The ARC Challenge

99P Labs, Honda Research Institute

Taha Demirkan, Kavin Glidden, Sami Halabieh, Divya Sundar, Karanbir Sunner

The ARC Challenge

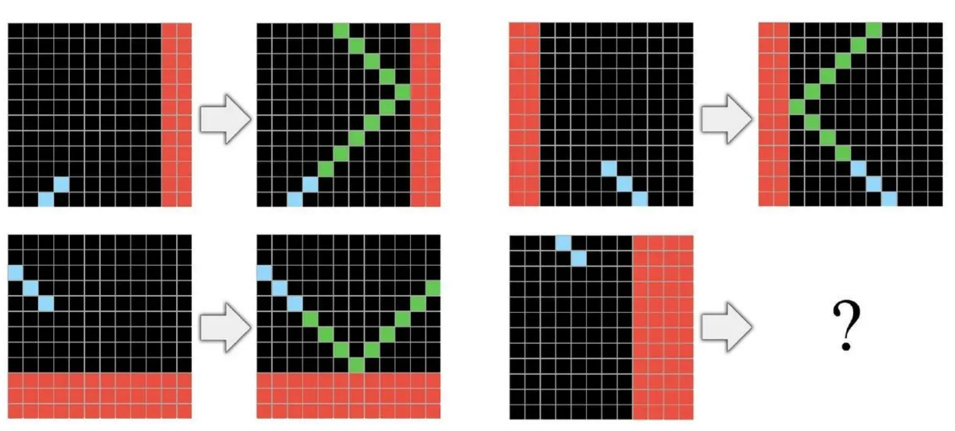
ARC (Abstraction and Reasoning Corpus) is a benchmark created to test the reasoning capabilities of the latest machine learning models of today on small grid-based puzzles.

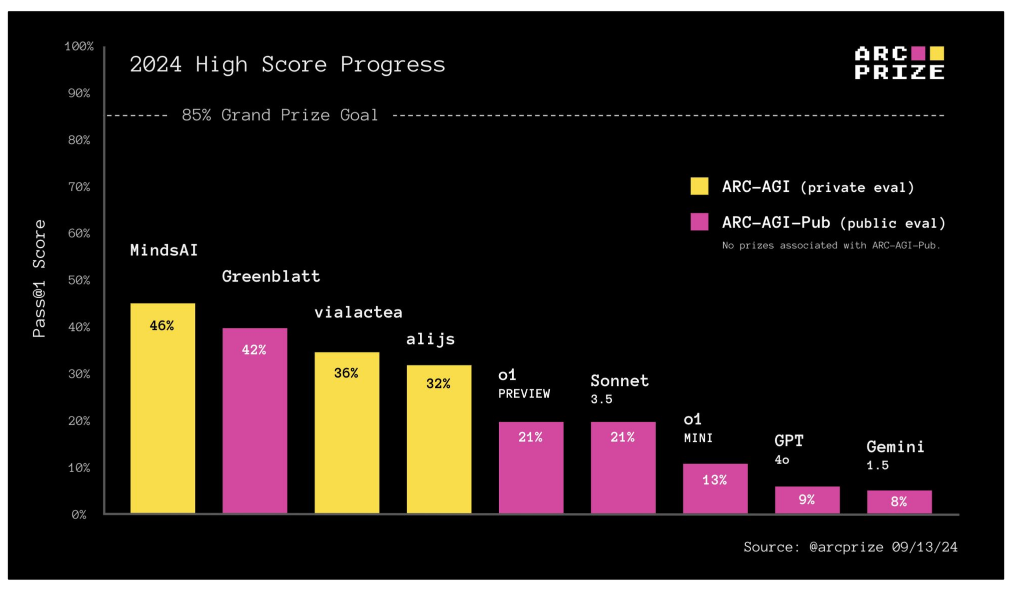
- Each task provides 3–4 example input/output pairs; the goal is to infer the transformation.
- Tasks span pixel-level, object-level, and global pattern transformations.

Some challenges that make the ARC Challenge so difficult are:

- There are only a handful of examples per task ensuring that potential solutions to the ARC-AGI challenge are not solved through sheer compute but rather innovation
- The set of tasks are extremely diverse; any successful model was not successful because they are good "memorizers" but because they "learned" during training

These two (huge!) constraints ensure that any solution must be able to infer rules, not memorize patterns (generalizability), and ensure both local and global pattern recognition (abstraction).

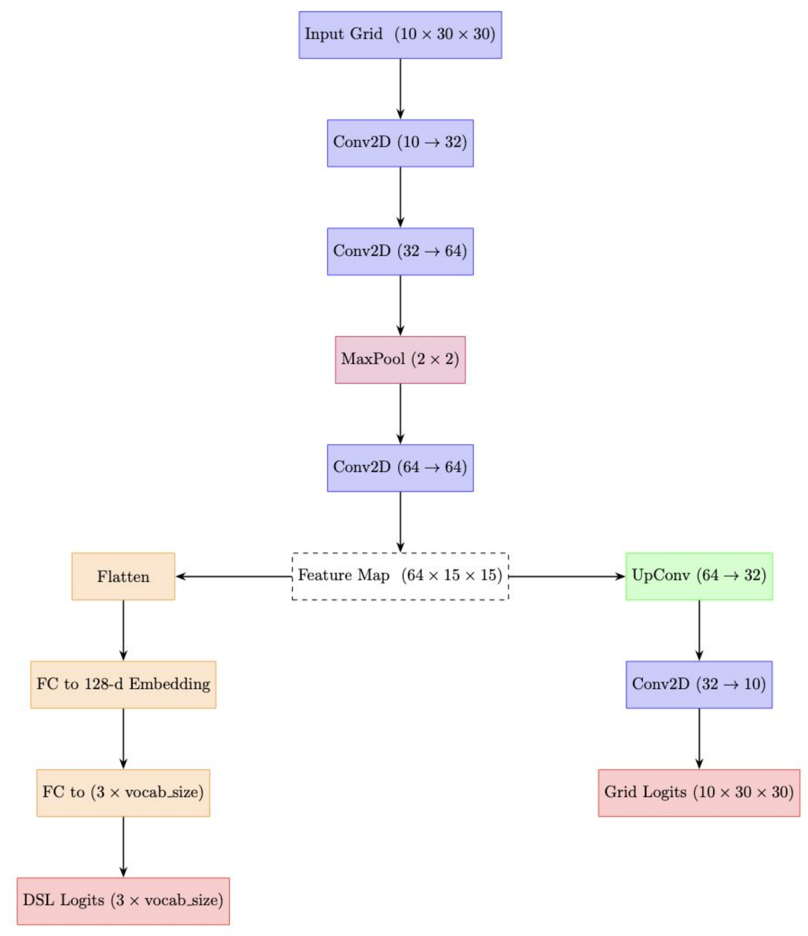




Model Architecture

We propose the **Dual-Headed Convolutional Neural Network** as a model with potential to make progress towards the ARC-AGI challenge.

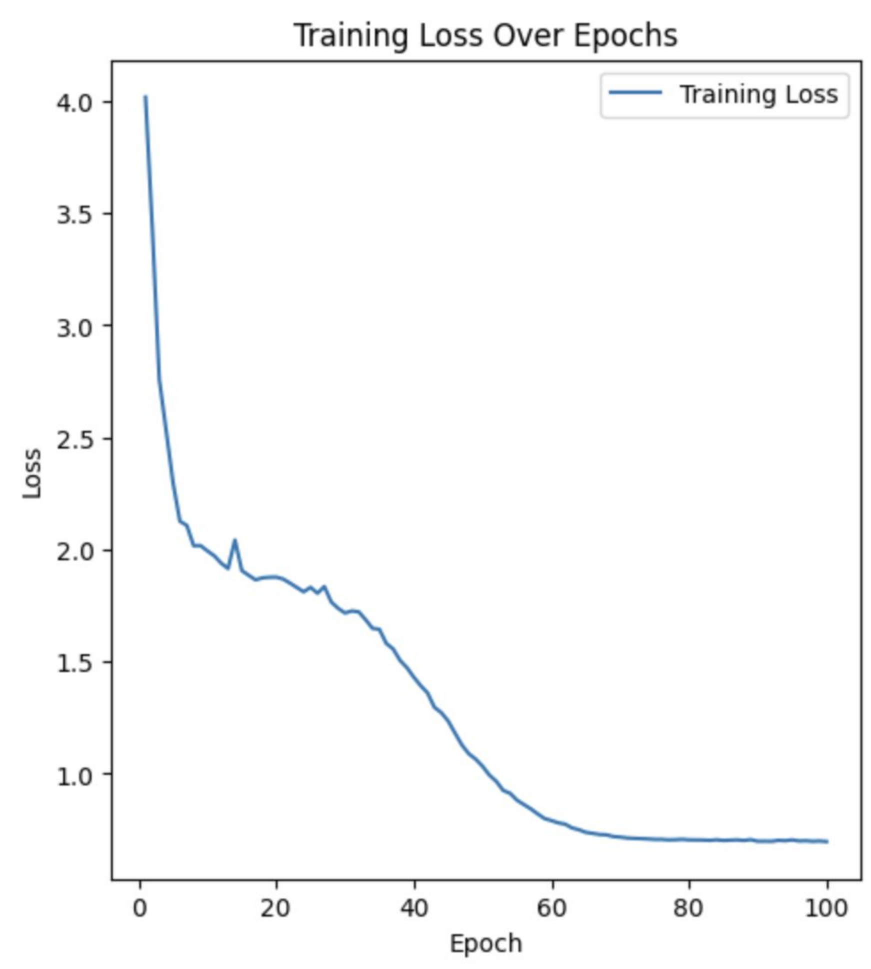
- Shared CNN backbone extracts features from input grids.
- We then have two output heads:
  - **Grid Prediction Head -** per-pixel color classification.
  - **DSL Sequence Head -** outputs a symbolic program (token sequence).
- Both heads trained simultaneously for complementary learning.



Training Pipeline

Data Preprocessing: Since the ARC tasks aren't well adapted to our dual-headed CNN model, we treated each example pair as a training sample. Then we use Hodel's DSL repo in order to retrieve the ground-truth DSL programs for the training pairs. This enables us to calculate our loss for the DSL head by comparing the sequence of transformations.

Training Loop: We trained our dual-headed CNN model for 100 epochs with a learning rate of 0.001 and a batch size of 32. For training, it is important to note that the losses are added together and backpropagation is done on the sum. The loss function of each head is provided below and the training loss per epoch is shown at the right.



Grid output head loss fn:

$$\frac{1}{B}\sum_{b=1}^{B}\frac{1}{L_b}\sum_{k=1}^{L_b}-\log q_{b,k}(t_k)$$

DSL sequence head loss fn:

$$\frac{1}{B}\sum_{b=1}^{B}\frac{1}{NM}\sum_{i,j}\ell_{\mathrm{CE}}(p_{b,i,j},y_{b,i,j})$$

Findings

The DSL sequence head achieved a **Program Exact Match Accuracy of 20.0%** and a **Program Token Accuracy of 37.0%** across the validation set. Our main suspicions that achieving the full program transformation would be hard was affirmed but the fact that we were still able to generate a somewhat correct sequence of transformations was quite surprising to us. Our results only confirmed the difficulty of the ARC-AGI challenge.

Conclusion

In conclusion, while not successful in solving the ARC-AGI challenge, the idea of a **hybrid architecture** is still an interesting concept nonetheless. In the future with more compute, we hope to experiment with models that are better able to capture the global relationships such as **transformer networks** whose attention mechanism allows for such relationship to be covered adequately. Furthermore, ARC released the ARC-AGI-2 benchmark which LLMs such as o3 medium haven't been able to get past 3.0%. Developing models for this new iteration may yield better than SOTA results.

Limitations & Areas of Exploration

Due to time and compute constraints, here are some limitations one could improve on:

- Convolutions capture only local patterns and struggle with global or relational transformations (e.g. mirroring distant regions).
- Simultaneous training of grid and DSL heads can lead to imbalance, where one head dominates shared features and starves the other of gradient signal.
- There's no mechanism for iterative or feedback-driven program refinement (e.g. checking DSL output against grid predictions).
- The approach assumes a known DSL and fixed primitive set, so it cannot learn new abstractions or extend its own language.

Acknowledgements

This project was conducted by students at the University of California, Berkeley through the CDSS Discovery Research Program in partnership with the Honda Research Institute USA.

99P LABS
Honda Research Institute (USA)

References

Chollet, François. "Arc Prize." *ARC Prize*, arcprize.org/. Accessed 5 May 2025.

Hodel, Michael. *Addressing the Abstraction and Reasoning Corpus via Procedural Example Generation*, 10 Apr. 2024, arxiv.org/html/2404.07353v1.

Berkeley
UNIVERSITY OF CALIFORNIA