MiniGPT Project Report

Summary of Model, Data, Training, and Findings

October 20, 2025

This report compiles the provided details on MiniGPT’s architecture, dataset, training, and observed outcomes.

Model Architecture

• MiniGPT follows a compact GPT-2 style decoder with pre-norm residual blocks and multi-head self-attention stacked twice, using embedding width 128, four attention heads, GELU-activated 4× feed-forward layers, tied positional embeddings, and a projection head over the 50 256-token GPT-2 vocabulary; details are in assignment2/mini_gpt.py:19 and assignment2/mini_gpt.py:94.

• Parameterization sums to ≈13.39 M weights: token embeddings (6.43 M) + positional embeddings (0.13 M) + two transformer blocks (0.40 M) + output head (6.43 M) + norms; the checkpoint metadata in assignment2/runs/exp01/mini_gpt_best.pt confirms embed_dim=128, num_layers=2, num_heads=4, feedforward_dim=512, and dropout=0.1.

• The model caches a causal mask and position ids as non-persistent buffers to avoid reallocations, and Xavier/normal initialisation is applied for linear/embedding layers, enabling fast warm-up and stable gradients (assignment2/mini_gpt.py:107).

• Generation utilities support temperature scaling and top-k filtering for qualitative checks (assignment2/mini_gpt.py:154), sharing the same forward pass to ensure sampling consistency with training-time behaviour.

Dataset Source

• Training corpus originates from the English slice of Hugging Face’s Wiki40B (dataset='wiki40b', dataset_name='en'), streamed to manage memory footprint while ingesting up to 1000 documents per run (assignment1/data_collection_preprocessing.ipynb:296).

• GPT-2’s 50 256-token tokenizer underpins both preprocessing (assignment1/data_collection_preprocessing.ipynb:319) and model vocabulary (assignment2/runs/exp01/mini_gpt_best.pt config), ensuring compatibility between saved tensors and MiniGPT embeddings.

• The raw notebook logs show pipeline execution on 21 Sep 2025, confirming reproducibility and provenance of the retained token batches (assignment1/data_collection_preprocessing.ipynb:349).

Dataset Processing

• Text cleaning removes HTML, links, punctuation, and collapses whitespace after Unicode-normalising to ASCII, producing lowercase plain text suitable for autoregressive modelling (assignment1/data_collection_preprocessing.ipynb:120).

• Deduplication and minimum-length filtering (≥50 words) reduce repetition and extremely short documents before tokenisation, balancing coverage with training efficiency (assignment1/data_collection_preprocessing.ipynb:139).

• Documents are tokenised with GPT-2 and chunked into contiguous 1 024-token blocks without special tokens, matching the model's max_seq_len to avoid truncation at training time (assignment1/data_collection_preprocessing.ipynb:169).

• A lightweight PyTorch-style dataset saves batched tensors (inputs and attention masks) to assignment1/sample_dataset.pt; given the checkpoint's 26 990 update steps over five epochs and default batch size 16, each epoch touches ≈86 k training sequences, implying the saved corpus spans ≈0.09 B tokens when accounting for both inputs and masks (assignment2/train_mini_gpt.py:137, checkpoint metadata).

• Padding and mask collation mirror the training loader, so loader outputs already align with MiniGPT's expectations (assignment1/data_collection_preprocessing.ipynb:208).

Training & Experiments

• Training runs via assignment2/train_mini_gpt.py, which enables Flash/Efficient attention when available, mixed precision, TF32 on CUDA, gradient clipping ($\|g\|_2 \le 1$), and periodic checkpointing/logging (assignment2/train_mini_gpt.py:81, assignment2/train_mini_gpt.py:137, assignment2/train_mini_gpt.py:236).

• The recorded experiment (runs/exp01) used GPU (cuda:0 tensors in mini_gpt_best.pt), AdamW with $\text{lr} = 5 \times 10^{-4}$, $\text{betas} = (0.9, 0.999)$, $\varepsilon = 1\text{e-}8$, weight decay 1e-4, and no AMSGrad or foreach kernels, as seen in the optimizer state stored in the checkpoint.

• Five epochs (26 990 gradient steps ≈ 5 398/epoch) were completed with validation split 10%; train and validation loaders share batch size and padding policy, keeping evaluation unbiased (assignment2/train_mini_gpt.py:149).

• Key metrics across epochs are summarised below.

• Checkpoint metadata captures best_val_loss=4.5130 at epoch 5, providing a clean resume point for further sweeps (learning rate, depth, batch size) without rerunning initial epochs.

Observations & Challenges

• Training curves show steady convergence with minimal train/validation gap, implying regularisation (dropout 0.1, weight decay 1e-4) is adequate for the current corpus size; nevertheless, perplexity ≈91 indicates the model underfits compared to full-scale GPT baselines, largely due to limited data and small model capacity.

• Despite mixed precision and efficient attention, throughput is bounded by single-GPU memory when handling 1 024-token blocks; longer contexts or deeper models would require gradient checkpointing or sequence truncation (assignment2/train_mini_gpt.py:133).

• Data quality remains the dominant bottleneck: Wiki40B is clean but relatively uniform; expanding document diversity or increasing max_docs beyond 1 000 would likely yield richer token statistics, lowering perplexity.

• Autoregressive loss computation masks padding tokens correctly (assignment2/train_mini_gpt.py:162), yet any upstream drift in attention masks (e.g., non-binary values) would silently propagate; adding dataset validation hooks before training could catch such issues early.

• Future iterations should prioritise (1) scaling depth/width alongside gradient accumulation to test capacity limits, (2) revisiting learning-rate schedules (cosine or warmup) to accelerate early convergence, and (3) broadening the data sweep to confirm the current trends generalise—checkpoint mini_gpt_best.pt is ready for transfer learning or resumed training toward those goals.

Key Metrics Across Epochs

epoch	train_loss	val_loss	train_ppl	val_ppl
1	5.2365	5.2774	188.01	195.86
2	4.9134	4.9805	136.1	145.55
3	4.7364	4.8184	114.02	123.77
4	4.5364	4.6167	93.36	101.16
5	4.4317	4.513	84.08	91.2