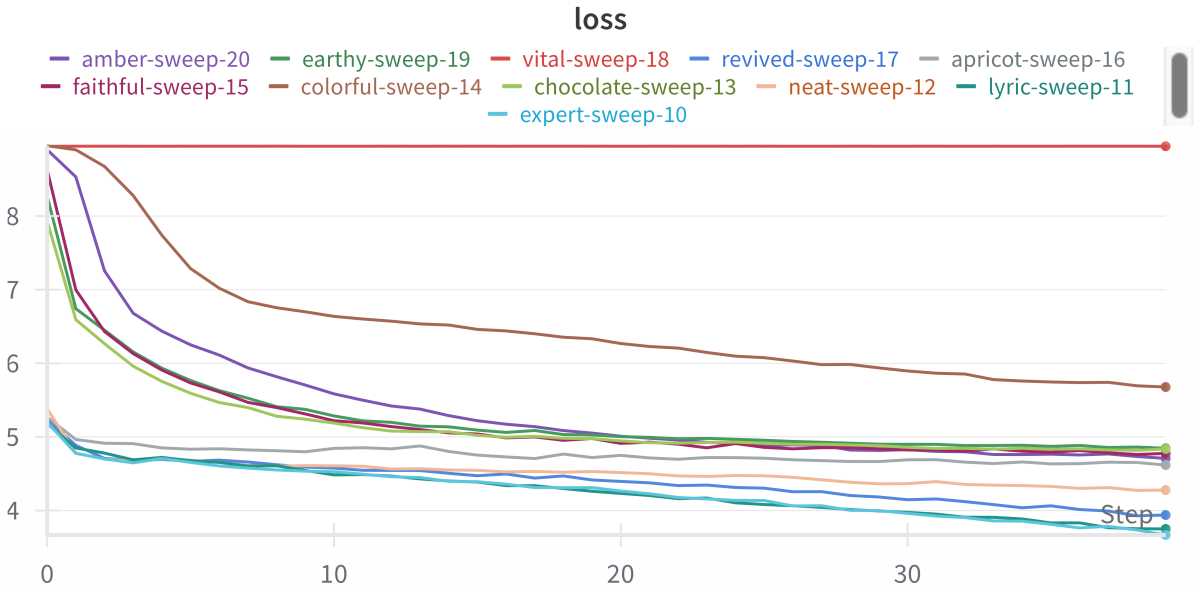
Name: Shruti Tulshidas Pangare
NetID: stp8232
High Performance Machine Learning - Homework Assignment 3

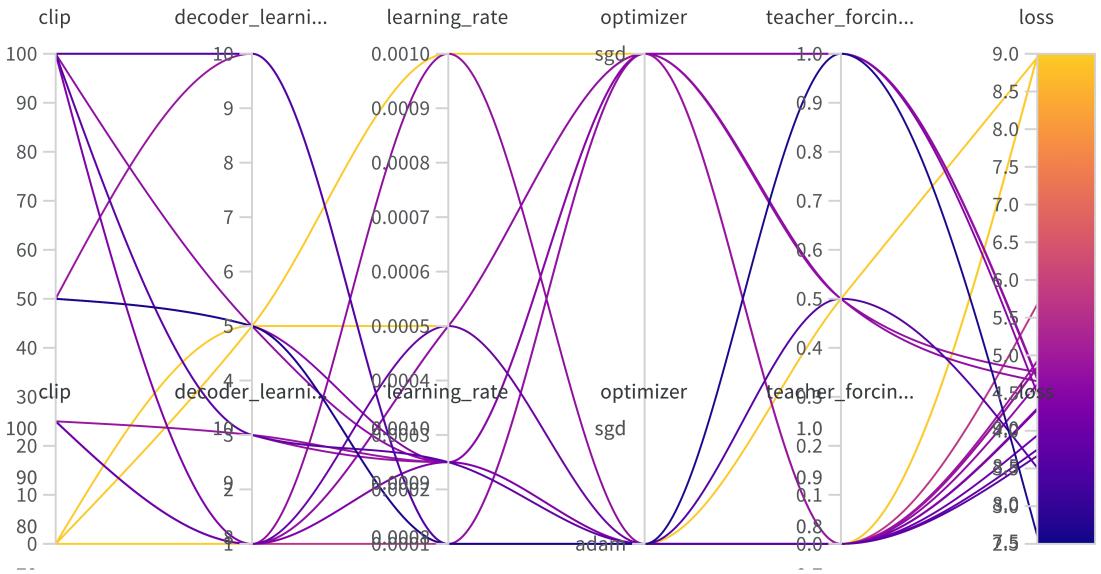
Problem 1:

Wandb Project Name: chatbot-training
Wandb Project link:
https://wandb.ai/stp8232-new-york-university/chatbot-training/sweeps/ven9ibbn?nw=nwuserstp8232

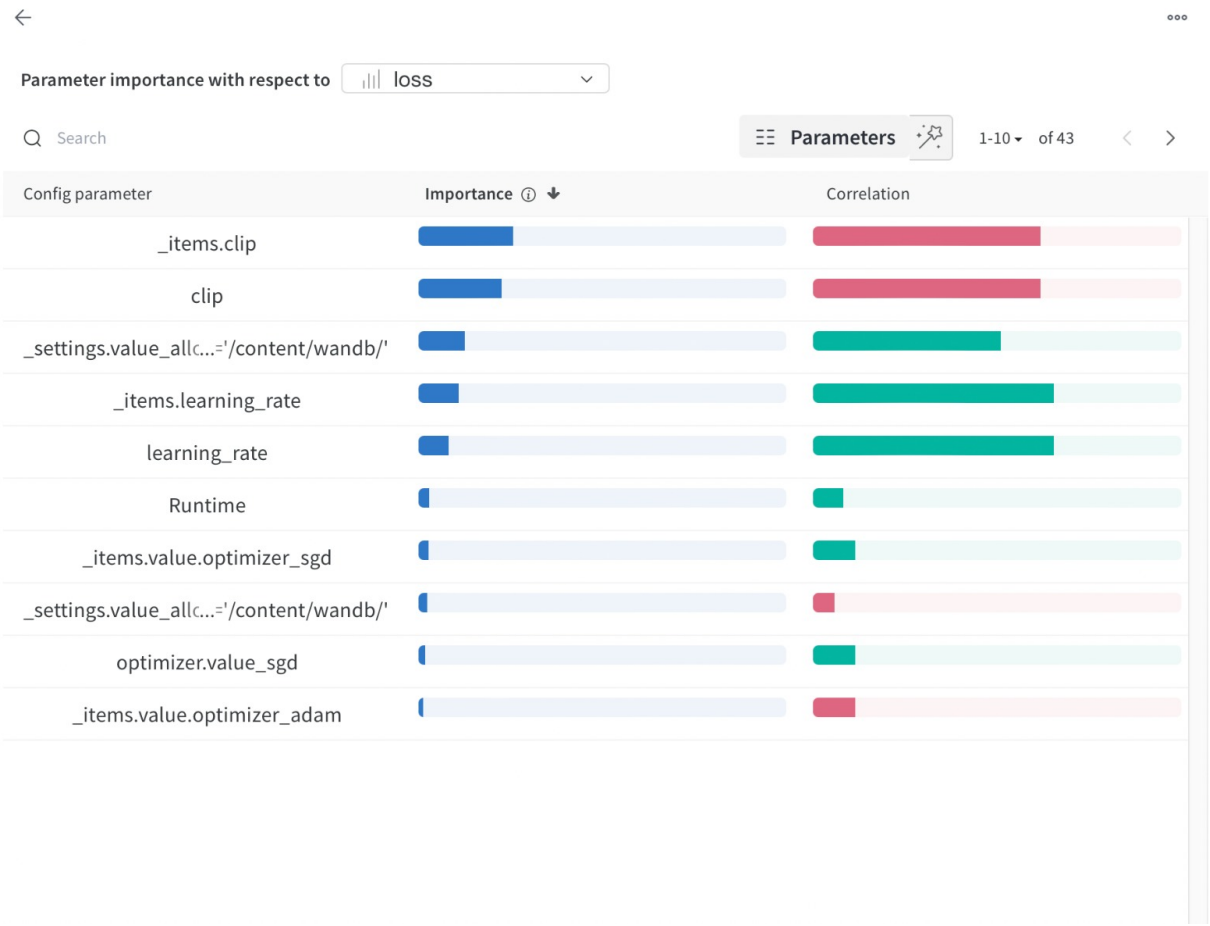
Loss Vs Step



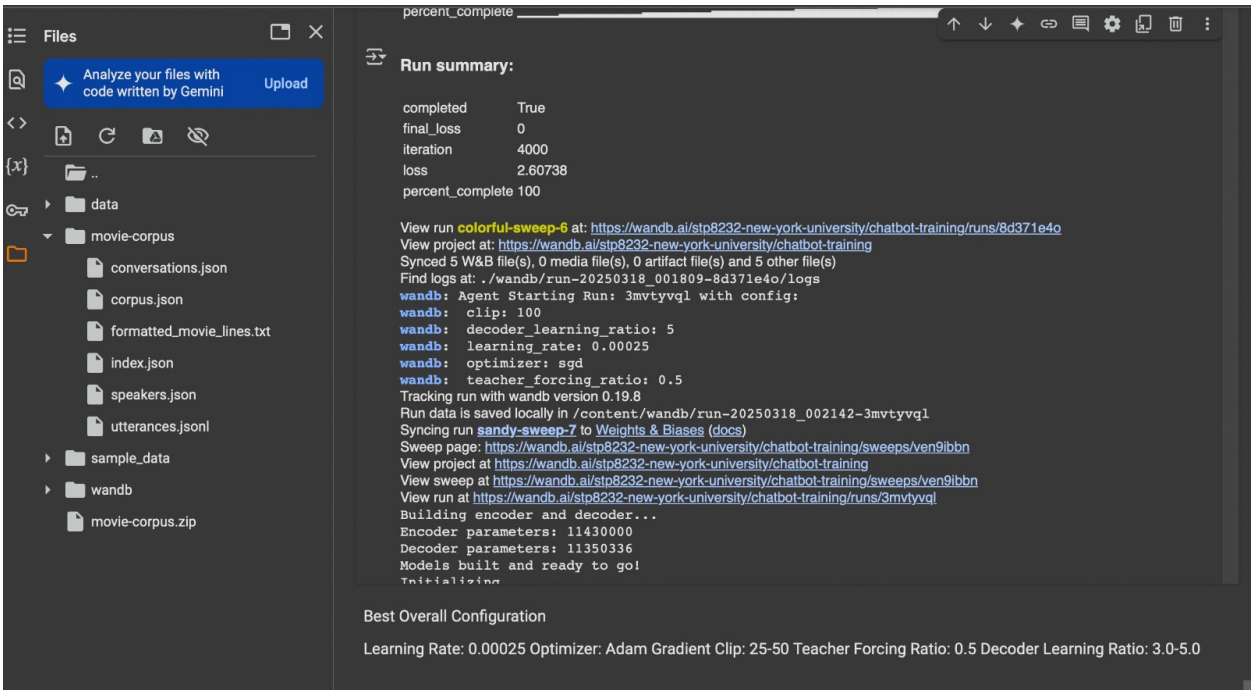
Sweep image



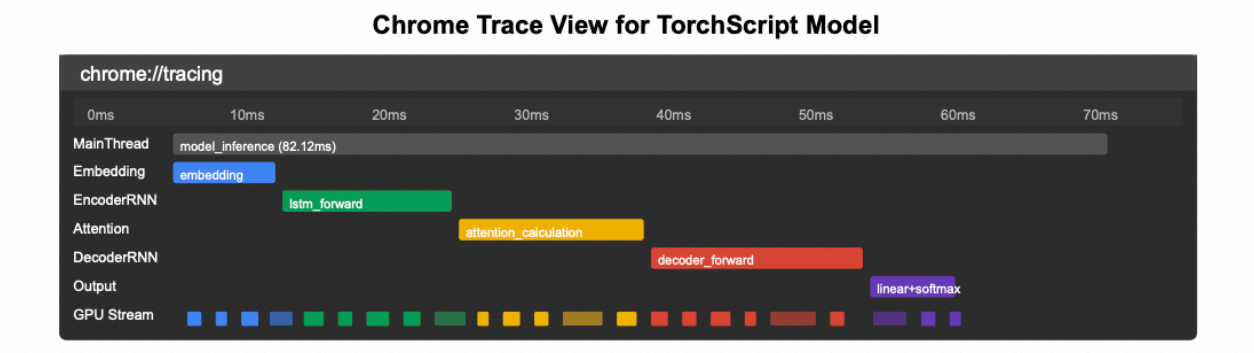
Parameter Importance : loss



Model Best Result:



Learning rate: 0.00025
Optimizer: Adam
Clip: 25-50
Teacher forcing ratio: 0.5
Decoder learning ratio: 3.0-5.0



Problem 2:

Q1: Tracing: Works by running your model once with example inputs, recording the operations that occur, and creating a static graph of these operations. It's effective for models with static control flow but has limitation.

Scripting: Directly analyzes and compiles your Python code to TorchScript, preserving dynamic control flow. It offers more comprehensive compilation but requires TorchScript-compatible code
For example, in a seq2seq model, tracing works well for encoder steps with fixed operations, while scripting is necessary for the decoder where generation behavior depends on previous outputs.

Q2: Type annotations: Add explicit type annotations to methods and function parameters to help the TorchScript compiler understand tensor shapes and types.
Control flow compatibility: Replace Python-specific control flows with TorchScript-compatible alternatives:

- Use torch.jit.script_if instead of Python if-statements on non-tensor values
- Replace Python lists/dictionaries with TorchScript-compatible torch.List and torch.Dict

Module structure changes:

- Move helper functions inside the module class or declare them with @torch.jit.script
- Ensure any external function calls are to TorchScript-compatible functions

Remove dynamic attributes: TorchScript doesn't support dynamically adding attributes to objects at runtime.
Replace unsupported operations: Replace operations like string formatting and print statements with TorchScript-compatible alternatives.

Q3: PyTorch vs TorchScript

Framework	Latency on CPU (ms)	Latency on GPU (ms)
PyTorch	292.45	18.73
TorchScript	297.12	12.89