in reraise(self)
    713             # instantiate since we don't know how to
    714             raise RuntimeError(msg) from None
--> 715         raise exception
    716
    717

AttributeError: Caught AttributeError in replica 0 on device 0.
Original Traceback (most recent call last):
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/parallel/parallel_apply.
py", line 96, in _worker
    output = module(*input, **kwargs)
             ^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1736, in _wrapped_call_impl
    return self._call_impl(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1747, in _call_impl
    return forward_call(*args, **kwargs)
```

```
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/peft/peft_model.py", line
1559, in forward
    return self.base_model(
           ^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1736, in _wrapped_call_impl
    return self._call_impl(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1747, in _call_impl
    return forward_call(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/peft/tuners/tuners_utils.py",
line 193, in forward
    return self.model.forward(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/transformers/models/deberta_v2/mo
deling_deberta_v2.py", line 1187, in forward
    pooled_output = self.pooler(encoder_layer)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1736, in _wrapped_call_impl
    return self._call_impl(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1747, in _call_impl
    return forward_call(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/transformers/models/deberta_v2/mo
deling_deberta_v2.py", line 1107, in forward
    pooled_output = self.dense(context_token)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1736, in _wrapped_call_impl
    return self._call_impl(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/torch/nn/modules/module.py", line
1747, in _call_impl
    return forward_call(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File
"/usr/local/lib/python3.11/dist-packages/bitsandbytes/nn/modules.py", line
477, in forward
    out = bnb.matmul_4bit(x, self.weight.t(), bias=bias,
quant_state=self.weight.quant_state)

^^^^^^^^^^^^^^^^^^^^^^
AttributeError: 'Tensor' object has no attribute 'quant_state'
```