Three-Digit Addition

The task and measured result

The model receives two numbers from 000 to 999. It generates the ones, tens and hundreds calculations, including incoming and outgoing carry, then writes the sum with four digits.

With 7,000 random training examples, a **24,870-parameter MLP** after seven epochs and a **7,144-parameter Transformer** after fourteen epochs both produced **3,472/3,472** exact replies on the shared random final benchmark. They were the smallest successful configurations within this study's fourteen-epoch reporting budget. Their second seeds did not repeat perfect success.

The four primary and confirmation recipes have been repeated with the current Lab training and generation code. Their training curves and endpoint scores reproduced the corresponding results below. The download includes their initial and selected models and complete saved replies. The final questions had already been evaluated in the original study, so this repetition checks reproduction of that result. The larger model and corpus comparisons retain the original study measurements.

Read the carry trace

This is an actual saved MLP reply. In the ones column, 5 + 6 + 0 gives digit 1 and carry 1. The tens column uses that carry. The final carry is zero, so the four-digit sum is 0711.

```
USER Add 325 and 386.
MODEL
5 + 6 + ↑0 -> 1 ↑1
2 + 8 + ↑1 -> 1 ↑1
3 + 3 + ↑1 -> 7 ↑0
result: 0711
[EOT]
```

Exactness requires all three intermediate lines, the four-digit answer and correct EOT. The model uses its own previously generated tokens. The reference answer is used only for scoring. The website reply browser preserves both readable and control-token views.

More training examples

These three runs used the same 105,584-parameter MLP and the same check and final questions. Seven epochs on a larger training set also means more SGD updates. All sets contain the 500 local digit and incoming-carry facts; the least frequent fact occurs 1, 8 and 18 times respectively.

TRAIN	Updates	TRAIN exact	Training checks	Shared final benchmark
1000	280000	941/1,000	804/1,168	2,337/3,472
3740	1047200	3,740/3,740	1,168/1,168	3,470/3,472
7000	1960000	7,000/7,000	1,168/1,168	3,472/3,472

Model capacity and repeated seeds

Model	Parameters	Epochs	First perfect checks	Shared final benchmark
MLP	24870	7	4	3,472/3,472
MLP	50090	7	3	3,471/3,472
MLP	105584	7	3	3,472/3,472
Transformer	7144	14	10	3,472/3,472
Transformer	13072	14	10	3,472/3,472
Transformer	23664	7	3	3,472/3,472
Transformer	49592	7	4	3,472/3,472

A qualifying endpoint had exact replies on every training example and on both of the last two check and known-question checkpoints. The benchmark is a separate measurement. At approximately 25,000 parameters, D4/H70 reached 3,472/3,472 while D16/H19 reached 650/3,472 with the same seven-epoch budget. The best of five approximately 10,000-parameter MLPs reached 750/3,472 within fourteen epochs. The 13,072-parameter Transformer rose from 3/3,472 after seven epochs to 3,472/3,472 after fourteen.

Model	Seed	Epochs	TRAIN exact	Training checks	Shared final benchmark
MLP 24,870	12345	7	7,000/7,000	1,168/1,168	3,472/3,472
MLP 24,870	23456	14	7,000/7,000	1,167/1,168	3,465/3,472
Transformer 7,144	12345	14	7,000/7,000	1,168/1,168	3,472/3,472
Transformer 7,144	23456	14	16/7,000	4/1,168	10/3,472

The second seeds changed both initial weights and shuffled order. The small Transformer was highly sensitive to this change. A successful run establishes that the configuration can learn under the recorded conditions; it does not establish reliable success for every seed or a universal minimum model size. The separate MLP final set, `test_capacity.txt`, scored 3,472/3,472 for the primary model and 3,465/3,472 for its confirmation. The table uses `test_random.txt`; the two final sets are scored separately.

Targeted carry material

The targeted and matched-control corpora formed a separate comparison on a balanced/challenge final suite. Keep it separate from the same-generator random benchmark above.

Material	Parameters	TRAIN exact	Training checks	Final evaluation
Targeted	105584	3,740/3,740	1,168/1,168	3,472/3,472
Targeted	248831	3,740/3,740	1,168/1,168	3,472/3,472
Matched control	105584	3,740/3,740	1,168/1,168	3,471/3,472
Matched control	248831	3,740/3,740	1,166/1,168	3,467/3,472

Files and current settings

Each project has one `experiment.json`, a `training/` folder and a `questions/` folder. Final questions are under `questions/final/`. The released snapshots belong in `models/initial.osai` and `models/selected.osai`. A new execution creates its own `runs/run-000001/`, with settings, frozen inputs, checkpoints and complete Prompt/Gold/Raw answers. The download contains those inputs and both snapshots for each of the four projects. Supplied replies are grouped by update under `answers/`, including `training_all.txt` at the endpoint. The separate final sets retain their own filenames under `answers/final/`. Every reply file contains the complete Prompt/Gold/Raw records.

Use a writable folder such as Documents/OshonSoft AI Language Model Lab. The application stays in Program Files. Training results are written to the project folder and do not require administrator rights. Larger research projects are separate downloads; they are not part of the ordinary-example installer.

Setting	MLP	Transformer
V / C	32 / 80	32 / 80
D / H or F	4 / 70	14 / 56
Heads / blocks	0 / 0	2 / 2
Parameters	24870	7144
Initialization	Wider embeddings +/-1.0	Standard Transformer
Primary seed	12345	12345
Confirmation seed	23456	23456
SGD	0.01	0.003
Primary epochs	7	14
Confirmation epochs	14	14
Checks	Every epoch	Every epoch
Greedy reply limit	128	128

The main folders are `three_digit_addition/mlp` and `three_digit_addition/transformer`; their `_confirmation` counterparts use seed 23456. Each includes `training/addition_train.txt`, `questions/known.txt`, `questions/dev_random.txt` and `questions/final/test_random.txt`. MLP projects also preserve `test_capacity.txt` as a separate final set. The download contains these four complete projects.

Use the numeric tokenizer with Preserve line breaks. Standalone `↑0` and `↑1` are each one token. Each complete reply has 40 targets including EOT; 7,000 training conversations give 280,000 updates per epoch. Seven epochs require 1,960,000 updates and fourteen require 3,920,000. Model dimensions and parameter counts are unchanged. Train fresh v1.0 models with this token sequence; checkpoints trained with split carry signs cannot be reused.

Repeat and inspect

On an AMD Ryzen 7 7735HS laptop, the repeated primary MLP run took about 1 h 10 min and the primary Transformer run 2 h 9 min. These times include training and all configured checks while several projects shared the CPU. Time spent paused is excluded. Other hardware and workloads will give different timings.

1. Extract the chosen project into the writable application folder. Open the installed Lab and select the **Project** page.

2. Click **Choose project…** and select the folder containing `experiment.json`. The settings fill the controls as a draft. Review them before preparing the model.

3. Click **Prepare and initialize model**. Read the actual training-target count. The prepared recipes use globally shuffled epochs and MODEL body plus EOT targets.

4. Review **Total target**, its **Epochs/Updates** unit, **Learning rate**, **Reply limit** and **Check every N epochs / update list**. Start a fresh run. Update zero and the endpoint are included in the saved checks.

5. Follow the saved checkpoints and answers with **Open run folder** and **Read replies…**. Use **Pause + save** before closing. To recover, choose the same `run.json`, click **Open / recover run**, then **Resume**. Resume keeps the original target.

6. Apply the study's checkpoint-selection rule. Load that checkpoint with **Use checkpoint…**, then choose **Final evaluation**. Keep final questions outside the ordinary checkpoint checks. Record every wrong reply as well as the correct ones.

Use Epochs, with checks every epoch: seven for the primary MLP and fourteen for its confirmation and both Transformer runs. Keep each final question set's score separate. Execution time depends on the CPU and other active work. Follow the epoch and update count on the Project page, and use Pause + save when needed.

The website presentation includes all saved comparison curves and recorded reply views through the fourteen-epoch cutoff. Use its model, question and epoch selectors to inspect the changing replies and failures. For the supplied snapshots, load models/selected.osai through Persistence, use Greedy and reply limit 128, and clear the conversation before each independent addition.