Representation Learning as Cultural Competition

[draft]

François Fleuret

July 15, 2024

Abstract

We propose to investigate the development of abstract representations as a social competition.

The process starts with hand-designed sets of "world quizzes", for which algorithmic generators have been implemented. They allow us to produce as many samples as necessary. GPTs trained on these samples are then used to create new types of quizzes that we dub "culture quizzes". The generation process enssures that they are solvable but complex enough and they are added to the training set. The models are continually improved on this growing "culture".

Experimental results show that this procedure leads effectively to the creation of novel concepts in the "culture quizzes" that expand and combine the ones underlying the hand-designed "world quizzes" used to bootstrap the process.

1 Introduction

The hypothesis behind this experiment is that high-level abstract reasoning is fueled by social competition. A group of communicating agents that try to demonstrate their cognitive superiority would end up developing a rich and consistent culture of patterns and formal rules.

The overall process for the emergence of high-level reasoning would start with natural selection. Agents would be equipped with the ability to solve cognitive problems related to the physical reality of their environment. Notions such as objectness, causality, 3D geometry, numerals, etc. would emerge in their cognitive machinery.

At some point these agents would learn to communicate, initially still under natural selection as one more strategy to survive by exchanging information related to their environment.

The key point here is that if such a communication channel exists, it can be used for social competition. These agents, among other capabilities, could use communication to demonstrate their cognitive abilities, in particular by demonstrating their ability at outdoing others' cognitive abilities.

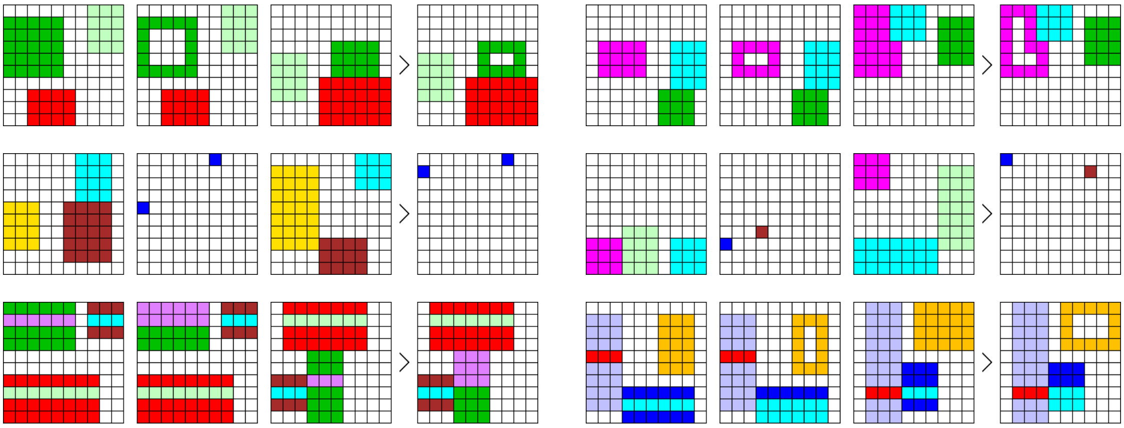

Figure 1: Examples of “world quizzes” (left) and “culture quizzes” (right) from the grid world (see § 3). Each quiz is composed of a prompt made of three grids, and a single-grid solution. Here the prompt and the solution are separated by a symbol ’>’. The “Frame” task (top left) has been generalized to non-rectangular shapes (top right). The “Detector” task (middle left) has been extended to shape-specific marker colors (middle right). The “Half-fill” task (bottom left) has been combined with the “Frame” task (bottom right).

This competition would move from the gene pool to the “meme pool”, since now agents would host new ideas in their cognitive machinery and produce new ones.

Besides advantages that makes sense in a physical world, namely that memes can evolve without needing to change the genotype, hence far more quickly, there are two remarkable properties for Artificial Intelligence:

- The improvement is pushed by a competition with adversaries that progress at the same pace. This is similar to self-play for Go or Chess.
- The selection by competition between cognitive agents decouples the generated culture from its immediate fit to the environment, allowing to explore and generate arbitrary abstract and rich cognitive tools.

2 Experimental Setup

The experiment is designed with a group of auto-regressive models that alternately learn to solve quizzes and generate new ones. We use for models a vanilla GPT with 37 million parameters (see Appendix A).

A “quiz” is a pair of two sequences of tokens: a “prompt” and a “solution”. They are designed so that the solution is deterministic given the prompt, and we differentiate:

- **World quizzes** that follow pre-defined distributions. They mimic the world’s physical and environmental patterns that an organism has to grasp to survive.

- **Culture quizzes** that are generated by the GPTs, and mimic the knowledge one has to master to perform socially, which is growing and constantly innovating.

In our experiments we designed algorithmic procedures to sample world quizzes, and generate as many of them as needed to avoid over-fitting.

2.1 Two way Prediction

An autoregressive model can solve a quizz by generating the tokens of the solution sequence, given the prompt sequence. However, as we will see, we are also interested in generating the prompt given the solution.

We allow these two modes by informing the model of the direction thanks to additional tokens. We define two tokens `[fwd]` and `[bck]` and train the model with two types of sequences.

The "forward quizzes" are the expected ones, where the prompt is given and the solution has to be generated (see Fig. 2).

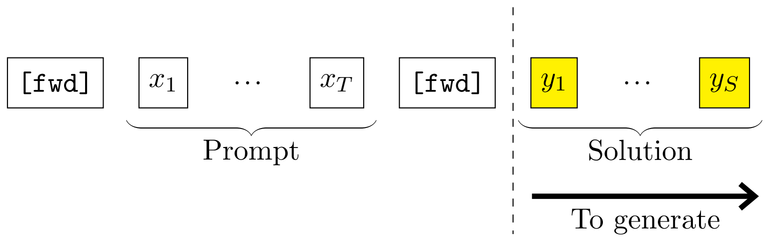

Figure 2: A forward quiz requires to generate the solution given the prompt. The `[fwd]` token is added at the beginning of the sequence and between the prompt and the solution to inform the autoregressive model that the solution is conditioned on the prompt.

The "backward quizzes" are reversed, the solution is given and the prompt has to be generated (see Fig. 3).

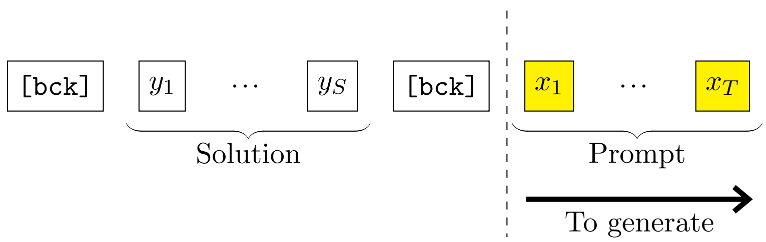

Figure 3: A backward quiz requires to generate the prompt given the solution. The token `[bck]` is added at the beginning of the sequence and between the solution and the prompt to inform the autoregressive model that the prompt is conditioned given the solution.

2.2 Performance measure

Given a forward quiz, we define that a model solves it if it generates a solution $\hat{S}$ given the prompt R that is exactly equal to the reference solution S, that is all the tokens are correct.

Defining that a model solves correctly a backward quiz is more problematic, since many different prompts correspond to the same solution S. In that case we consider the predicted prompt correct *if the model computes the initial solution when it solves the resulting forward quizz.*

2.3 Generating Culture Quizzes

When their accuracy gets above 95% we generate new quizzes as follows:

1. generate a solution (without conditioning) at temperature $T=2$, then generate a prompt for that solution at temperature $T=1/2$, and then generate a solution for that prompt at temperature $T=1/2$ (see Fig 4).
2. generate one solution for that prompt with each of the 5 GPTs at temperature $T=1$, if 4 of them are correct, validate that quiz and include it in the training data.

3 Grid Quizzes

Our main experiment relies on quizzes with a prompt of the form $A, f(A), B$ and a solution of the form $f(B)$, where the four elements $A, f(A), B, f(B)$ are elements of $\mathcal{G} = \{1, \ldots, 11\}^{10\times 10}$, and f is an unknown mapping

$$f : \mathcal{G} \to \mathcal{G}.$$

The elements of $\mathcal{G}$ can be interpreted as 10×10 grids of colored cells. In the interpretation of the patterns in those grids, the first color is white and corresponds to an empty cell. It is represented with the token `[nul]`.

The prompt and solution are serialized with the raster scan order, that is row after row, one token per grid cell. We also add tokens between grids so that the four images ends up at regular positions in the final sequences. Hence, given a prompt $A, f(A), B$, we serialize it as a sequence:

$$\begin{aligned}
&\underbrace{A[1,1], A[1,2] \cdots A[1,10]}_{A \text{ first row}}, \underbrace{A[2,1] \cdots A[2,10]}_{A \text{ second row}} \cdots A[10,10], \texttt{[nul]},\\
&\underbrace{f(A)[1,1], f(A)[1,2] \cdots f(A)[1,10]}_{f(A) \text{ first row}}, f(A)[2,1] \cdots f(A)[2,10] \cdots\\
&f(A)[10,1] \cdots, f(A)[10,10], \texttt{[nul]}, B[1,1], B[1,2] \cdots B[1,10],\\
&B[2,1] \cdots B[2,10] \cdots B[10,1], \ldots, B[10,10]
\end{aligned}$$

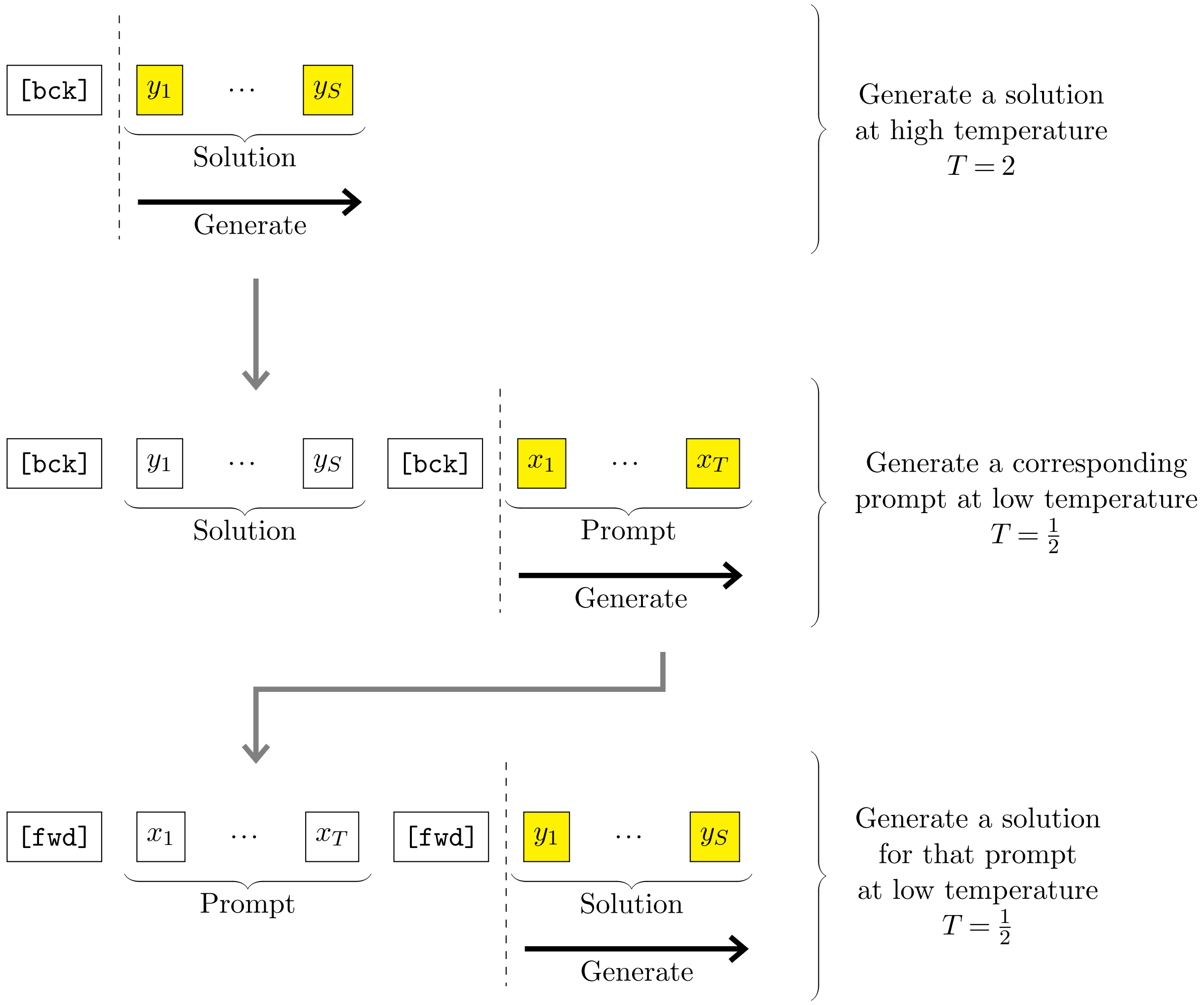

Figure 4: Given a trained model, we generate a new quiz in three steps. First a solution is sampled without conditioning on a prompt, by starting the autoregressive generation of tokens after an initial token `bck`. This is done at temperature $T = 2$ to encourage original variation. Second, a prompt is generated, conditioned on this solution, at low temperature $T = 1/2$. This prompt is then used in a forward quiz to generate the solution, again at low temperature $T = 1/2$.

For a total of 302 tokens in the prompt, and similarly 100 tokens for the solution.

3.1 World Quizzes

We have defined 12 “tasks”, each corresponding to a distribution of quizzes, that involve different high-level concepts such as color, size, contours, collision, and so on. To each of these families correspond an algorithmic implementation that allows to generate as many samples as necessary, in particular to avoid over-fitting.

The list of tasks is: “Replace color” (see Fig. 5), “Half-Fill” (see Fig. 6), “Detect” (see Fig. 7), “Frame” (see Fig. 8), “Grow” (see Fig. 9), “Translate” (see Fig. 10), “Bounce” (see Fig. 11), “Count” (see Fig. 12), “Scale” (see Fig. 13), “Trajectory” (see Fig. 14), "Symbols" (see Fig. 15), and "Isometry" (see Fig. 16).

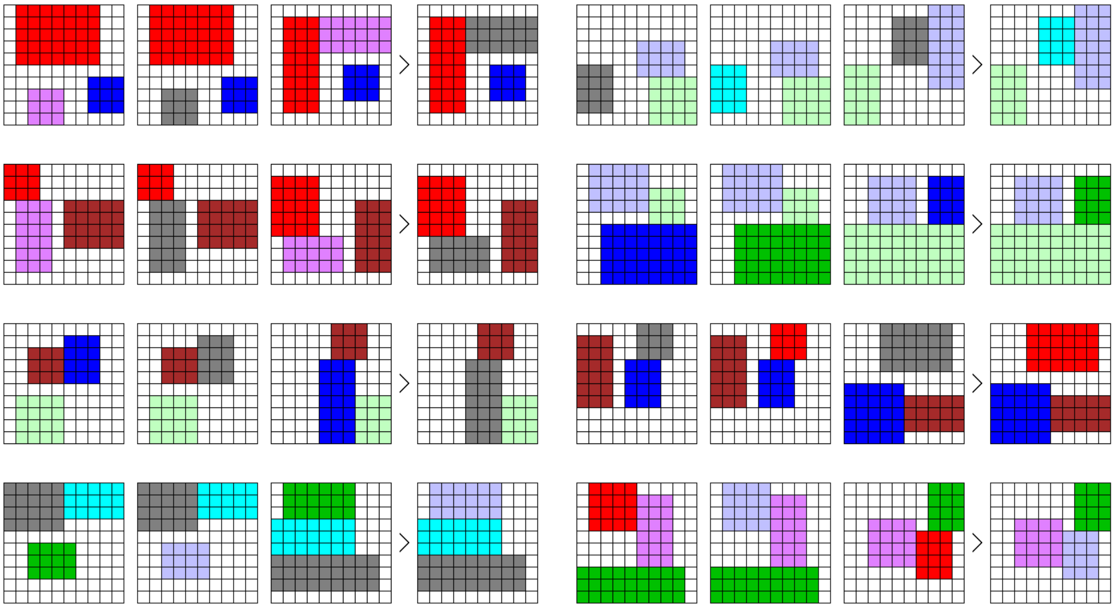

Figure 5: Examples of the "Replace color" quizzes. The first grid contains three rectangles, the second is obtained by changing one of the colors. The same must be done to the third grid to obtain the solution.

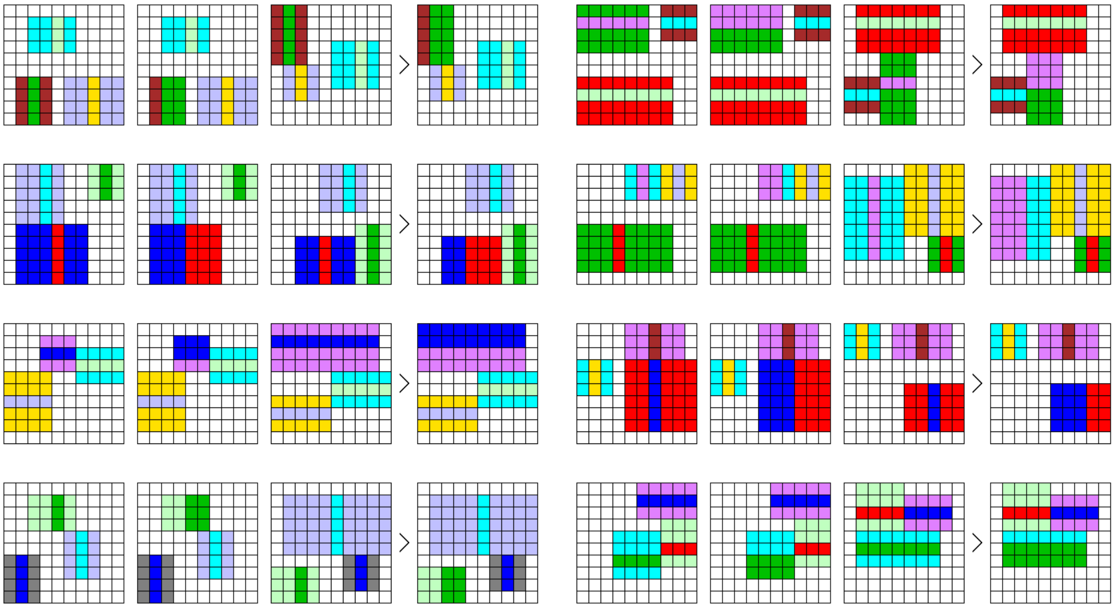

Figure 6: Examples of the “Half Fill” quizzes. The first grid contains three rectangles, each with a vertical or an horizontal line of another color in its middle. The second grid is identical with one of the rectangle having one half filled. The third grid contains three rectangles of identical colors as the first grid, of different size and locations. The solution is obtained by filling similarly one of the half of a rectangle of the third grid.

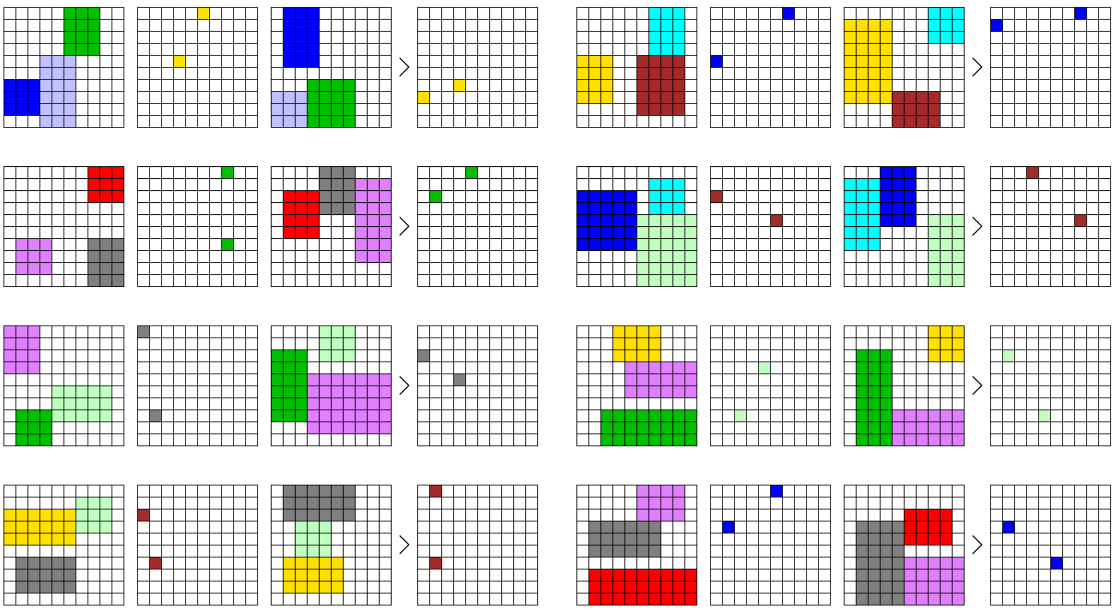

Figure 7: Examples of the “Detect” quizzes. The first grid contains three rectangles, the second has two pixels of same color located in the top-left corner of two of them. The solution is obtained by marking in the fourth grid the top-left corners of the rectangles of same colors in the third.

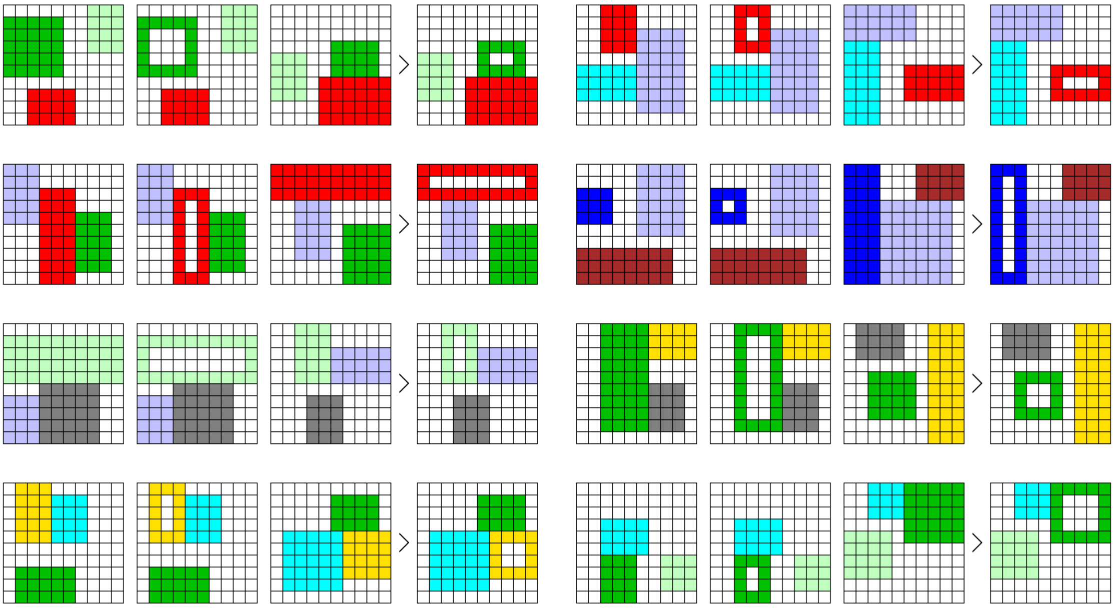

Figure 8: Examples of the “Frame” quizzes. The first grid contains three rectangles, and the second is identical except that one rectangle has been replaced by its frame. The same should be done to the similarly colored rectangles of the third grid to obtain the solution.

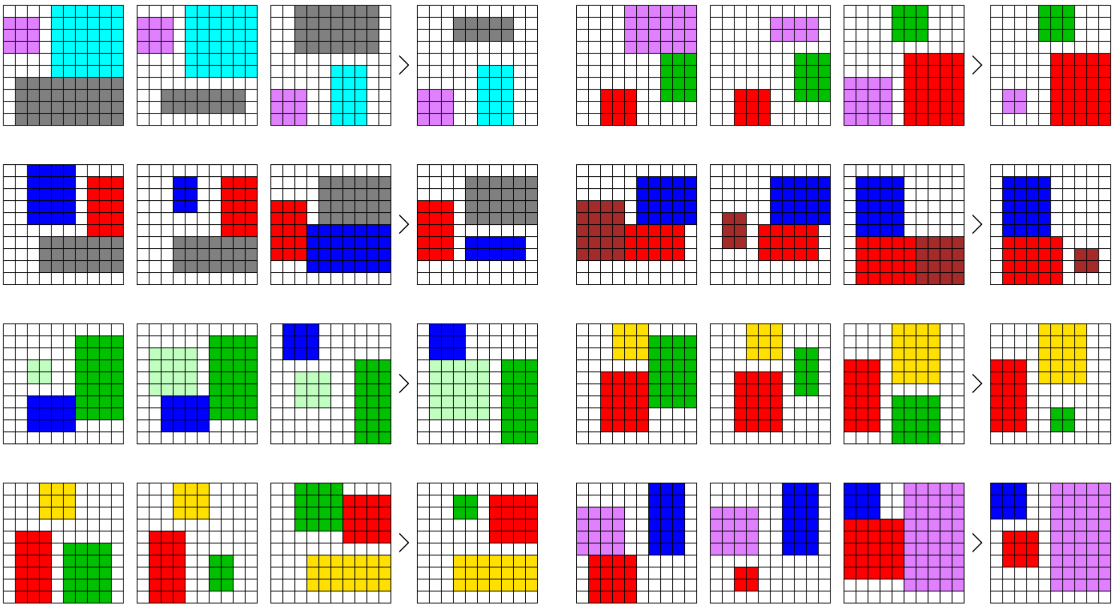

Figure 9: Examples of the “Grow” quizzes. The first grid contains three rectangles, one of them getting one pixel thicker or thinner in the second. The same should be done to the similarly colored rectangles of the third grid to get the solution.

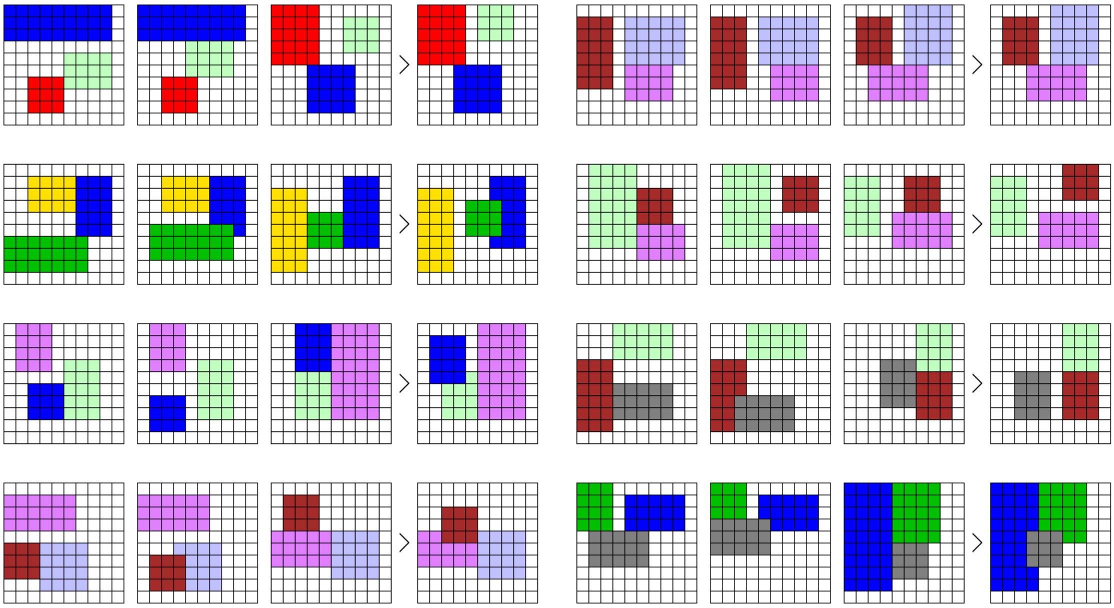

Figure 10: Examples of the "Translate" quizzes. The first grid contains three rectangles. The second is obtained by displacing one of them by one pixel in both direction. The solution is obtained by applying the same motion to the similarly colored rectangle in the third grid.

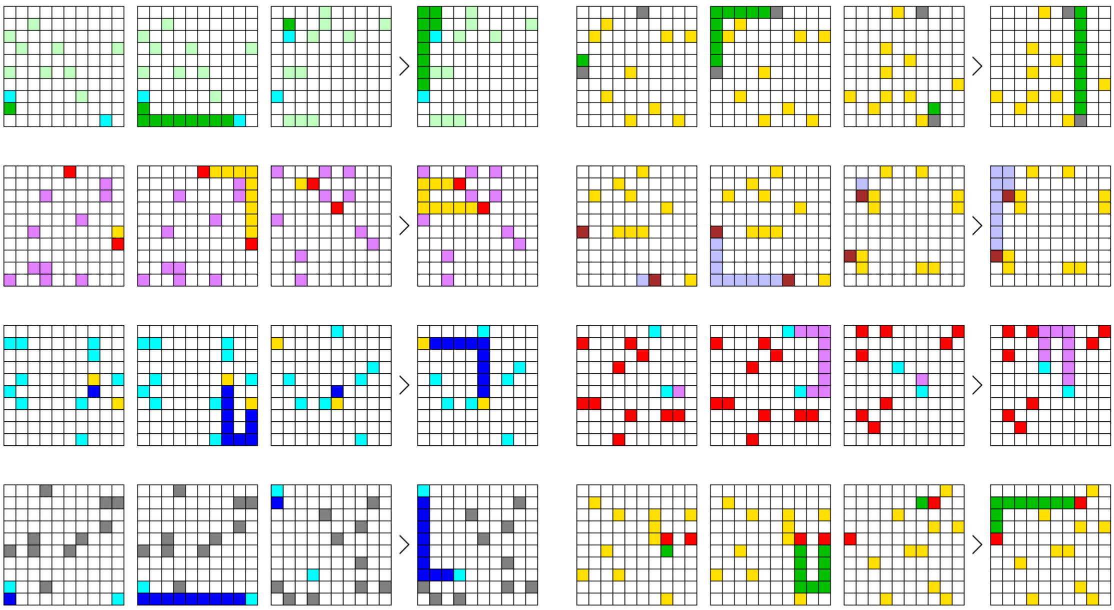

Figure 11: Examples of the "Bounce" quizzes. The solution should join the two pixels of same color, with a path of another color, starting in the direction indicated by a pixel of that color, and changing direction only when colliding with a pixel of a third color or one of the lattice border.

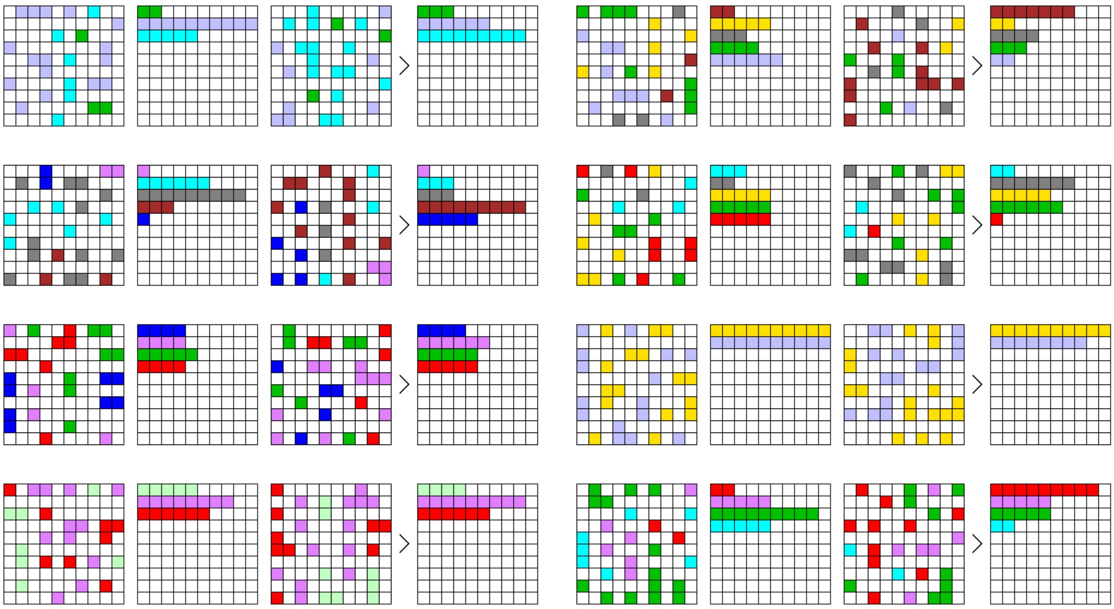

Figure 12: Examples of the "Count" quizzes. The first and third grid contains disjoint elements of a few cells and of different colors, the second grid has one row dedicated to each color appearing in the first grid. This row pictures a colored bar starting from the left, whose length is the number of items of that color appearing in the first grid. The solution follows the same rule for the third grid.

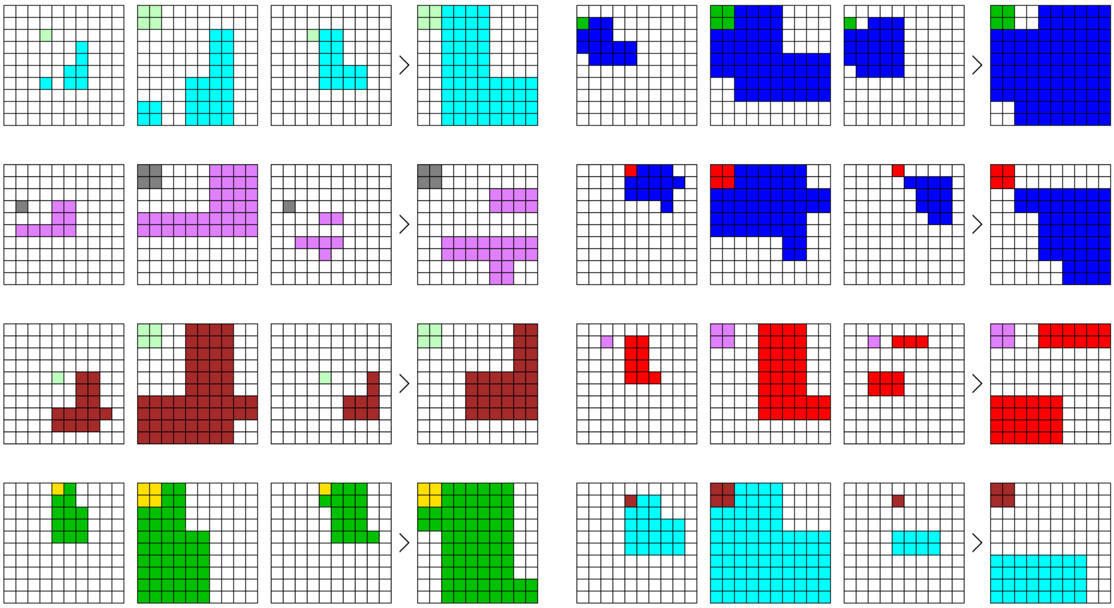

Figure 13: Examples of the “Scale” quizzes. The first grid contains a shape that fits in a 5×5 pixel box, located at a random position in the grid. It is of uniform color c_1 with one pixel of a different color c_2 in the top-left of the bounding square. The second grid is that same shape, of same color c_1, scaled by a factor of two, with a square of four pixels in the top-left corner, of color c_2. The third grid is similar to the first, except for the shape and its location, and the fourth corresponds to the shape of the third scaled up, as the shape of the first is scaled in the second.

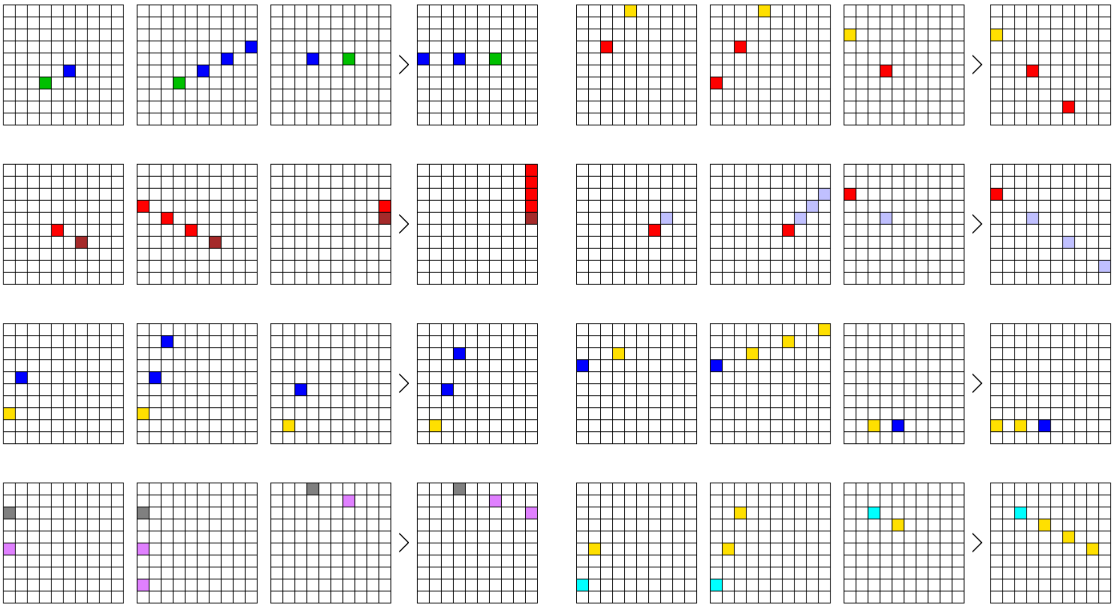

Figure 14: Examples of the “Trajectory” quizzes. The first grid contains two points of different colors, and of coordinates that differ by at most three. The second grid replicate one the two pixels, adding to the position on each one the same relative displacement there is between the two original pixels. The third grid contains two pixels of same colors as the ones in the two previous grids, and the fourth follow the same rule with respect to the third as the second with respect to the first.

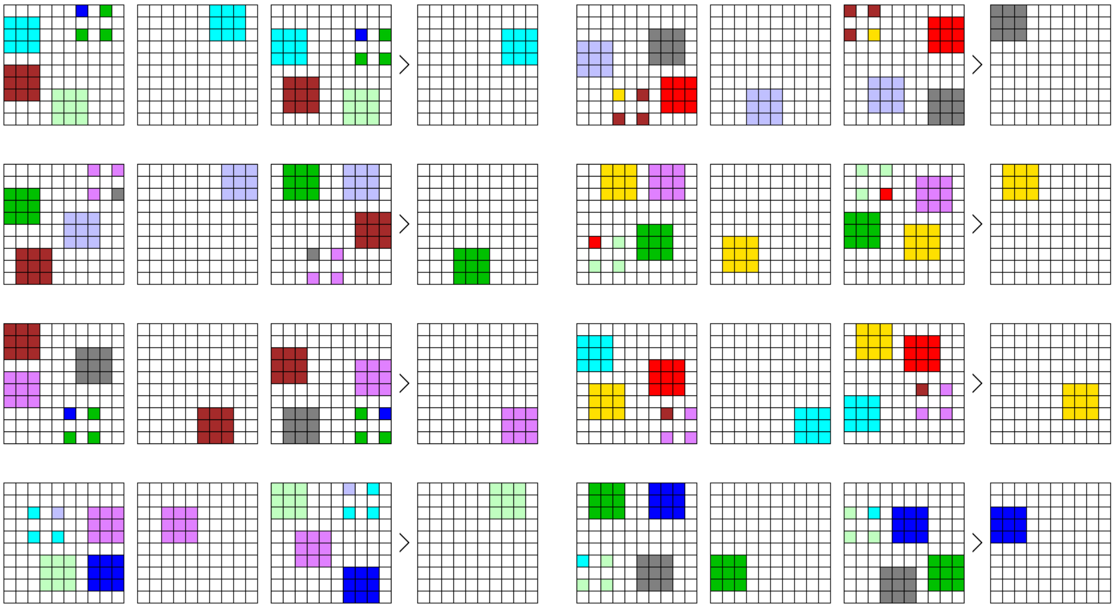

Figure 15: Examples of the “Symbols” quizzes. The firs grid contains three 3×3 squares of different colors c_1, c_2, c_3, and a group of four pixels, three of them of color c_4, the fourth of color c_5. The second grid contains a single square located where the four pixels were, and it color is that of the square in the first grid whose position corresponded to the position of the pixel of color c_5 in the group of four pixels. The third and fourth grids follow the same rule, with the same colors.

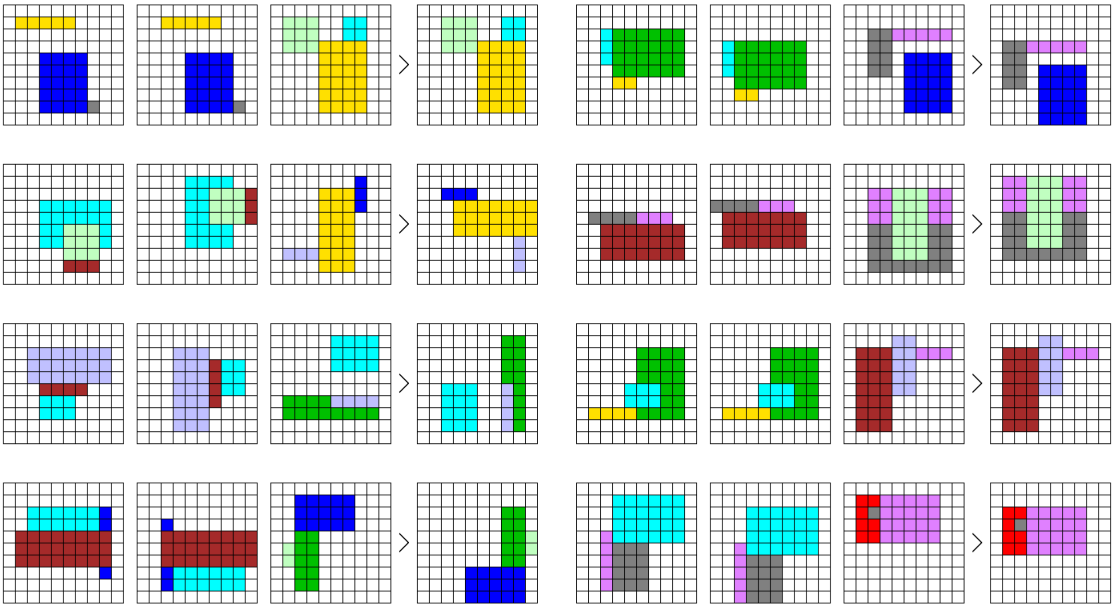

Figure 16: Examples of the “Isometry” quizzes. The first grid contains a figure composed of three possibly overlapping rectangles of different colors, and the second grid is the same figure transformed by a random composition f of rotations of $\pi/2$, symmetry and translation by one pixel in both directions. The third grid contains a different random picture generated with the same procedure as the first, and the fourth grid contains the same picture transformed by the same isometry f.

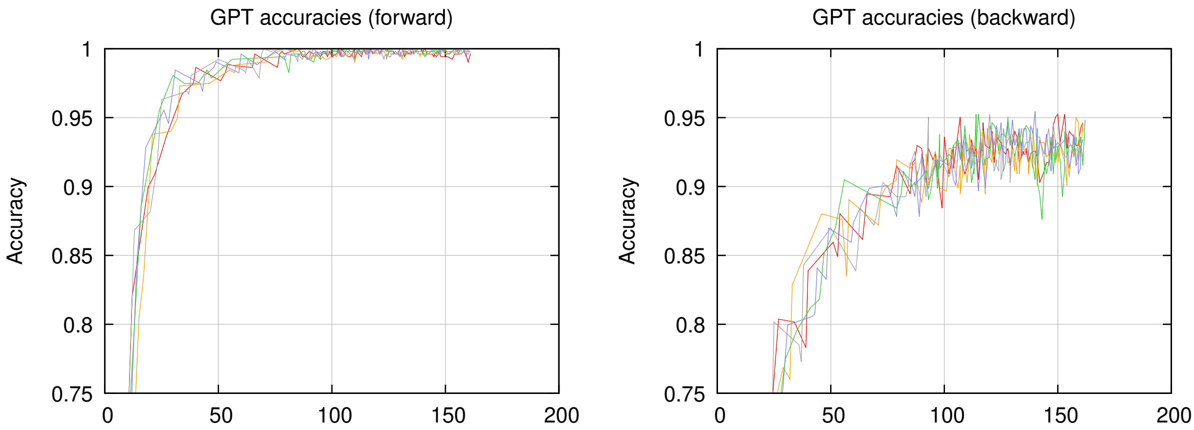

Figure 17: Accuracy for forward prediction (left) and backward prediction (right). As stated in § 2.2, the accuracy for the forward prediction is defined as the proportion of solutions which are exactly predicted, that is all the tokens are exact. Since there are many prompts leading to a given solution, the accuracy for the backward prediction is defined by re-generating the solutions given the generated prompts, and checking the proportion that match exactly the initial solution.

3.2 Prediction

The GPT performs very well in prediction. As shown on Fig. 17, the accuracy for forward prediction, that for solving the quizzes, reaches near 99%. The accuracy for backward prediction reaches values around 92%.

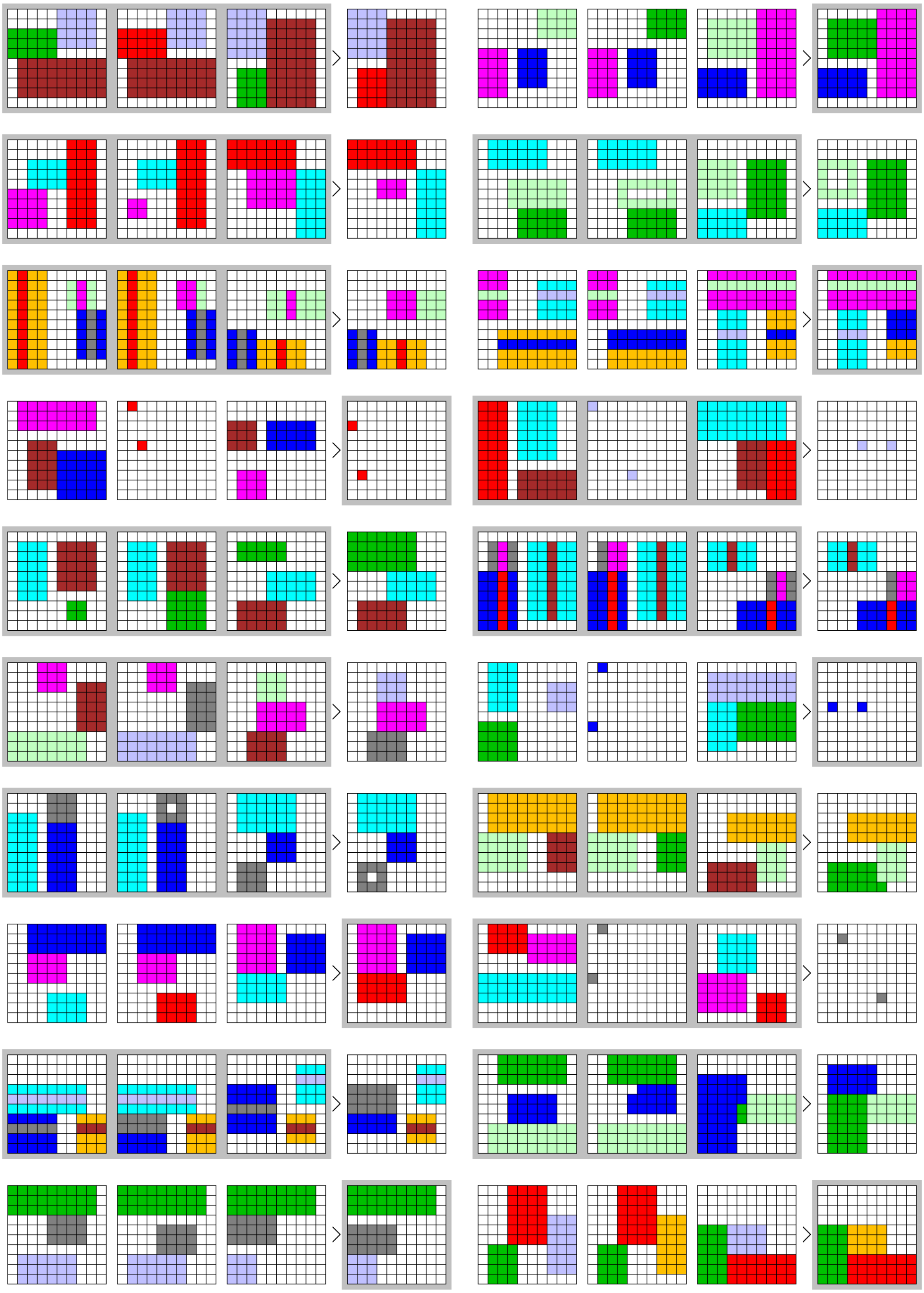

Figure 18: Predictions of one of the GPT after 30 epochs. The thick gray frames indicate the generated parts.

3.3 Generated Culture Quizzes

We list here some generated quizzes that exhibit features that were not present in the "world quizzes" used for training.

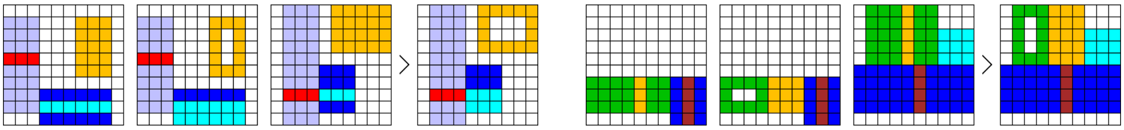

Figure 19: The quizzes "frame" and "half fill" have been combined in a single quiz.

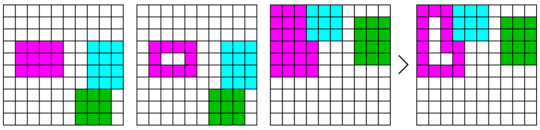

Figure 20: The "frame" quiz has been generalized to non-rectangular shapes.

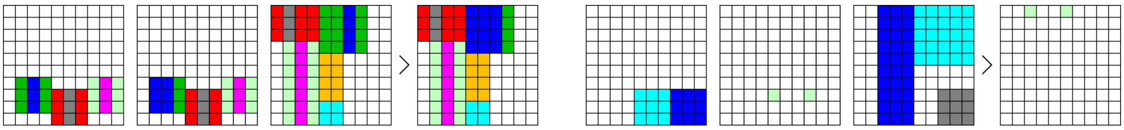

Figure 21: More rectangles were added as distractors.

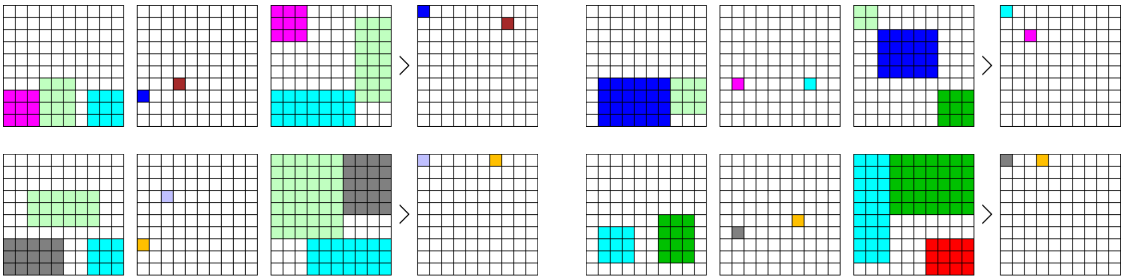

Figure 22: Variation of "Detect" with location markers colored according to the color of the rectangle they mark.

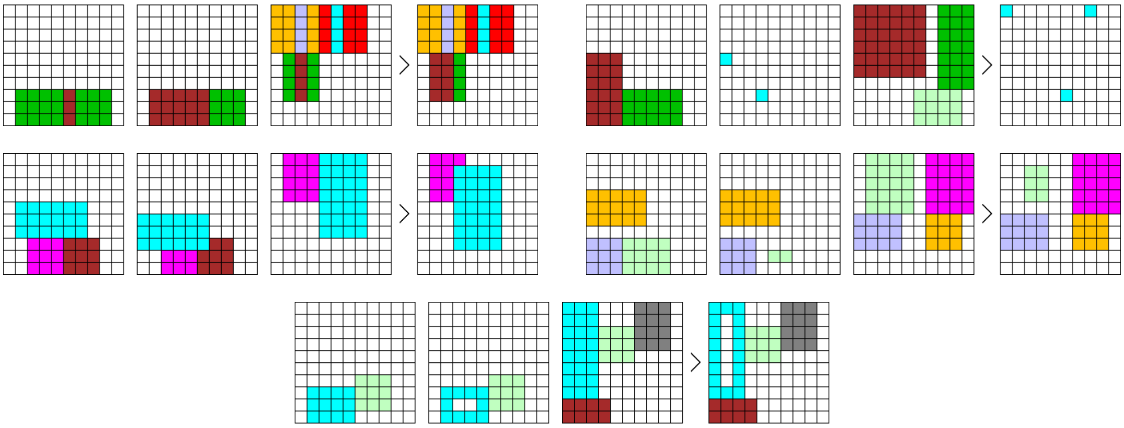

Figure 23: Variations of “Half Fill”, “Detect”, “Translate”, “Grow”, and “Frame” with a number of rectangles not equal to three.

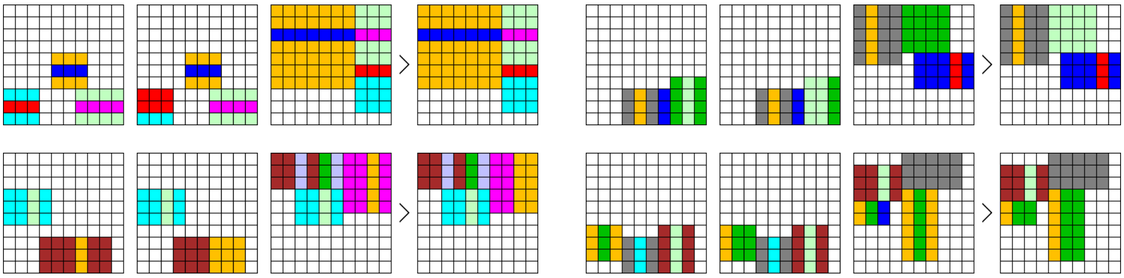

Figure 24: Variations of “Half Fill” where the shapes to change have more complex coloring.

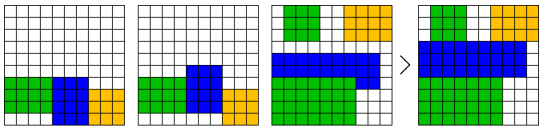

Figure 25: Variation of “Translate” where the moving part is occluded, which was never the case.

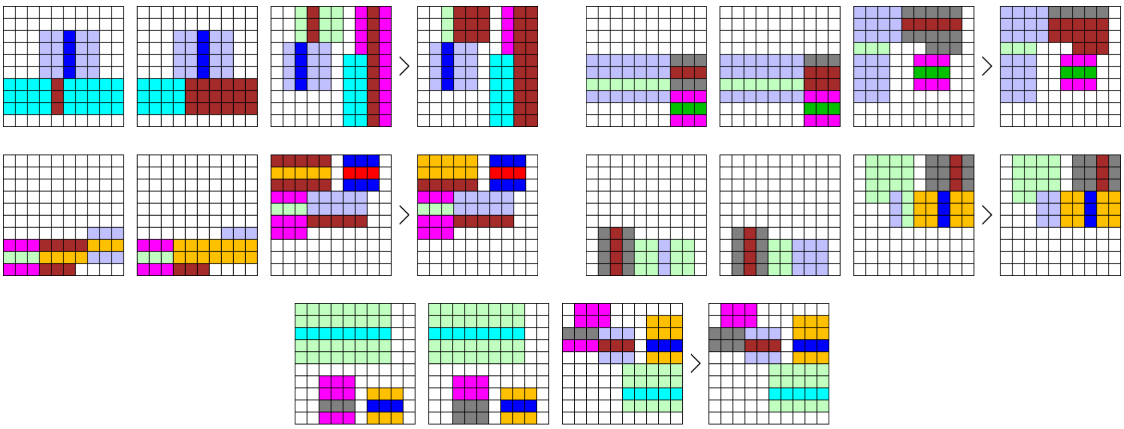

Figure 26: Variations of “Half Fill” with non-rectangular shapes.

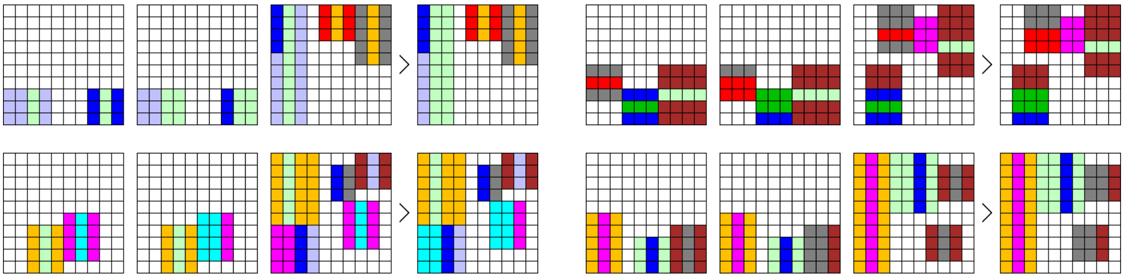

Figure 27: Variations of “Half Fill” with two colors or two rectangles have to be modified.

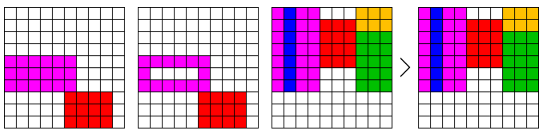

Figure 28: Variation of “Frame” with no rectangle of adequate size to be modified.

4 Discussion

The overall process we have defined is an alternating scheme that aims at expanding the distributions $P(Q, A)$ of pairs of a prompt Q and an answer A so that their "internal logic" is maintained. This is achieved by building a sequence of distributions $P_k(Q, A)$ so that conditional distributions remain unchanged, that is: $\forall k, P_k(A \mid Q) = P(A \mid Q)$ and $P_k(Q \mid A) = P(Q \mid A)$.

A Appendix

The code is available at

https://fleuret.org/git/culture

The experiments are done with a GTX 4090.

The GPT used has 37M parameters and the following structure:

`dim_model`	512
`dim_keys`	64
`dim_hidden`	2048
`nb_heads`	8
`nb_blocks`	12

Adam, $\eta = 1e-4$, no scheduling.

There are $N_{\text{train}} = 250'000$ original quizzes for training and $N_{\text{test}} = 10'000$ for test.

At each epoch, for both train and test samples, we mix original quizzes and the generated ones.

For training for instance, if there are less than $N_{\text{train}}/2$ new quizzes, we take all of them, otherwise we sample $N_{\text{train}}/2$ of them without replacement and then we sample without replacement enough original quizzes to get N_{train} samples in total.

We proceed similarly to get N_{test} samples for test.