ATTENTION MODELS, WORLD MODELS, AND GOING BEYOND WORLD UNDERSTANDING

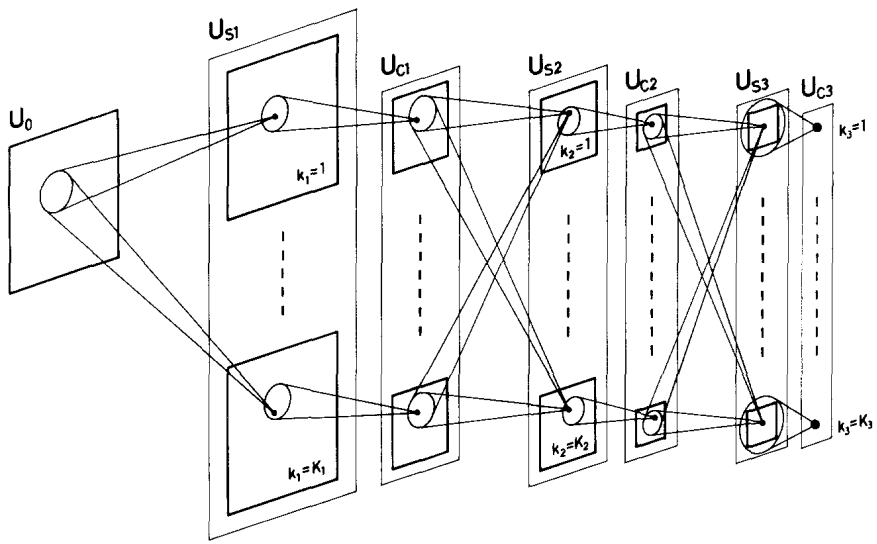
François Fleuret

UNIVERSITÉ
DE GENÈVE

Deep convolutional models

One key technology of “modern AI” are the convolutional models.

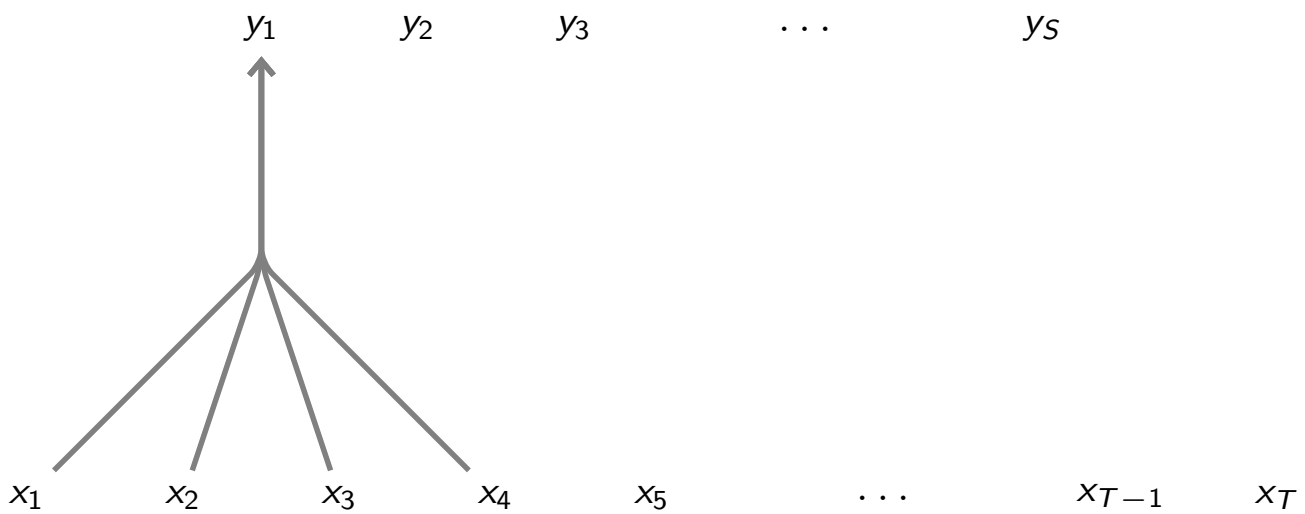
- They are powerful function approximators.
- Scale well with data set size and computation.
- Fitting for hierarchical signal structures.



Neocognitron (Fukushima, 1980)

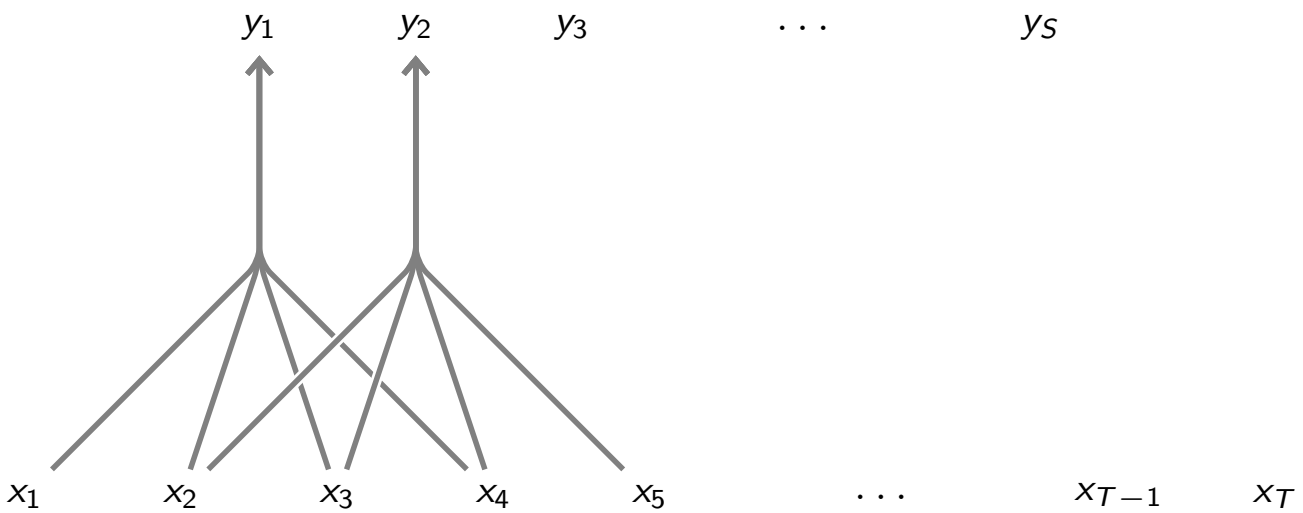
Deep convolutional models

A convolution applies the **same linear operation** at every location in the signal.



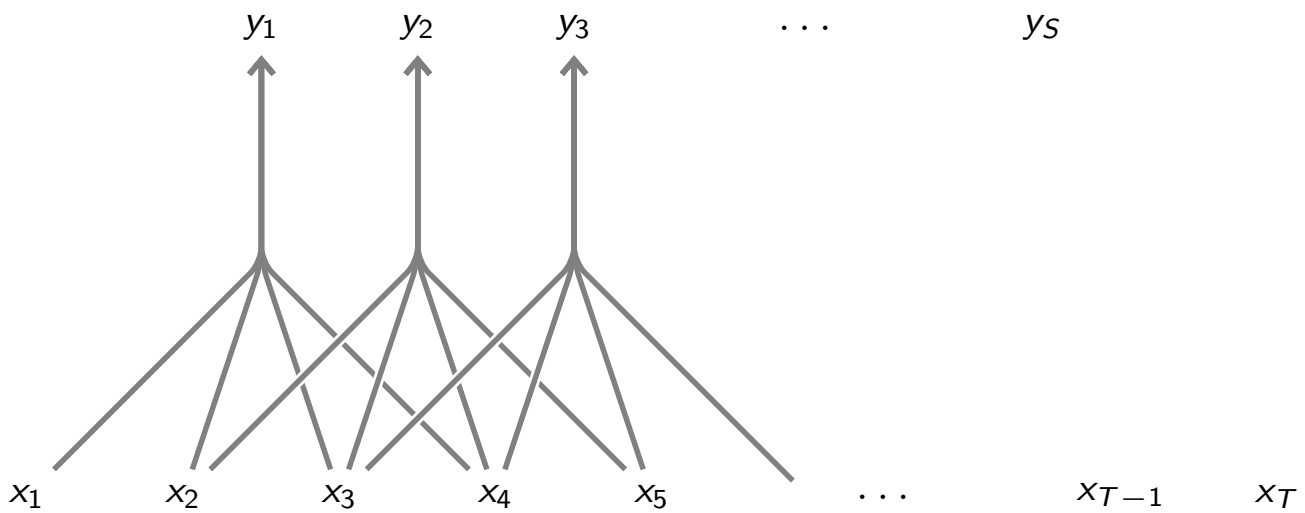
Deep convolutional models

A convolution applies the **same linear operation** at every location in the signal.



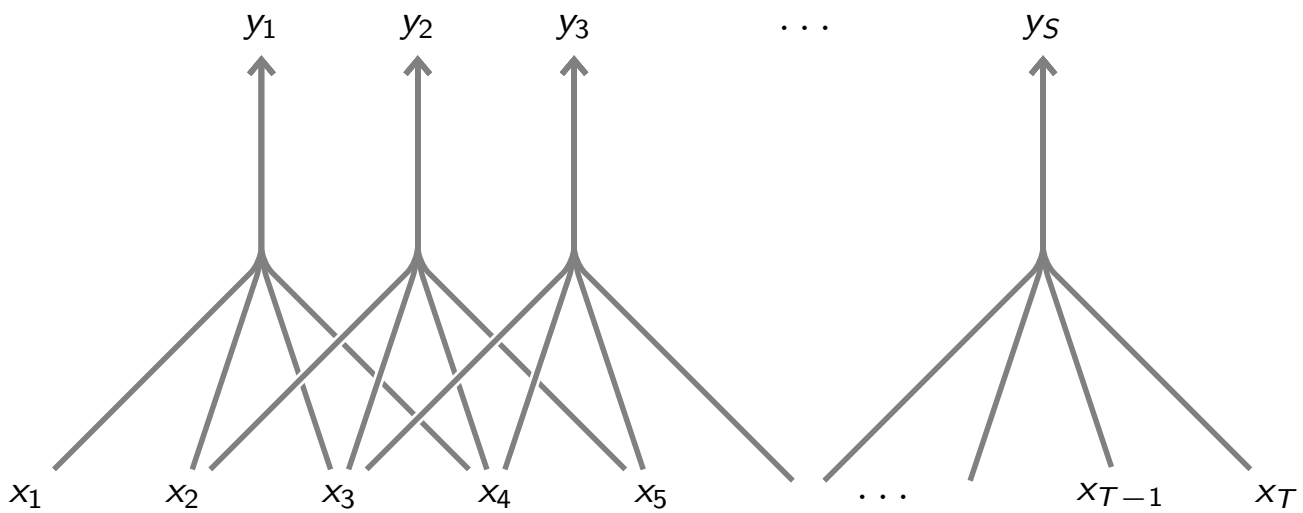
Deep convolutional models

A convolution applies the **same linear operation** at every location in the signal.



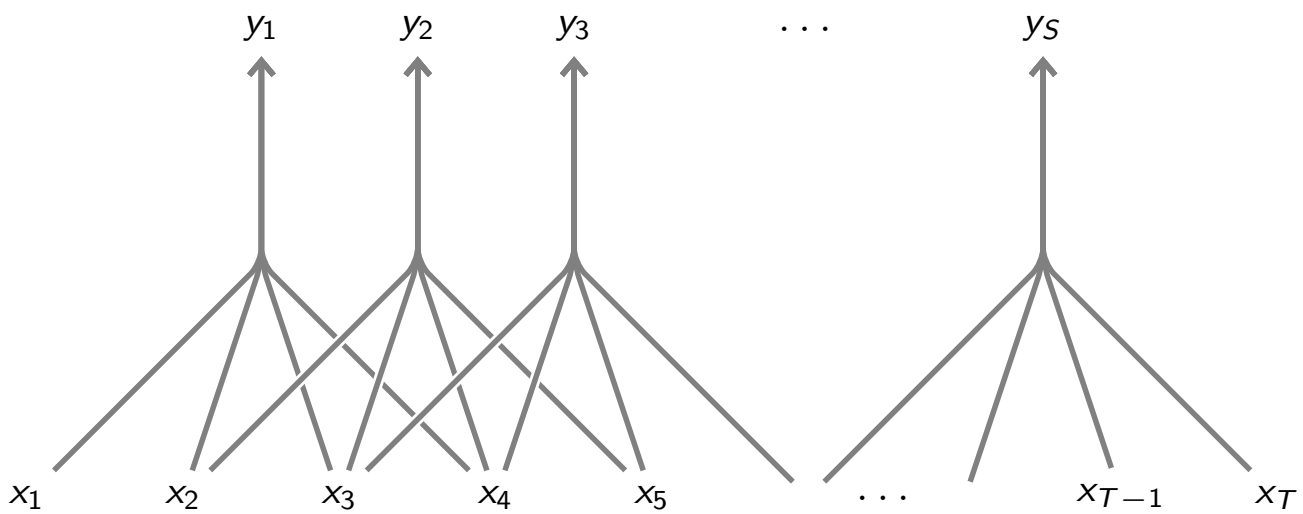
Deep convolutional models

A convolution applies the **same linear operation** at every location in the signal.



Deep convolutional models

A convolution applies the **same linear operation** at every location in the signal.



Such mechanisms are very efficient for image or sound processing where

- the signal is stationary, and
- local structures are very informative.

Attention mechanisms

However some tasks involve more than composing local structures, e.g. translation:

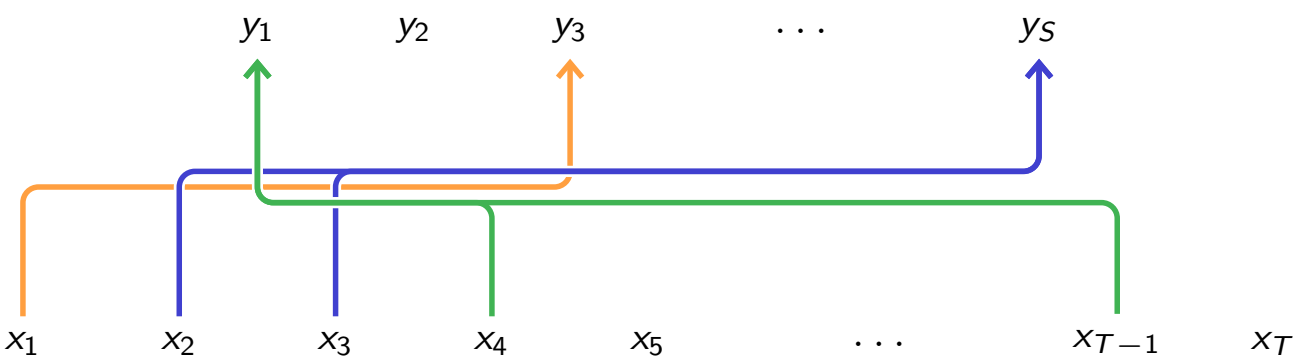
> “**An apple** that had been on the tree in the garden for weeks had finally been **picked up.**”
>
> “**Une pomme** qui était sur l’arbre du jardin depuis des semaines avait finalement été **ramassée.**”

Attention mechanisms

However some tasks involve more than composing local structures, e.g. translation:

> "**An apple** that had been on the tree in the garden for weeks had finally been **picked up.**"
>
> "**Une pomme** qui était sur l'arbre du jardin depuis des semaines avait finalement été **ramassée.**"

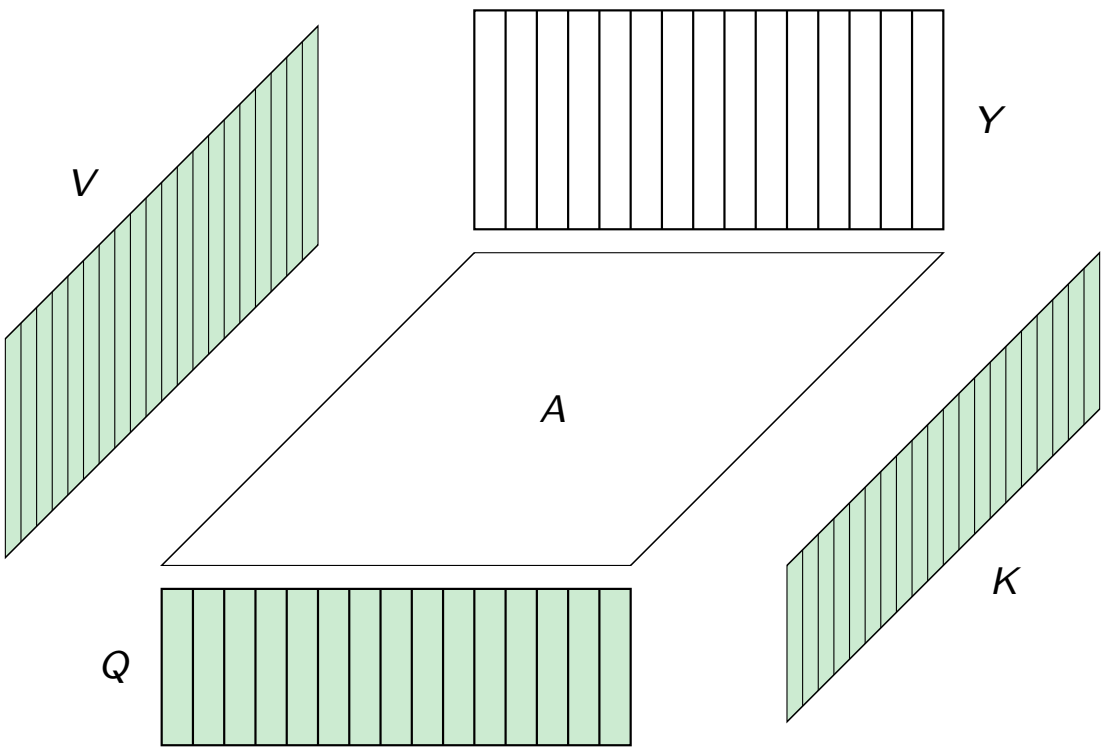
It has motivated the development of **attention-based processing** to transport information from parts of the signal to other parts dynamically identified.



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

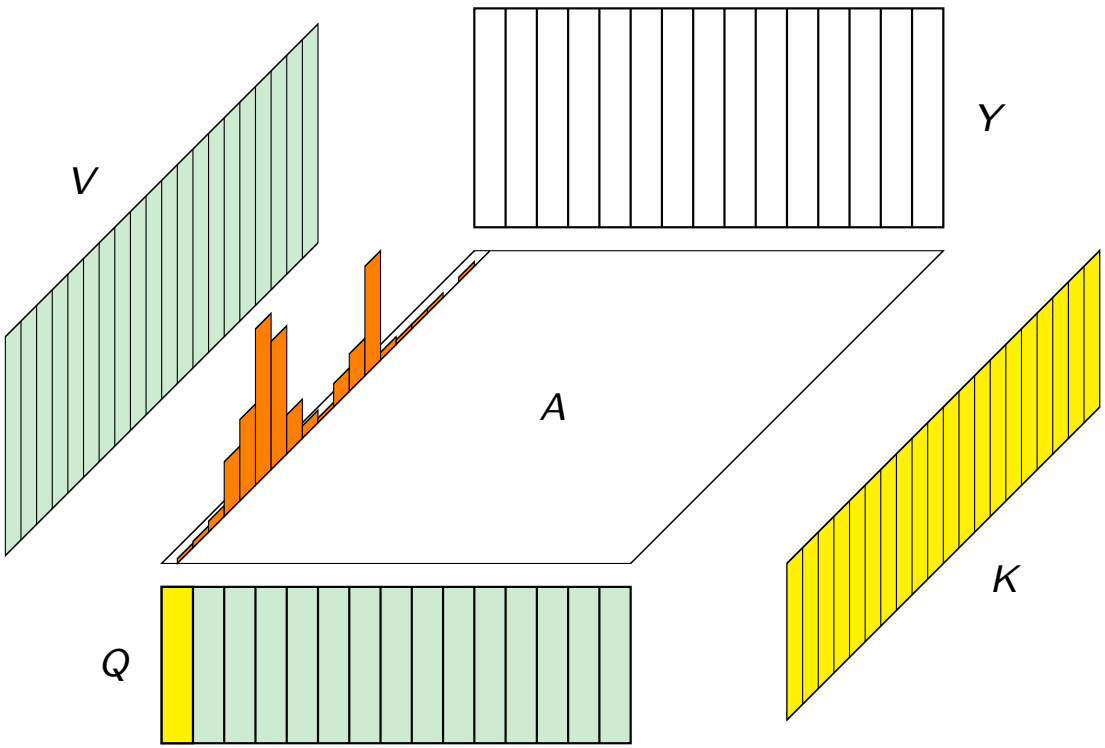
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

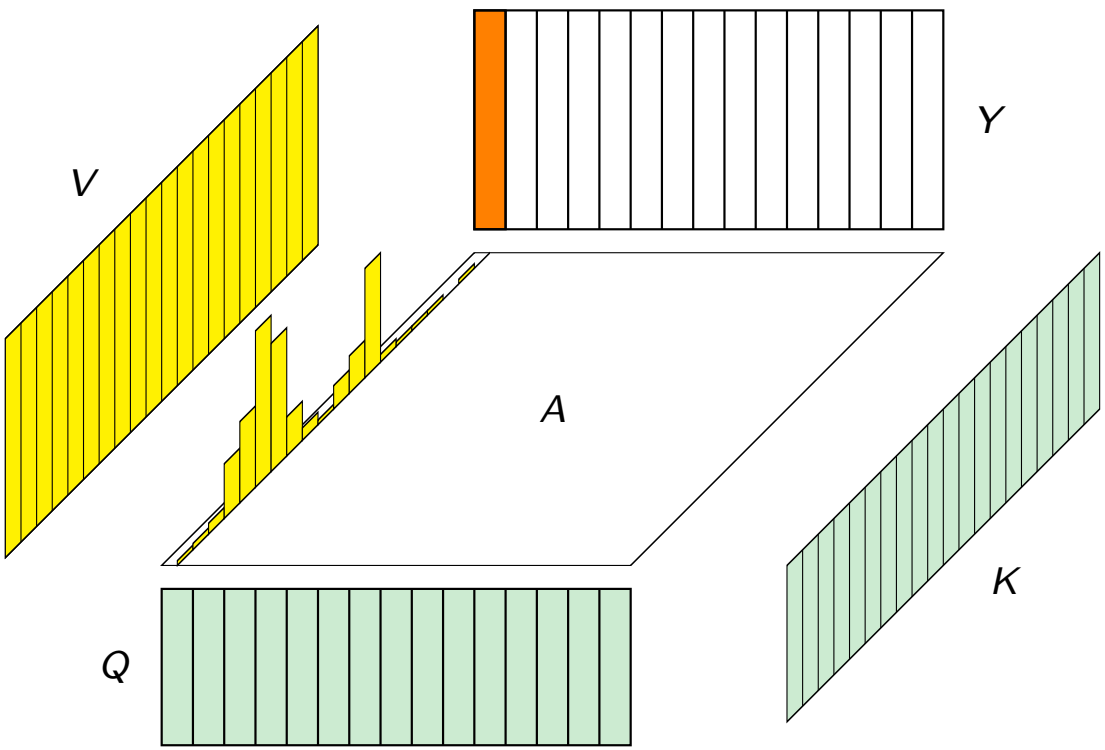
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

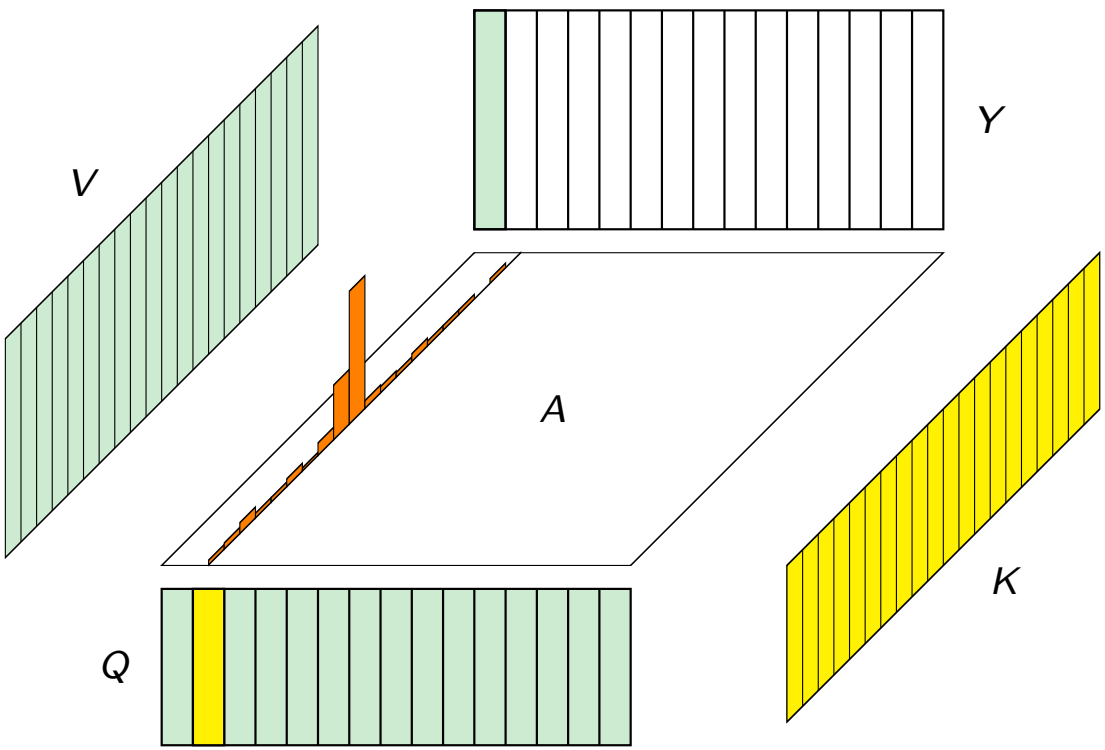
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

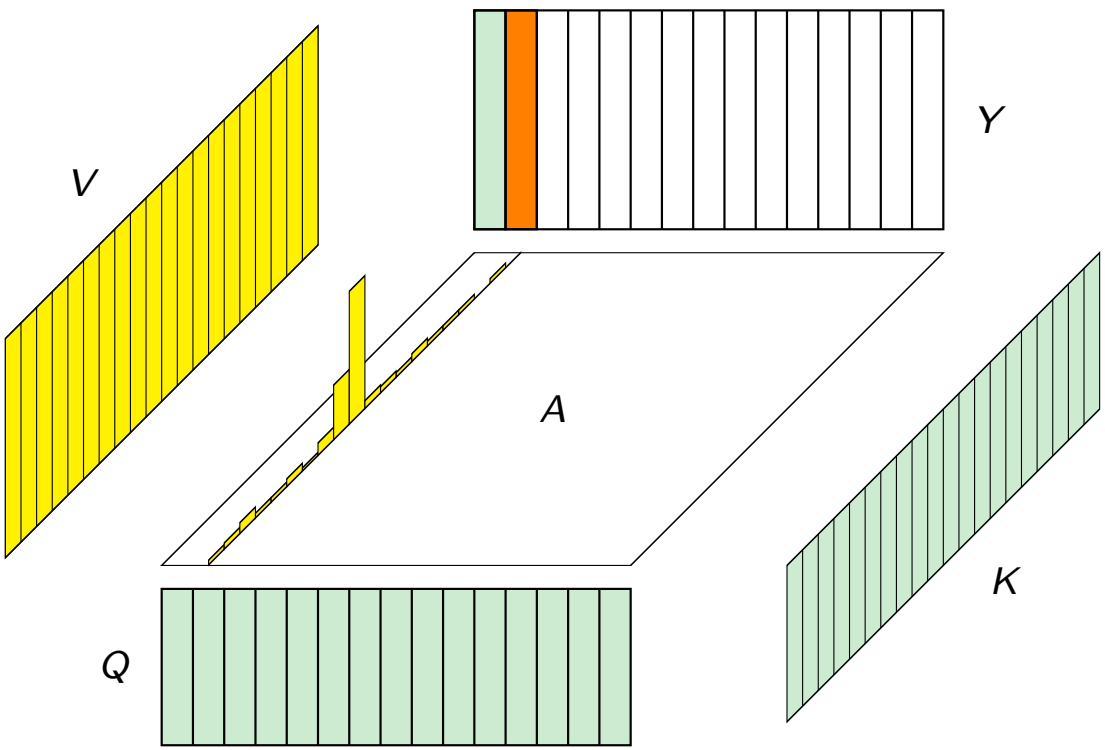
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

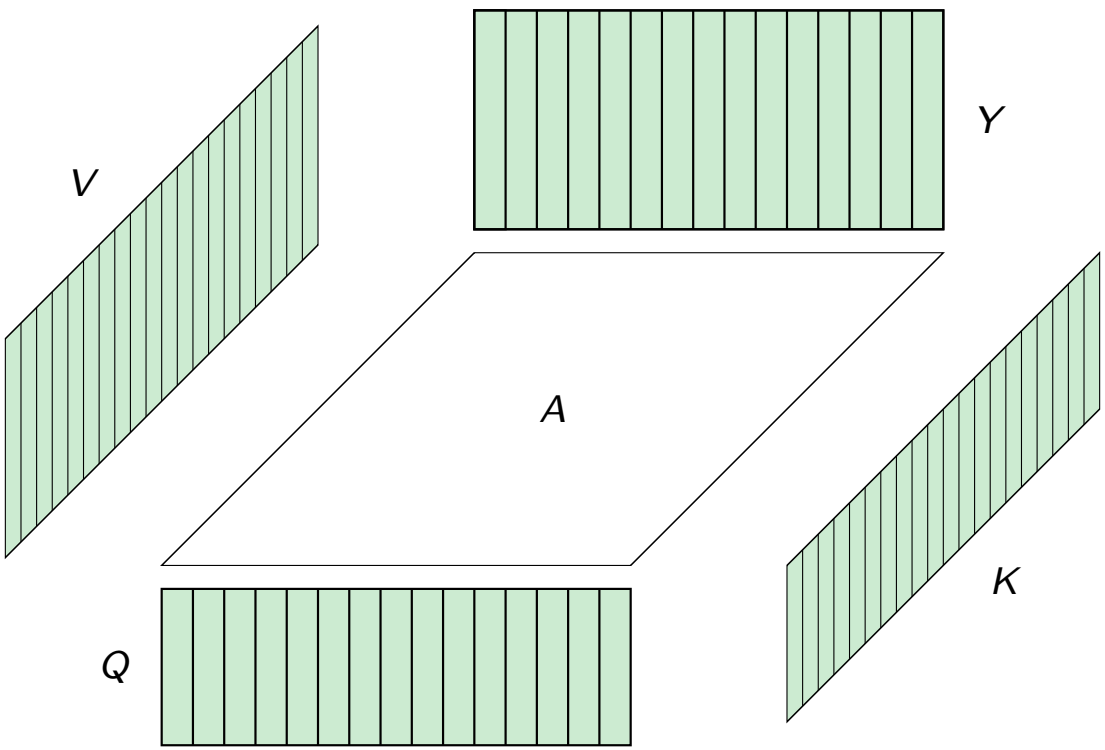
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

Given a query sequence Q, a key sequence K, and a value sequence V, compute an attention matrix A by matching Qs to Ks, and weight V with it to get the sequence Y.

$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right) \qquad Y = AV$$



Attention mechanisms

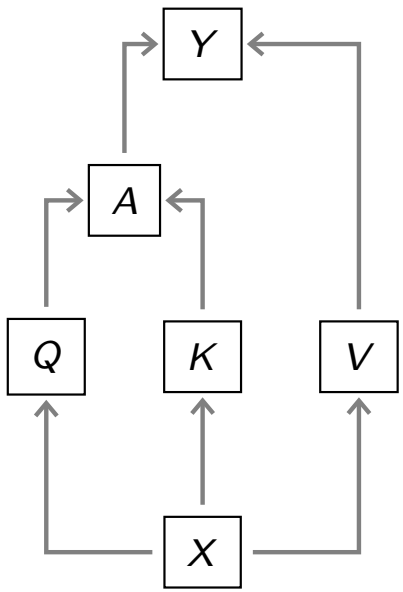
A standard attention layer takes as input two sequences X and X' and computes

$$Q = X W^{Q\top}$$
$$K = X' W^{K\top}$$
$$V = X' W^{V\top}$$
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right)$$
$$Y = AV$$

Attention mechanisms

A standard attention layer takes as input two sequences X and X' and computes

$$Q = XW^{Q\top}$$
$$K = X'W^{K\top}$$
$$V = X'W^{V\top}$$
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right)$$
$$Y = AV$$

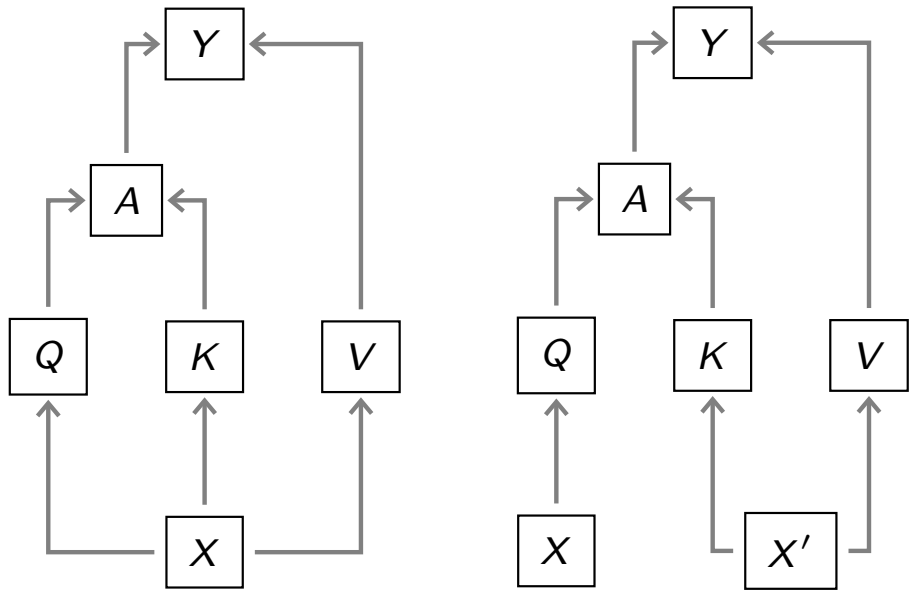


When $X = X'$, this is **self attention**

Attention mechanisms

A standard attention layer takes as input two sequences X and X' and computes

$$Q = XW^{Q\top}$$
$$K = X'W^{K\top}$$
$$V = X'W^{V\top}$$
$$A = \text{softmax}_{row}\left(\frac{QK^\top}{\sqrt{d}}\right)$$
$$Y = AV$$



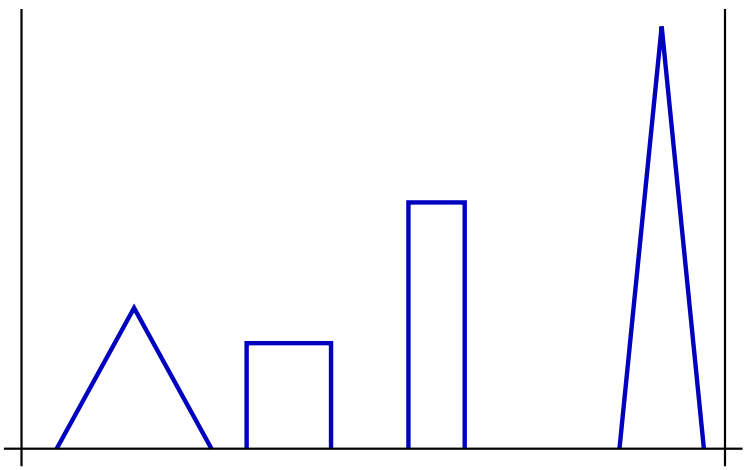
When $X = X'$, this is **self attention**, otherwise **cross attention.**

Toy seq2seq example

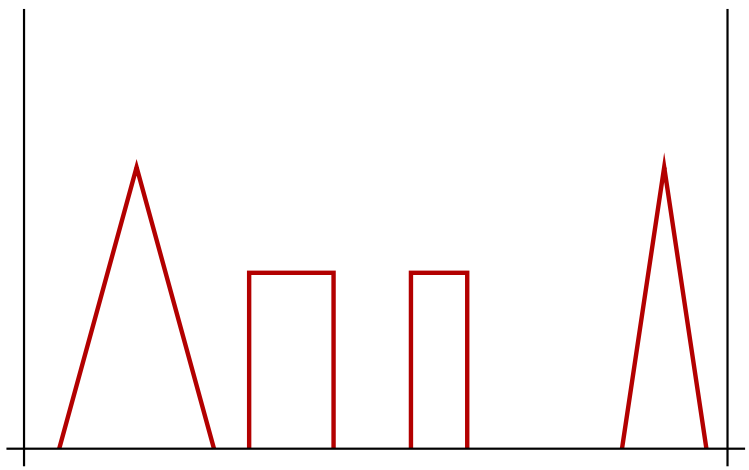
Consider a task with 1d sequences composed of two triangles and two rectangles, where the goal is to average heights in each **pair of shapes**.

Toy seq2seq example

Consider a task with 1d sequences composed of two triangles and two rectangles, where the goal is to average heights in each **pair of shapes**.



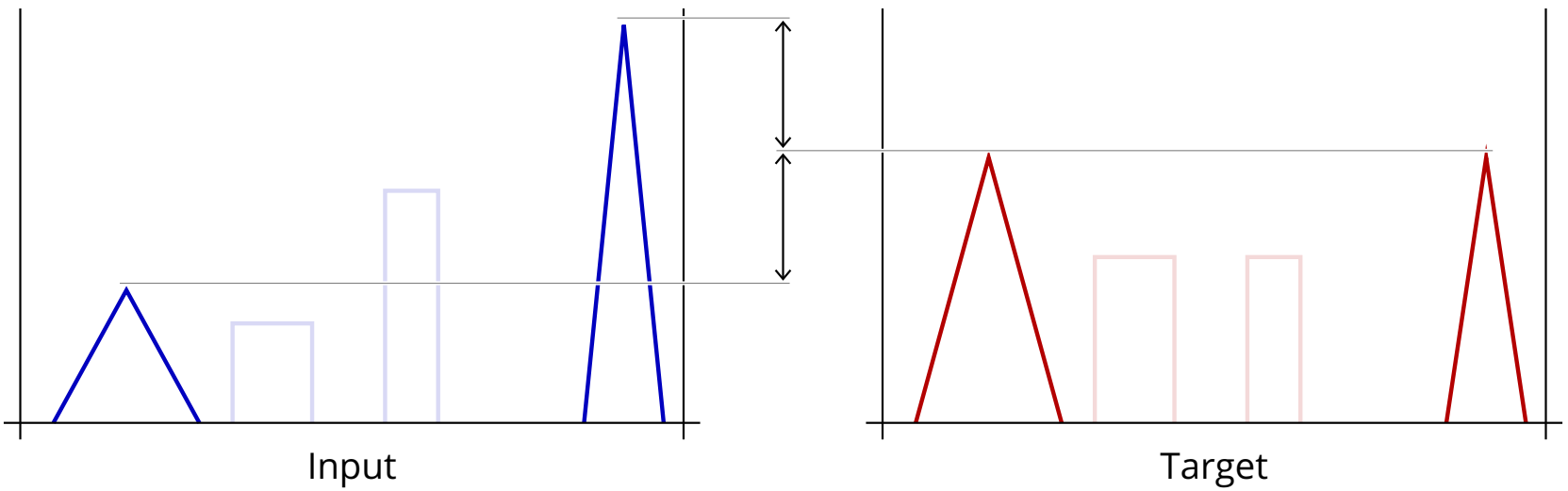
Input



Target

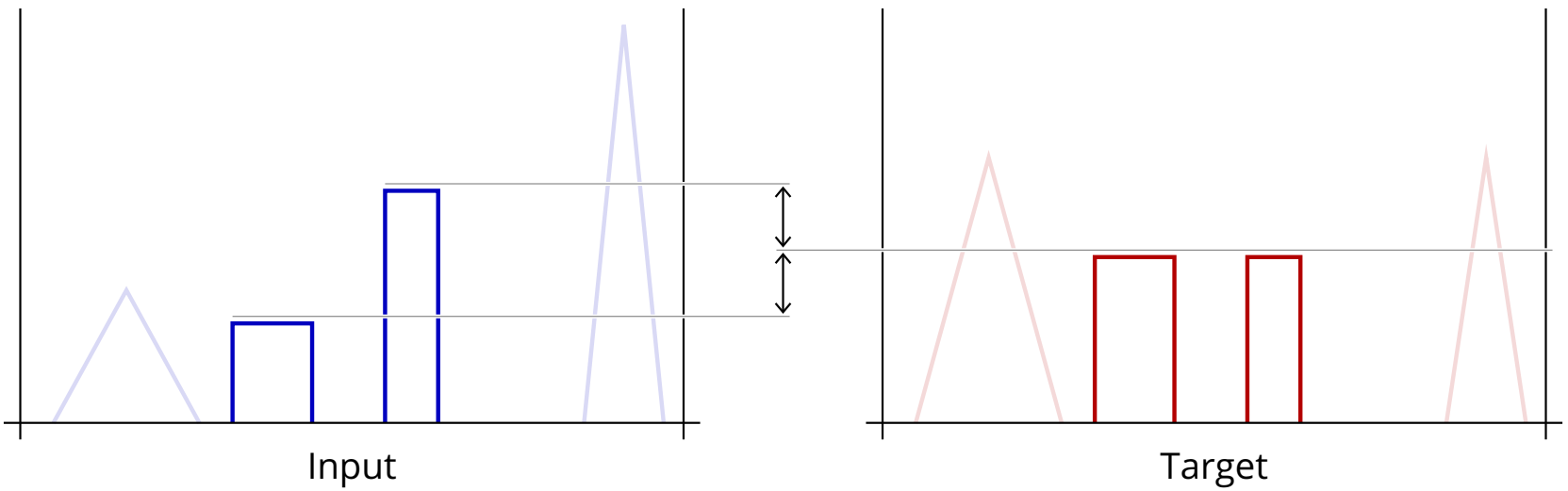
Toy seq2seq example

Consider a task with 1d sequences composed of two triangles and two rectangles, where the goal is to average heights in each **pair of shapes**.



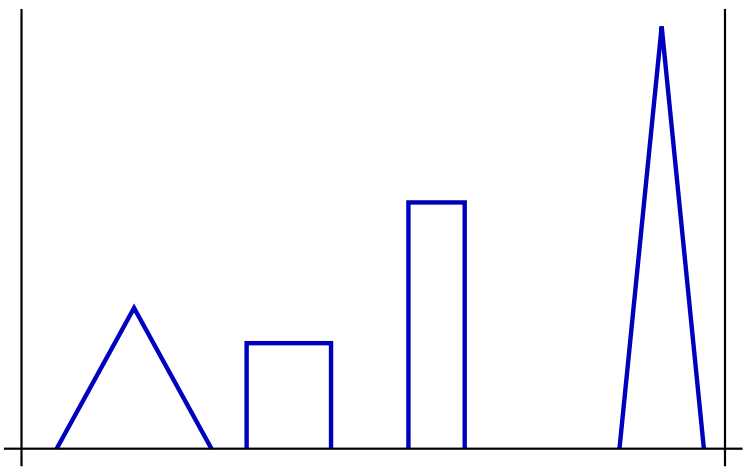
Toy seq2seq example

Consider a task with 1d sequences composed of two triangles and two rectangles, where the goal is to average heights in each **pair of shapes**.

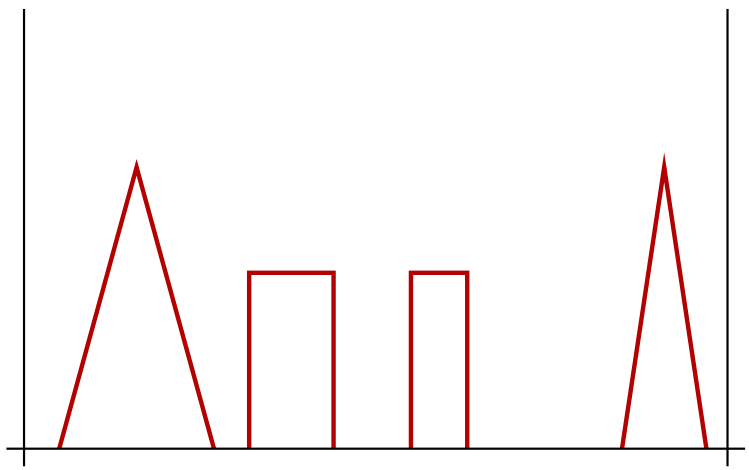


Toy seq2seq example

Consider a task with 1d sequences composed of two triangles and two rectangles, where the goal is to average heights in each **pair of shapes**.

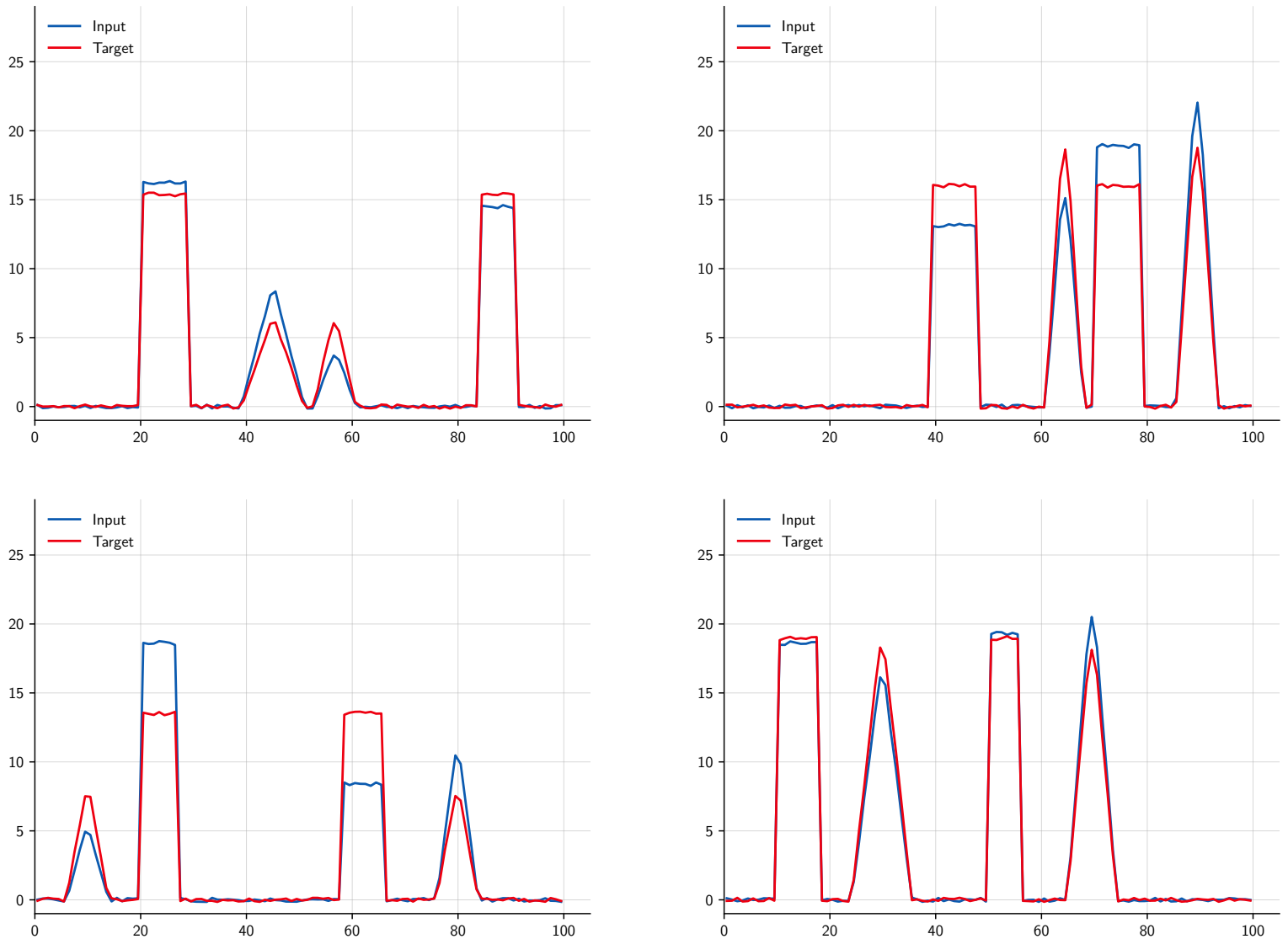


Input



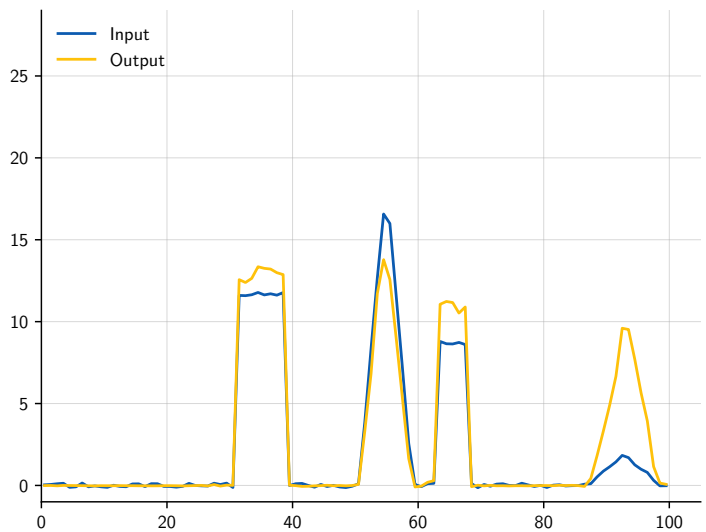
Target

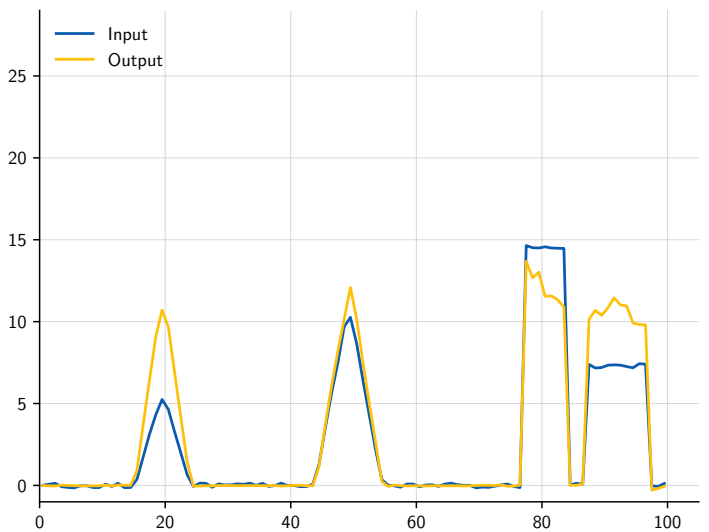
Toy seq2seq example



Toy seq2seq example

```
Sequential(
  (0): Conv1d(1, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (1): ReLU()
  (2): Conv1d(64, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (3): ReLU()
  (4): Conv1d(64, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (5): ReLU()
  (6): Conv1d(64, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (7): ReLU()
  (8): Conv1d(64, 1, kernel_size=(5,), stride=(1,), padding=(2,))
)
```





Toy seq2seq example

The poor performance of this model is not surprising given its inability to channel information from “far away” in the signal.

More layers, global averaging, or fully connected layers could possibly solve the problem. However it is more natural to equip the model with the ability to fetch information from parts of the signal that it actively identifies as relevant.

This is exactly what an **attention layer** does.

Toy seq2seq example

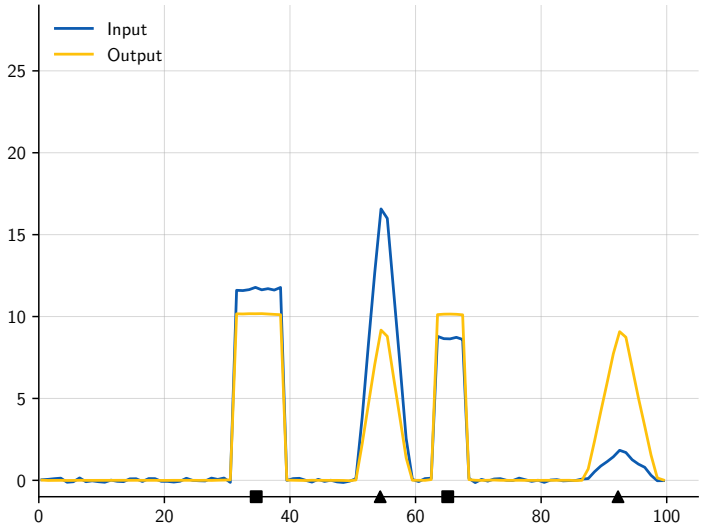
```
class SelfAttentionLayer(nn.Module):
    def __init__(self, in_dim, out_dim, key_dim):
        super().__init__()
        self.conv_Q = nn.Conv1d(in_dim, key_dim, kernel_size = 1, bias = False)
        self.conv_K = nn.Conv1d(in_dim, key_dim, kernel_size = 1, bias = False)
        self.conv_V = nn.Conv1d(in_dim, out_dim, kernel_size = 1, bias = False)

    def forward(self, x):
        Q = self.conv_Q(x)
        K = self.conv_K(x)
        V = self.conv_V(x)
        A = torch.einsum('nct,ncs->nts', Q, K).softmax(2)
        y = torch.einsum('nts,ncs->nct', A, V)
        return y
```

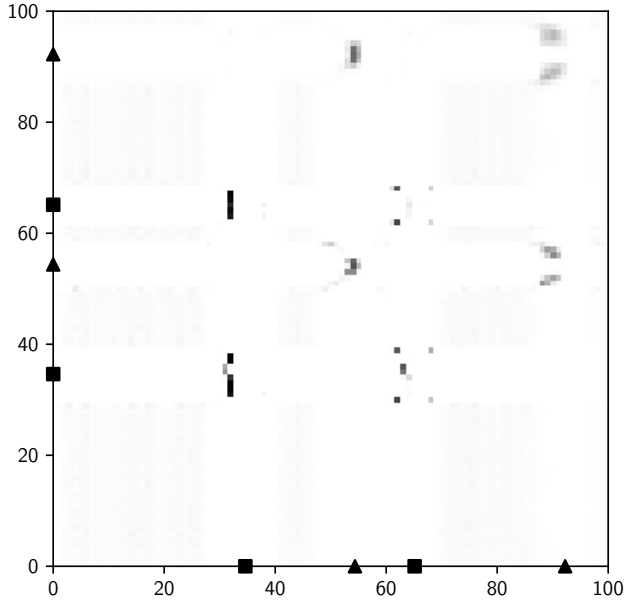
Toy seq2seq example

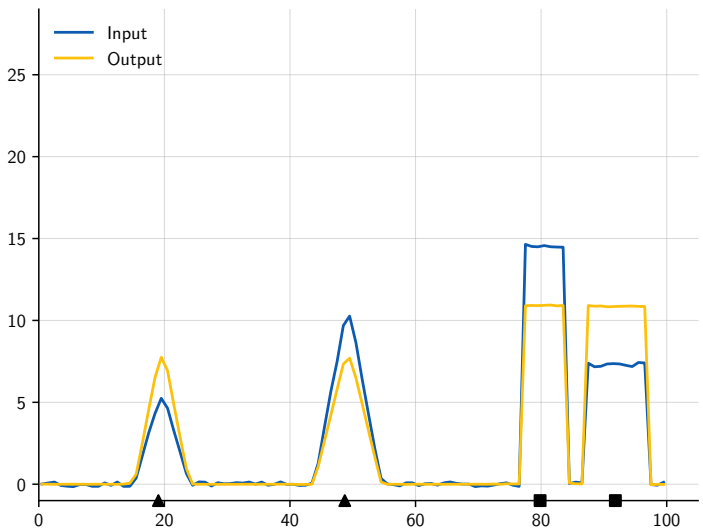
```
class SelfAttentionLayer(nn.Module):
    def __init__(self, in_dim, out_dim, key_dim):
        super().__init__()
        self.conv_Q = nn.Conv1d(in_dim, key_dim, kernel_size = 1, bias = False)
        self.conv_K = nn.Conv1d(in_dim, key_dim, kernel_size = 1, bias = False)
        self.conv_V = nn.Conv1d(in_dim, out_dim, kernel_size = 1, bias = False)

    def forward(self, x):
        Q = self.conv_Q(x)
        K = self.conv_K(x)
        V = self.conv_V(x)
        A = torch.einsum('nct,ncs->nts', Q, K).softmax(2)
        y = torch.einsum('nts,ncs->nct', A, V)
        return y
```

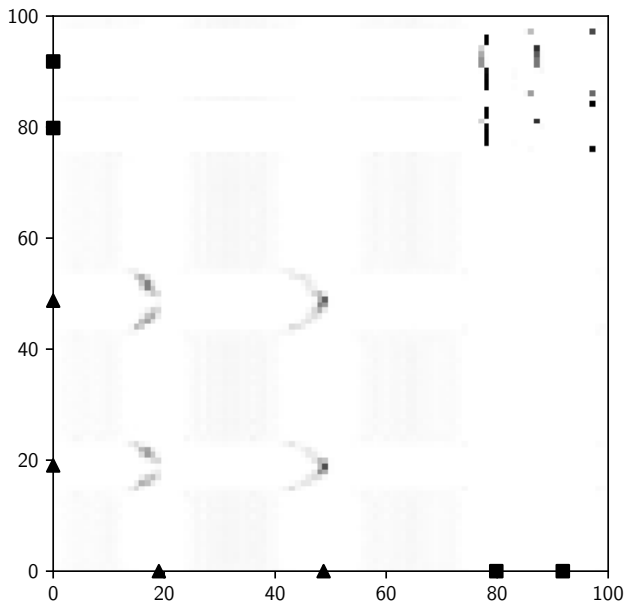
```
Sequential(
  (0): Conv1d(1, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (1): ReLU()
  (2): Conv1d(64, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (3): ReLU()
  (4): SelfAttentionLayer(in_channels=64, out_channels=64, key_channels=64)
  (5): Conv1d(64, 64, kernel_size=(5,), stride=(1,), padding=(2,))
  (6): ReLU()
  (7): Conv1d(64, 1, kernel_size=(5,), stride=(1,), padding=(2,))
)
```

Toy seq2seq example



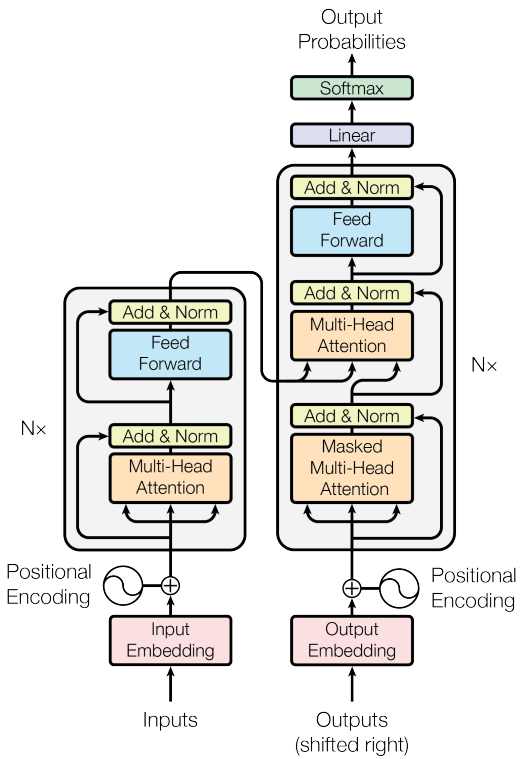






Transformers

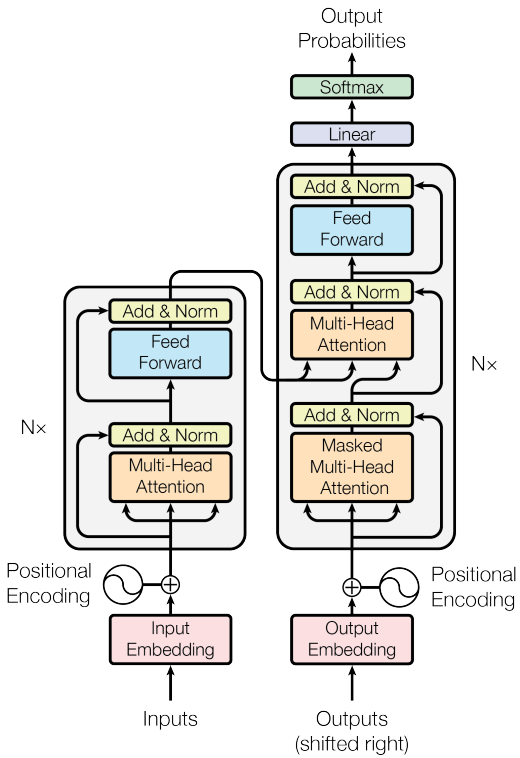
The original transformer combines a stack of self-attention layers in an encoder, and a stack of causal self-attention and cross-attention layers in a decoder.



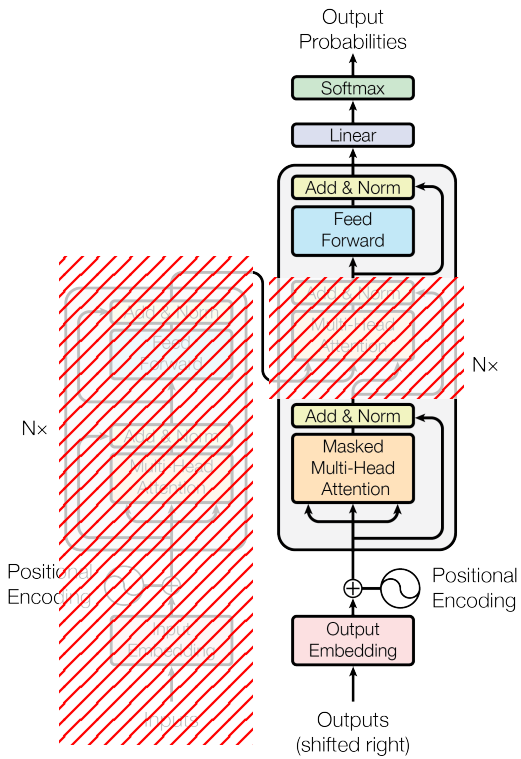
(Vaswani et al., 2017)

Transformers

The original transformer combines a stack of self-attention layers in an encoder, and a stack of causal self-attention and cross-attention layers in a decoder. The GPT model keeps only a causal encoder.



(Vaswani et al., 2017)



GPT

Transformers

Large language models have been shown to exhibit some “zero shot learning” capabilities when they are properly “primed” (Brown et al., 2020).

For instance using Hugging Face’s `gpt2` model with 120M parameters, we can get these sentence completions, where the generated parts are in bold:

I: water boils at 100 degrees, O: physics, I: the square root of two is irrational, O: mathematics, I: the set of prime numbers is infinite, O: mathematics, I: gravity is proportional to the mass, O: **physics,**

I: water boils at 100 degrees, O: physics, I: the square root of two is irrational, O: mathematics, I: the set of prime numbers is infinite, O: mathematics, I: squares are rectangles, O: **mathematics,**

Transformers

Large language models have been shown to exhibit some “zero shot learning” capabilities when they are properly “primed” (Brown et al., 2020).

For instance using Hugging Face’s `gpt2` model with 120M parameters, we can get these sentence completions, where the generated parts are in bold:

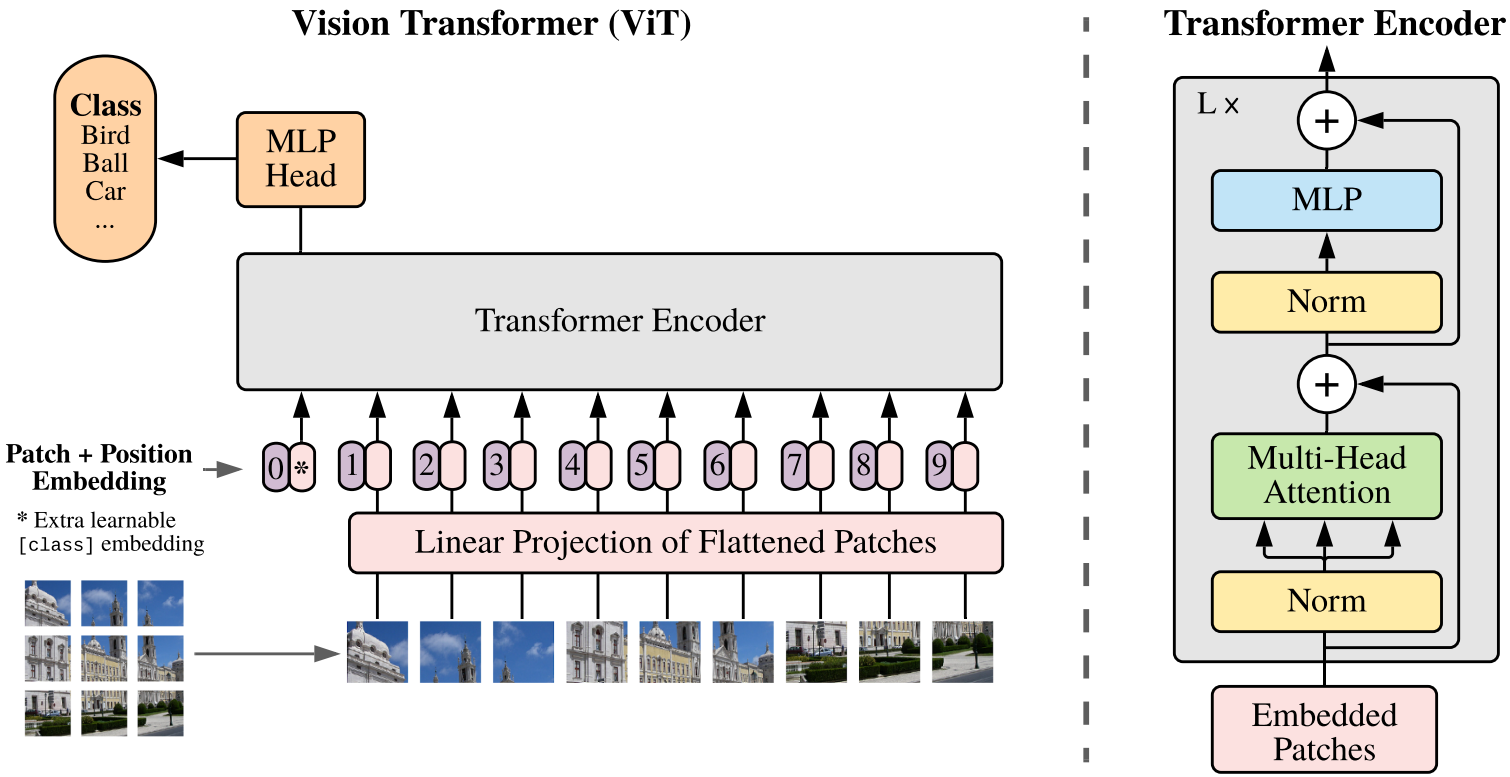
I: water boils at 100 degrees, O: physics, I: the square root of two is irrational, O: mathematics, I: the set of prime numbers is infinite, O: mathematics, I: gravity is proportional to the mass, O: **physics,**

I: water boils at 100 degrees, O: physics, I: the square root of two is irrational, O: mathematics, I: the set of prime numbers is infinite, O: mathematics, I: squares are rectangles, O: **mathematics,**

I: I love apples, O: positive, I: music is my passion, O: positive, I: my job is boring, O: negative, I: frozen pizzas are awesome, O: **positive,**

I: I love apples, O: positive, I: music is my passion, O: positive, I: my job is boring, O: negative, I: frozen pizzas taste like cardboard, O: **negative,**

Transformers



(Dosovitskiy et al., 2020)

GPTS FOR WORLD MODELS

World model

Reinforcement learning is in general sample-inefficient, since learning a policy

$$\pi : \mathcal{S} \to \mathcal{A}$$
$$X_t \mapsto A_t$$

to maximize an accumulated reward requires to propagate back rewards from the sparse states where it occurs.

World model

Reinforcement learning is in general sample-inefficient, since learning a policy

$$\pi : \mathcal{S} \to \mathcal{A}$$
$$X_t \mapsto A_t$$

to maximize an accumulated reward requires to propagate back rewards from the sparse states where it occurs.

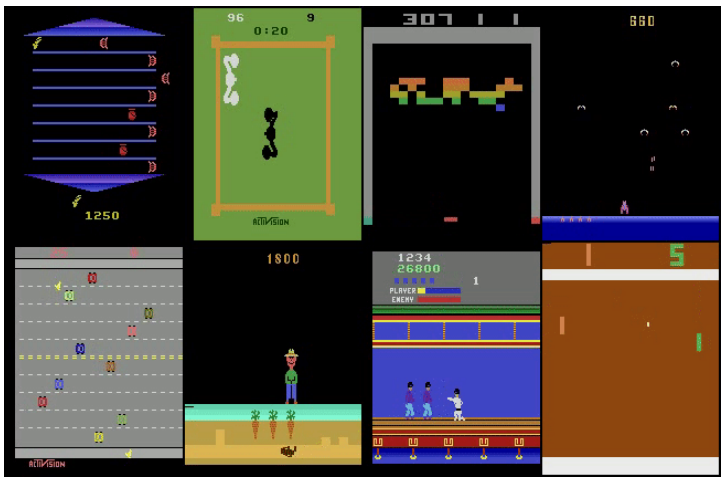
A solution is to train a **world model** that is a model of

$$P(X_{t+1} \mid A_s, X_s, s \leq t)$$

that does not need a sophisticated "lookahead", and then to train an agent in that model. For image-based RL, this is close to next-frame prediction.

World model

We focus on training with limited samples, in particular Atari with 100k frames.

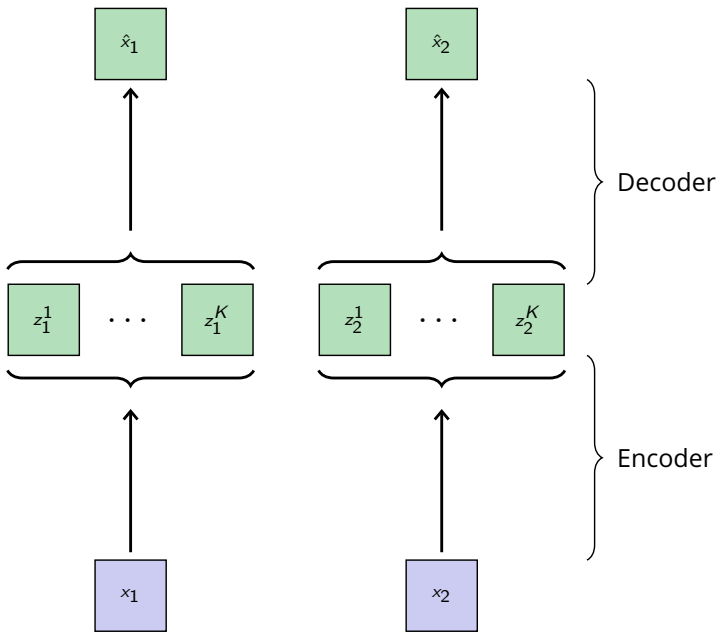


(fleuret.org/vid)

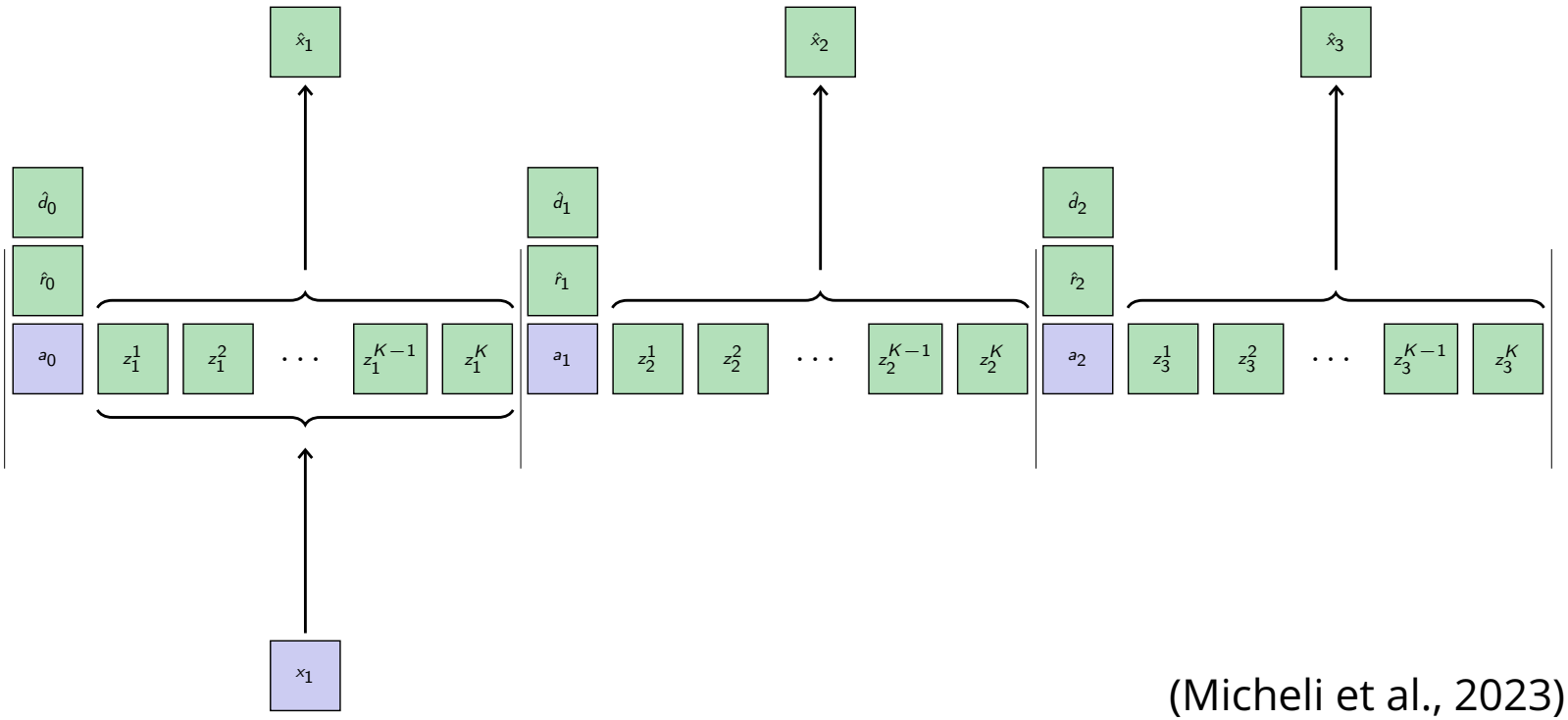
This is a standard benchmark with varied dynamics and “semantics”.

IRIS

We developed IRIS, that combines a discrete autoencoder (VQ-GAN) with a GPT.



IRIS



(Micheli et al., 2023)

The agent is an LSTM-based model entirely trained in generated episodes.

IRIS

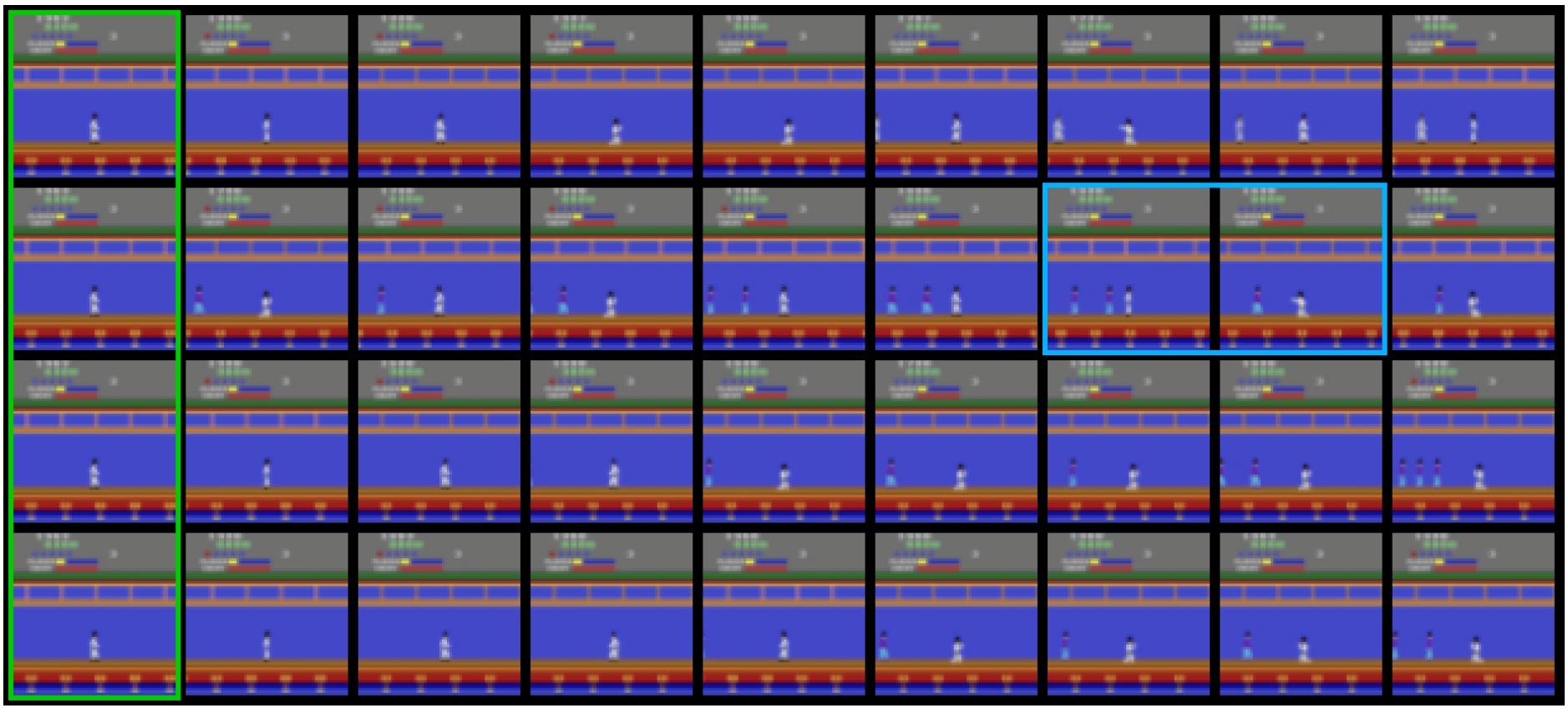
Repeat 500 times:

- From the current state, execute 200 steps in the environment ($\simeq$ 3s) with an ϵ-greedy policy.
- Add the frames/actions to the sample set.
- Improve the autoencoder and the GPT world model from the collected samples.
- Optimize the value network and the policy from generated trajectories.

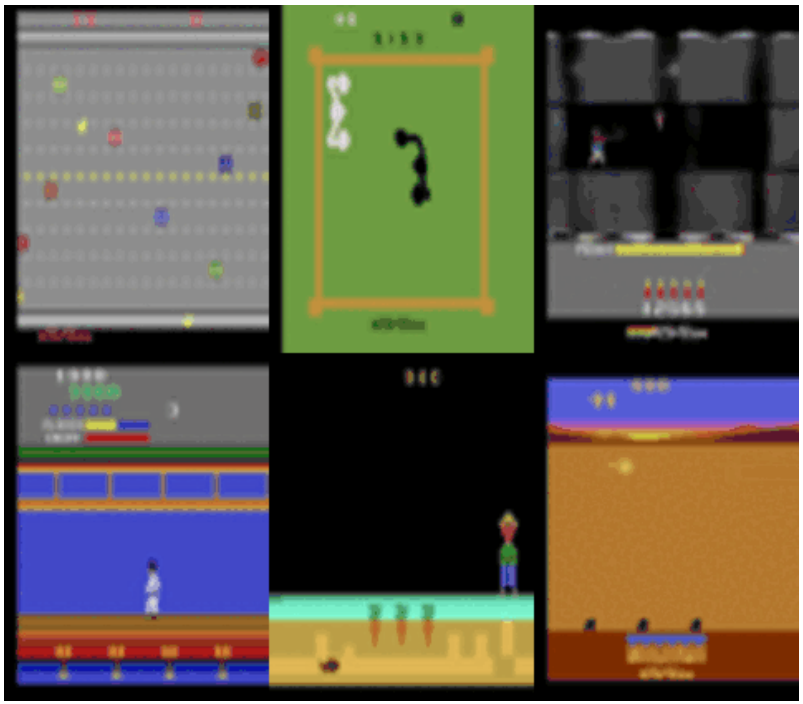
IRIS

			With lookahead		Without lookahead				
Game	Random	Human	MuZero	EfficientZero	SimPLe	CURL	DrQ	SPR	IRIS
Alien	227.8	7127.7	530.0	808.5	616.9	711.0	**865.2**	841.9	420.0
Amidar	5.8	1719.5	38.8	148.6	74.3	113.7	137.8	**179.7**	143.0
Assault	222.4	742.0	500.1	1263.1	527.2	500.9	579.6	565.6	**1524.4**
Asterix	210.0	8503.3	1734.0	25557.8	**1128.3**	567.2	763.6	962.5	853.6
BankHeist	14.2	753.1	192.5	351.0	34.2	65.3	232.9	**345.4**	53.1
BattleZone	2360.0	37187.5	7687.5	13871.2	4031.2	8997.8	10165.3	**14834.1**	13074.0
Boxing	0.1	12.1	15.1	52.7	7.8	0.9	9.0	35.7	**70.1**
Breakout	1.7	30.5	48.0	414.1	16.4	2.6	19.8	19.6	**83.7**
ChopperCommand	811.0	7387.8	1350.0	1117.3	979.4	783.5	844.6	946.3	**1565.0**
CrazyClimber	10780.5	35829.4	56937.0	83940.2	**62583.6**	9154.4	21539.0	36700.5	59324.2
DemonAttack	152.1	1971.0	3527.0	13003.9	208.1	646.5	1321.5	517.6	**2034.4**
Freeway	0.0	29.6	21.8	21.8	16.7	28.3	20.3	19.3	**31.1**
Frostbite	65.2	4334.7	255.0	296.3	236.9	**1226.5**	1014.2	1170.7	259.1
Gopher	257.6	2412.5	1256.0	3260.3	596.8	400.9	621.6	660.6	**2236.1**
Hero	1027.0	30826.4	3095.0	9315.9	2656.6	4987.7	4167.9	5858.6	**7037.4**
Jamesbond	29.0	302.8	87.5	517.0	100.5	331.0	349.1	366.5	**462.7**
Kangaroo	52.0	3035.0	62.5	724.1	51.2	740.2	1088.4	**3617.4**	838.2
Krull	1598.0	2665.5	4890.8	5663.3	2204.8	3049.2	4402.1	3681.6	**6616.4**
KungFuMaster	258.5	22736.3	18813.0	30944.8	14862.5	8155.6	11467.4	14783.2	**21759.8**
MsPacman	307.3	6951.6	1265.6	1281.2	**1480.0**	1064.0	1218.1	1318.4	999.1
Pong	-20.7	14.6	-6.7	20.1	12.8	-18.5	-9.1	-5.4	**14.6**
PrivateEye	24.9	69571.3	56.3	96.7	35.0	81.9	3.5	86.0	**100.0**
Qbert	163.9	13455.0	3952.0	13781.9	1288.8	727.0	**1810.7**	866.3	745.7
RoadRunner	11.5	7845.0	2500.0	17751.3	5640.6	5006.1	11211.4	**12213.1**	9614.6
Seaquest	68.4	42054.7	208.0	1100.2	**683.3**	315.2	352.3	558.1	661.3
UpNDown	533.4	11693.2	2896.9	17264.2	3350.3	2646.4	4324.5	**10859.2**	3546.2
# Superhuman	0	N/A	5	14	1	2	3	6	**10**
Mean	0.000	1.000	0.562	1.943	0.332	0.261	0.465	0.616	**1.046**
Median	0.000	1.000	0.227	1.090	0.134	0.092	0.313	**0.396**	0.289

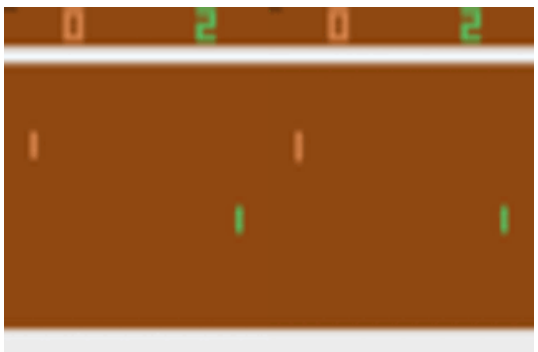
IRIS



IRIS



(fleuret.org/vid)



(fleuret.org/vid)

Δ-IRIS

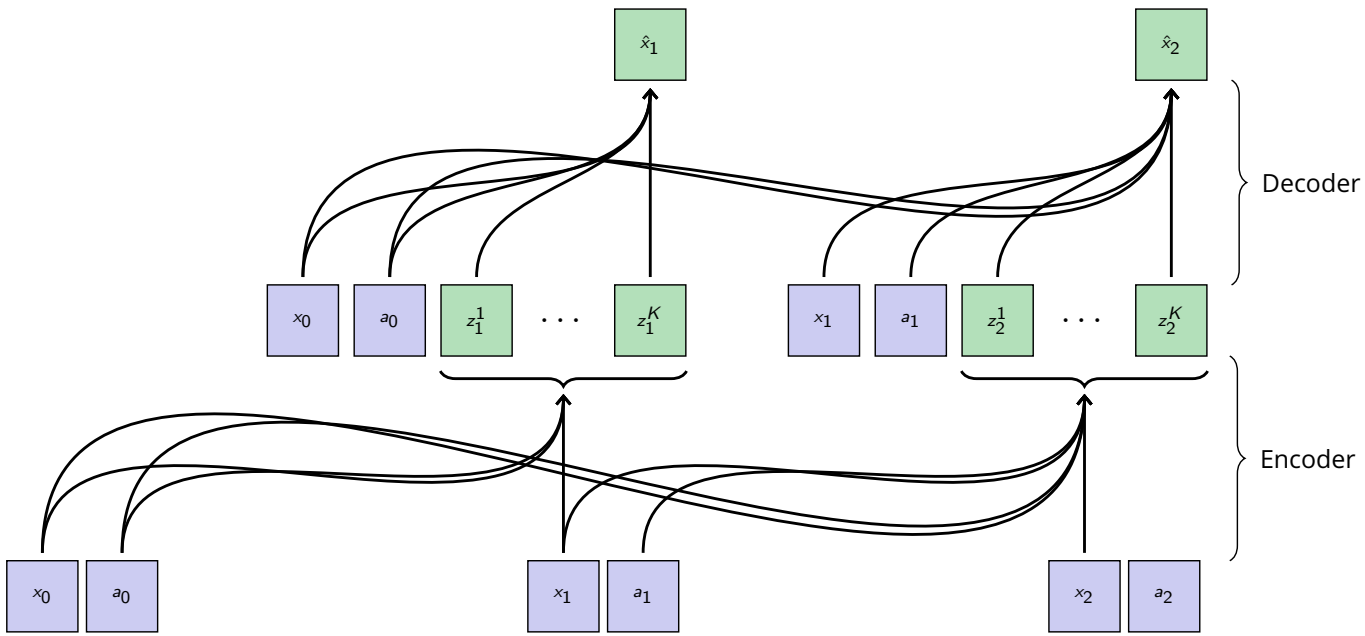
The core issue with IRIS is the number of tokens per frame. We had to use 16 tokens, leading to a substantial degradation of performance.

We address this with Δ-IRIS, which encodes a frame as a small number of discrete tokens **given the four previous frames and actions.**

Δ-IRIS

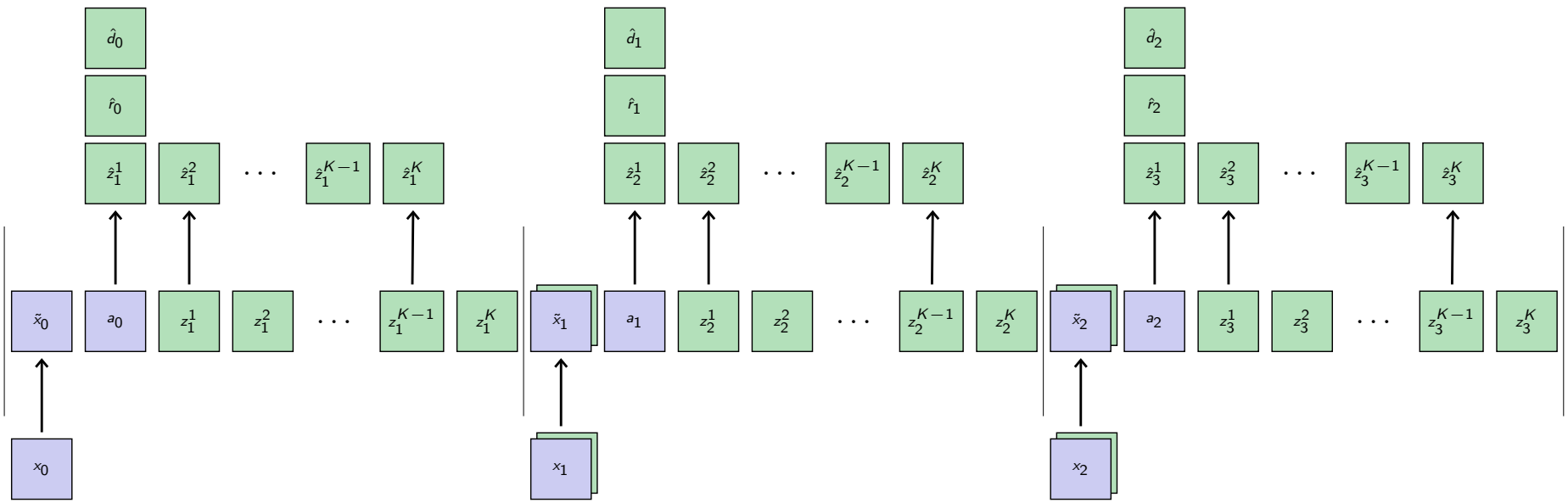
The core issue with IRIS is the number of tokens per frame. We had to use 16 tokens, leading to a substantial degradation of performance.

We address this with Δ-IRIS, which encodes a frame as a small number of discrete tokens **given the four previous frames and actions.**



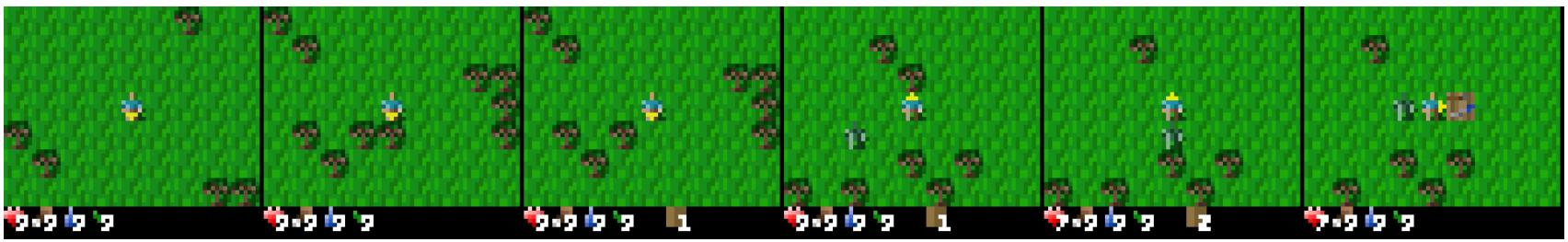
Δ-IRIS

A GPT generates the z_t^k, but attend to image tokens produced with the z_s^k, $s < t$.

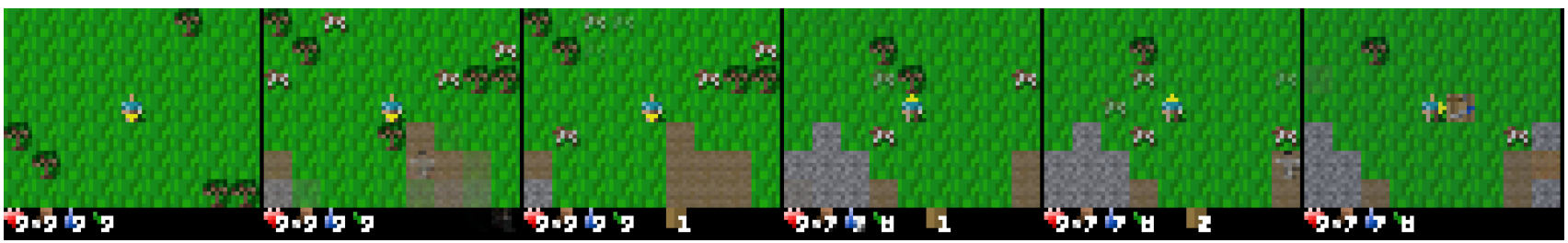


Δ-IRIS

We benchmark Δ-IRIS on Crafter. The z_t^k encode the residual randomness, given previous frames and actions.

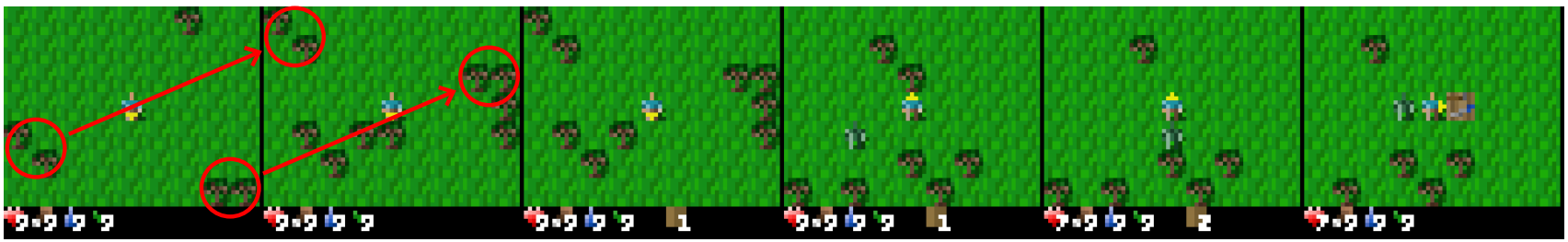


GPT-generated Δ-tokens

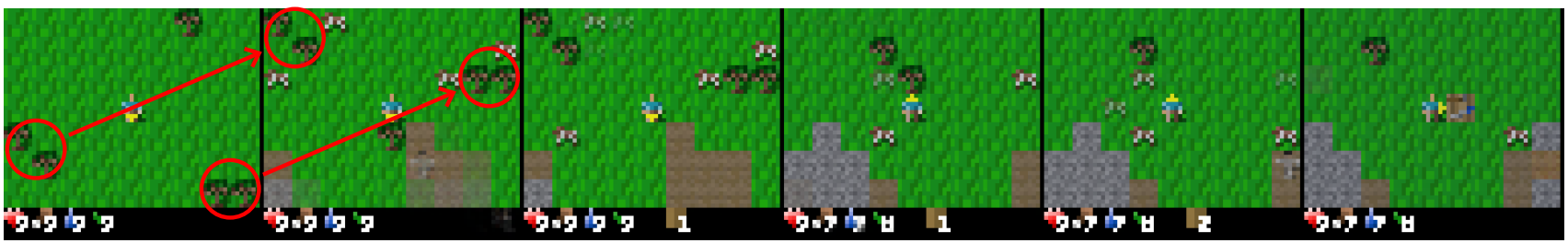


Random Δ-tokens

We benchmark Δ-IRIS on Crafter. The z_t^k encode the residual randomness, given previous frames and actions.

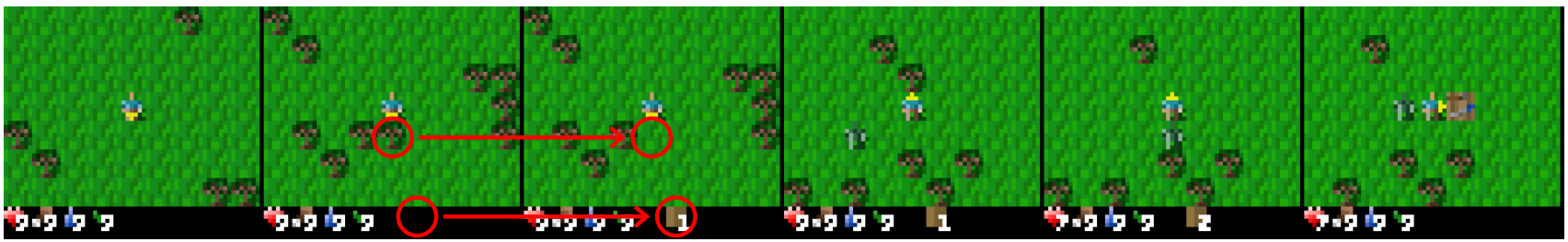


GPT-generated Δ-tokens

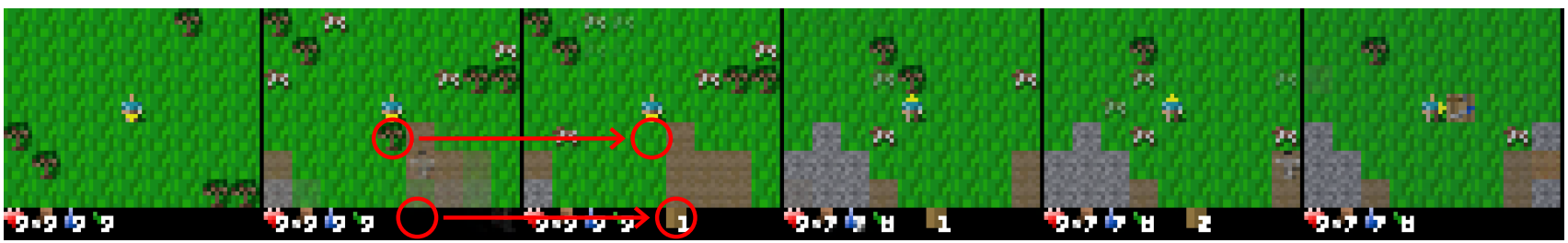


Random Δ-tokens

We benchmark Δ-IRIS on Crafter. The z_t^k encode the residual randomness, given previous frames and actions.

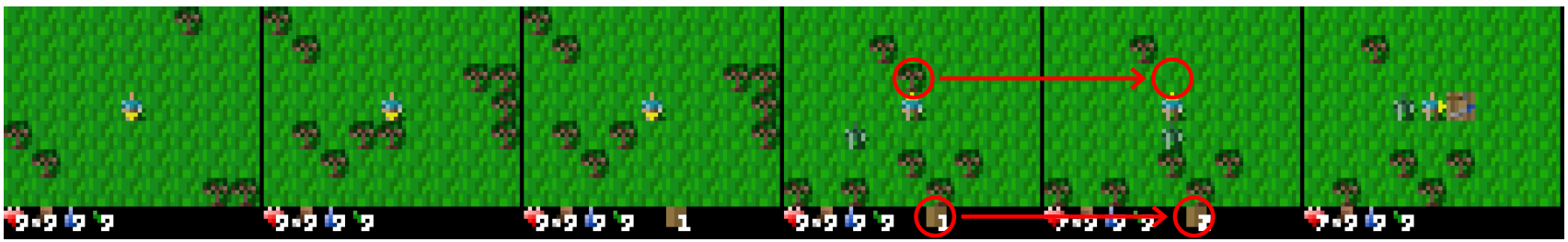


GPT-generated Δ-tokens

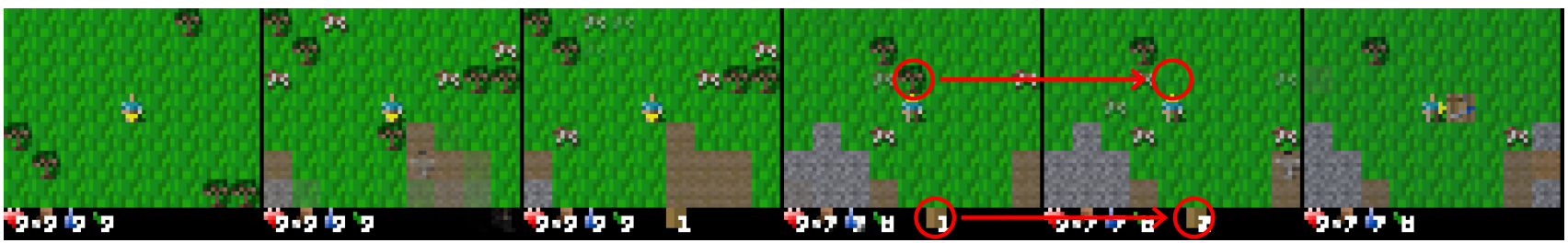


Random Δ-tokens

We benchmark Δ-IRIS on Crafter. The z_t^k encode the residual randomness, given previous frames and actions.



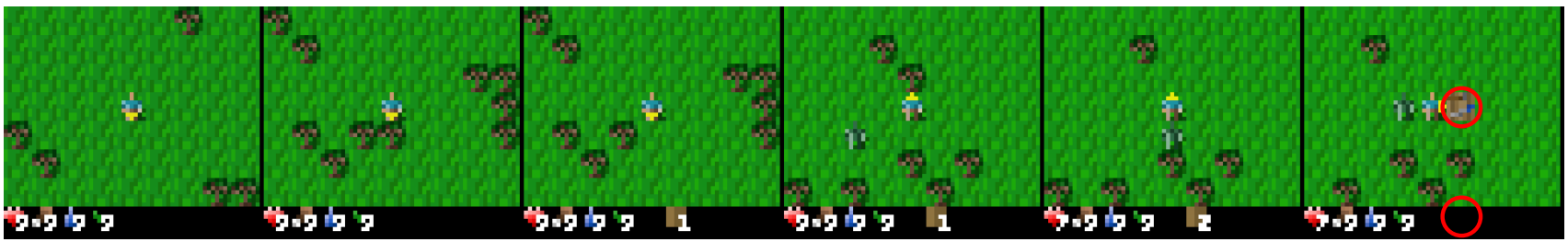
GPT-generated Δ-tokens



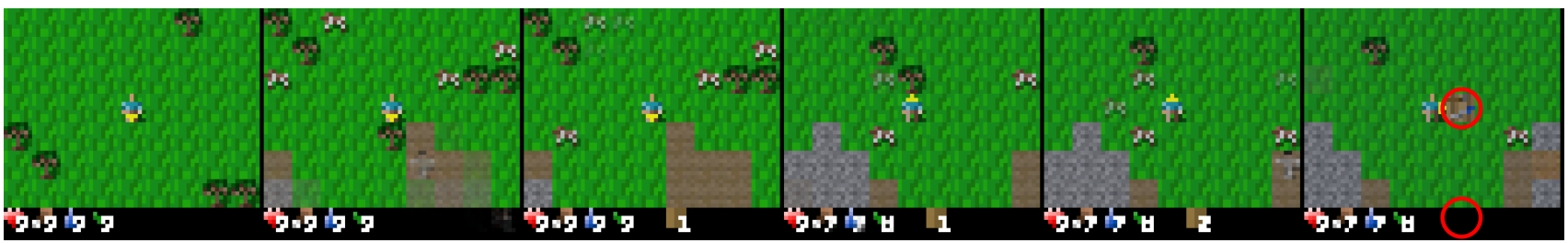
Random Δ-tokens

Δ-IRIS

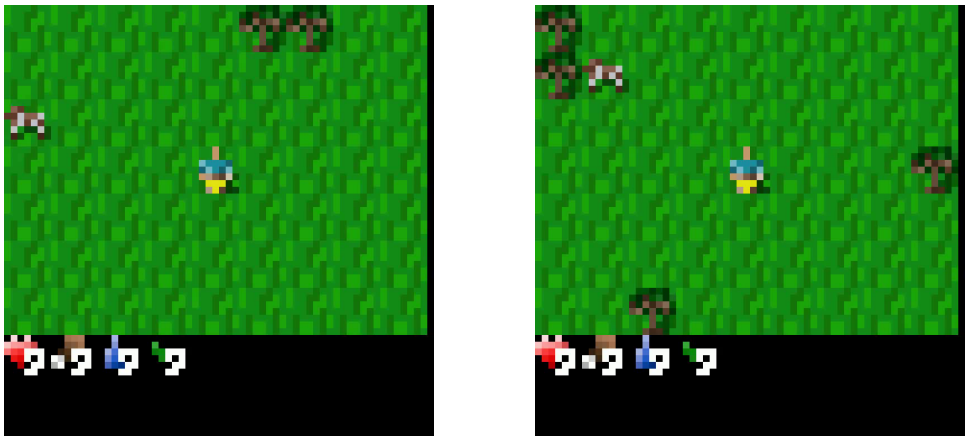
We benchmark Δ-IRIS on Crafter. The z_t^k encode the residual randomness, given previous frames and actions.



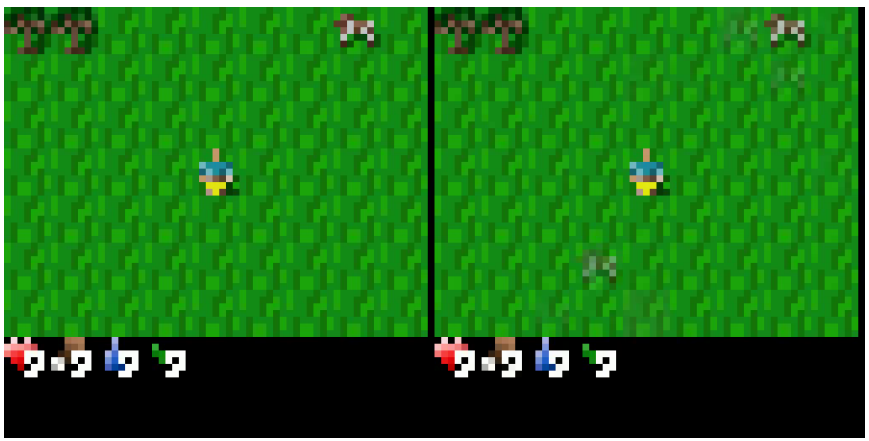
GPT-generated Δ-tokens



Random Δ-tokens



GPT



GPT vs. random tokens

Two key realizations:

- **discrete tokens are needed for prediction** to benefit from cross-entropy and deal with multi-modality, but past information can be processed and integrated into **continuous tokens.**

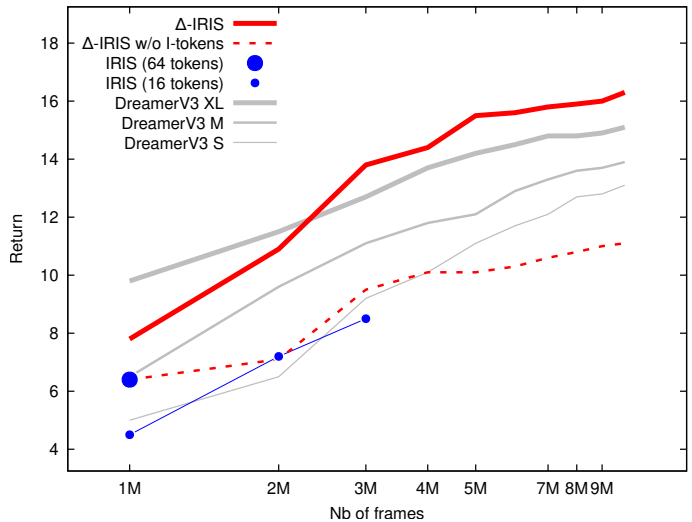
Two key realizations:

- **discrete tokens are needed for prediction** to benefit from cross-entropy and deal with multi-modality, but past information can be processed and integrated into **continuous tokens.**

- A perfect differential encoder would generate **totally random delta embeddings** and be in itself a world model.

Δ-IRIS

Crafter 10M frames, Δ-IRIS solves 17 out of Crafter's 22 tasks.



Atari 100k frames, Δ-IRIS is SOTA for # Superhuman.

Game	Random	Human	SimPLe	DreamerV3	STORM	IRIS	Δ-IRIS
Boxing	0	12	8	78	**80**	70	71
Breakout	2	30	16	31	16	84	**379**
Freeway	0	30	17	0	0	**31**	**31**
Gopher	258	2413	597	3730	8240	2236	**12916**
Krull	1598	2666	2204	7782	**8413**	6616	5681
KungFuMaster	259	22736	14862	21420	**26182**	21760	22051
Pong	-21	15	13	18	11	15	**19**
RoadRunner	12	7845	5641	15565	**17564**	9615	13526
# Superhuman	0	N/A	0	6	5	6	**7**
Mean	0.00	1.00	0.60	2.37	2.69	2.32	**4.24**
Median	0.00	1.00	0.61	1.35	1.70	1.13	**2.77**
Interquartile Mean	0.00	1.00	0.61	1.43	2.00	1.55	**3.13**

Conclusion

- The computational abilities of auto-regressive models are amazing.
- The inductive bias of transformers as auto-regressive models is amazing.
- Are the computational abilities leveraged by LLMs?
- Should spatial and motor understanding be learned first as a foundation for abstract reasoning?

BEYOND WORLD UNDERSTANDING

Reasoning Competition

[Warning! Computer scientist's armchair philosophy]

How can we go beyond reasoning tools for planing in the environment?

How to get high-level abstract "thinking" (mathematics, humor, poetry?)

Reasoning Competition

[Warning! Computer scientist's armchair philosophy]

How can we go beyond reasoning tools for planing in the environment?

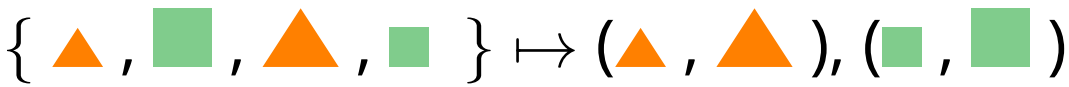
How to get high-level abstract "thinking" (mathematics, humor, poetry?)

An hypothesis is that this happens through reasoning challenges in social competition.

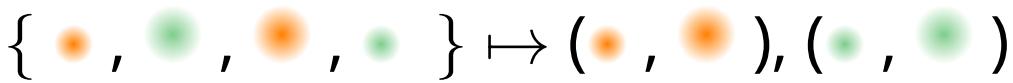
This allows to transfer capabilities from one model to the others and could trigger a "cognitive arm race" and go beyond reasoning tools to model the environment.

Reasoning Competition

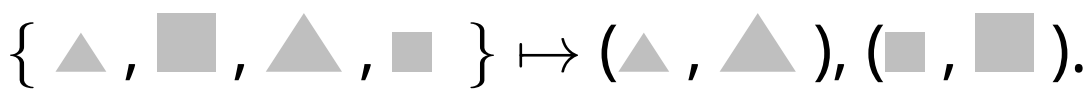
If, to operate in its environment, an agent has to solve



it can do it by being only “color aware”

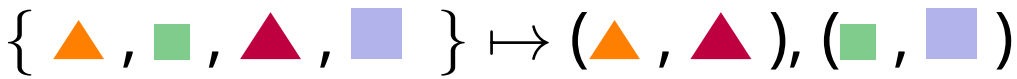


or “shape aware”



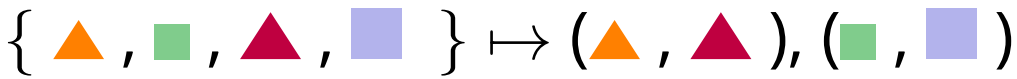
Reasoning Competition

A “shape aware” agent could craft a challenge that those who are not would fail

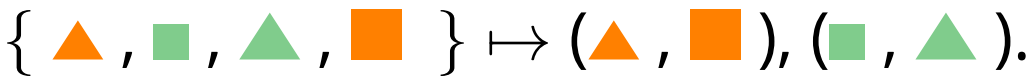


Reasoning Competition

A “shape aware” agent could craft a challenge that those who are not would fail



and similarly for a “color aware” agent



Reasoning Competition

To test this experimentally we use GPT models as agents and “quizzes” as elements of reasoning.

Let $\mathcal{G} = \{\text{“white”}, \text{“red”}, \dots\}^{10\times 10}$ be the set of colored 10×10 grids. A quiz consists of four grids that should be interpreted as

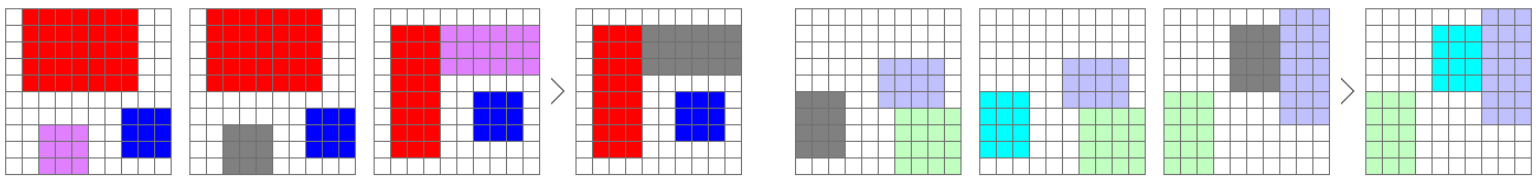
$$(A, f(A), B, f(B))$$

where $f : \mathcal{G} \to \mathcal{G}$ is a latent function.

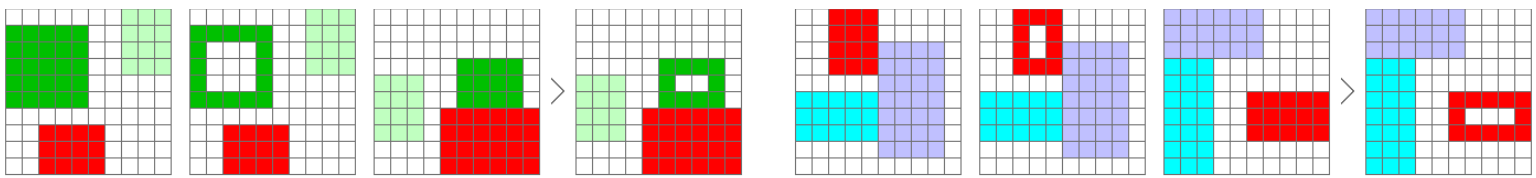
Such a quiz should be interpreted as a prompt composed of three 10×10 colored cell grids, and a solution which is a single grid.

World and Culture quizzes

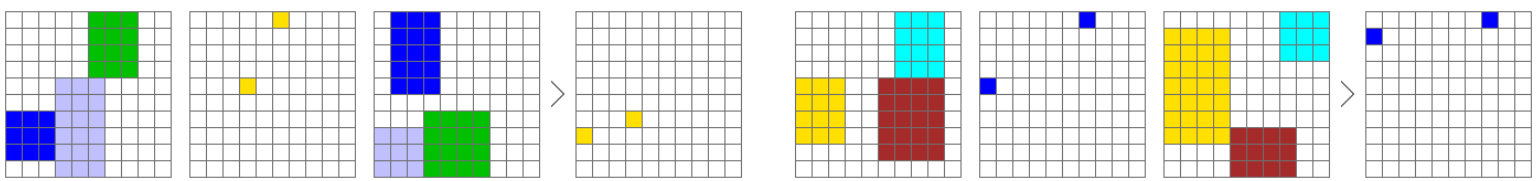
We define seven “tasks”, each of them being a distribution of “world quizzes”, such that the solution is unique given the prompt.



“Replace Color”

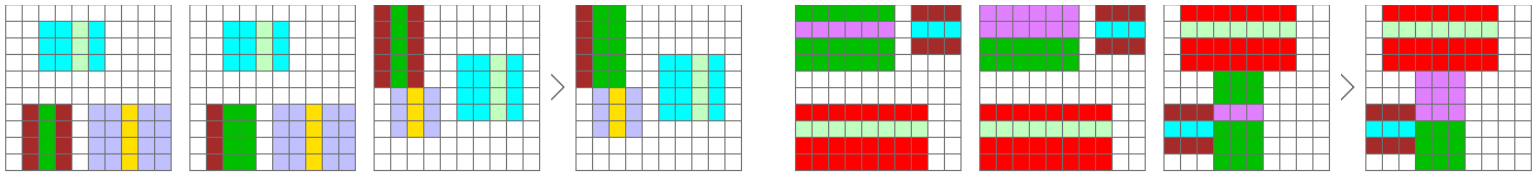


“Frame”

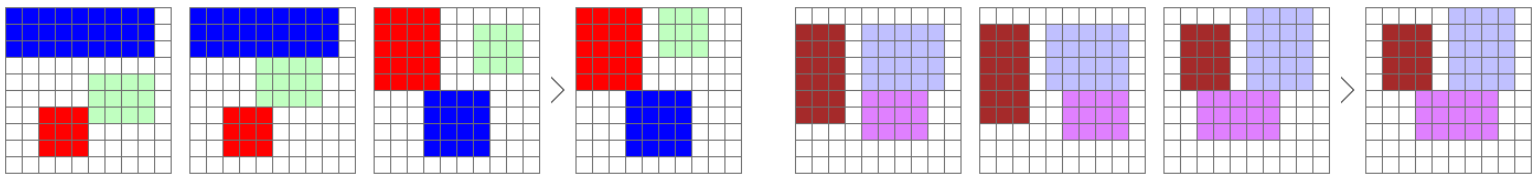


“Detect”

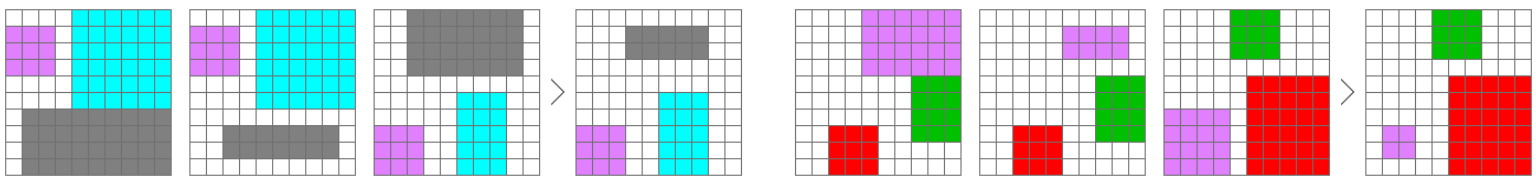
World and Culture quizzes



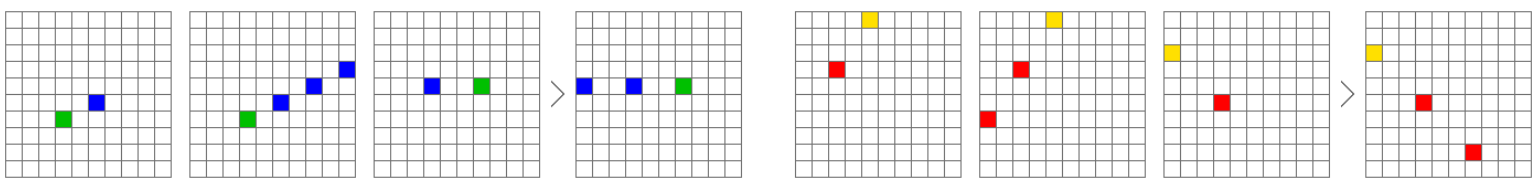
“Half Fill”



“Translate”



“Grow”



“Motion”

World and Culture quizzes

We differentiate:

- “world quizzes” whose distributions are those predefined tasks, and play the role of the cognitive challenges from the environment, and
- “culture quizzes” which are generated by agents, and whose role is to prove that its author masters certain concepts that its opponents do not.

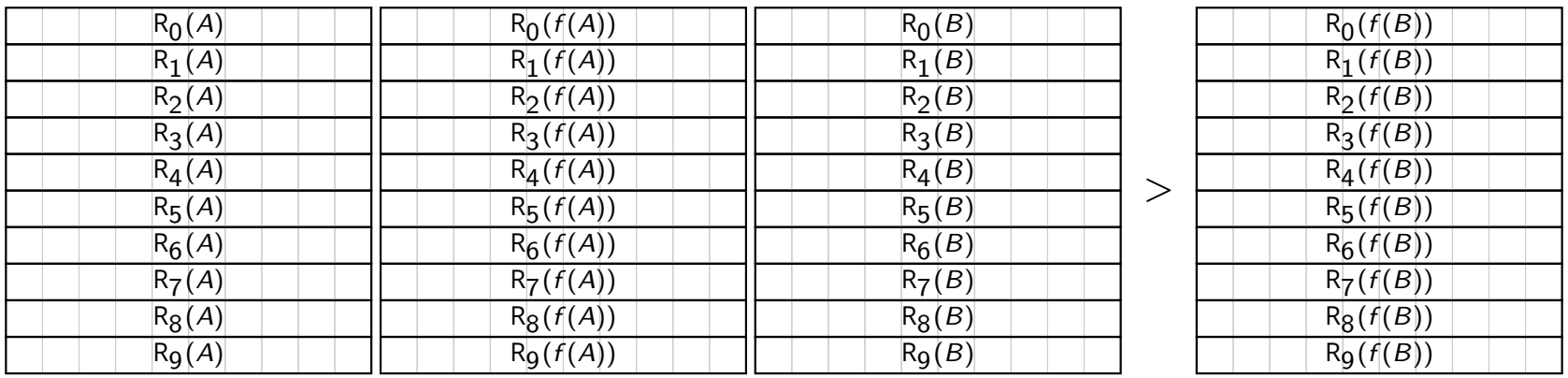
World and Culture quizzes

Experiments are done with a 37 millions parameter GPT trained from scratch.

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

World and Culture quizzes

Given a quiz composed of three 10×10 grids for the prompt and one 10×10 grid for the solution



we can represent the prompt as a sequence of 300 tokens

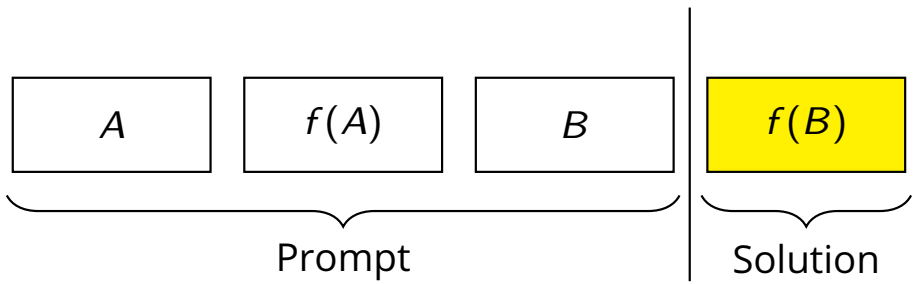
$$R_0(A), \ldots, R_9(A), R_0(f(A)), \ldots, R_9(f(A)), R_0(B), \ldots, R_9(B)$$

and the solution as a sequence of 100 tokens

$$R_0(f(B)), \ldots, R_9(f(B))$$

World and Culture quizzes

We can train a GPT on a complete sequence composed of the prompt followed by the solution, and then use it to “solve” a quiz, that it to generate the solution given the prompt.



We re-sample the training examples at every epoch, so there is no over-fitting.

The accuracy of a GPT trained on all the tasks combined gets easily above 99%.

World and Culture quizzes

Our main objective is to **generate new meaningful quizzes**, which are both outside the domain of the training examples and consistent with them.

A GPT can produce a full quiz by generating all the tokens without conditioning.

However, if the GPT is properly trained, such a quiz follows the data-distribution, and there would be no novelty beside unstructured sampling noise.

World and Culture quizzes

Our main objective is to **generate new meaningful quizzes**, which are both outside the domain of the training examples and consistent with them.

A GPT can produce a full quiz by generating all the tokens without conditioning.

However, if the GPT is properly trained, such a quiz follows the data-distribution, and there would be no novelty beside unstructured sampling noise.

We propose to train N GPTs in parallel, to generate quizzes with structured noise, and to "validate" and keep a generated quiz **only if exactly** $N - 1$ **GPTs solve it correctly.**

World and Culture quizzes

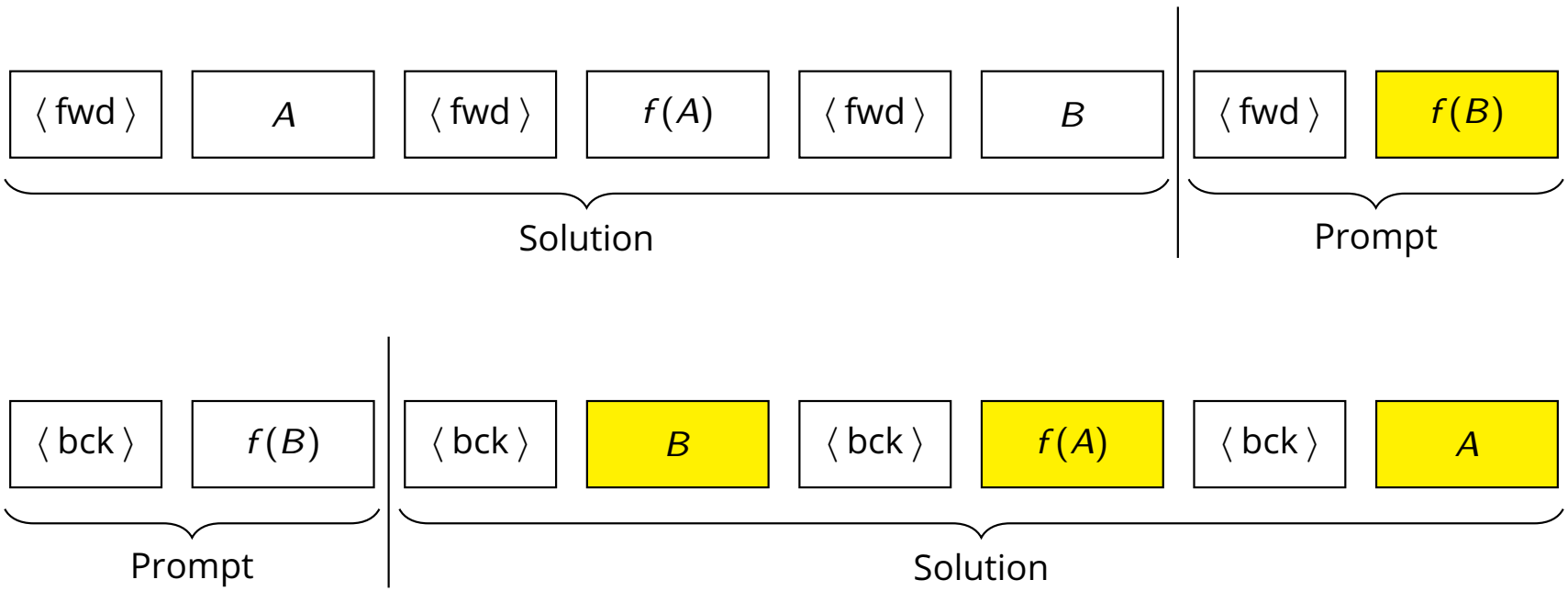
To generate quizzes with a structured noise, we

1. generate a solution at high temperature,
2. generate a consistent prompt at low temperature, and
3. re-generate the solution given this prompt at low temperature.

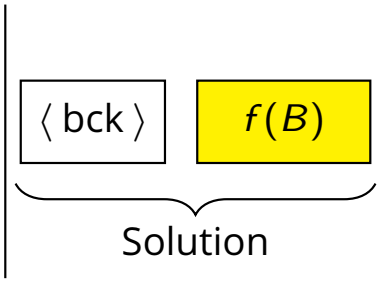
That way we get a quiz that may be slightly outside the support of the training samples, but it should be “functionally consistent.”

World and Culture quizzes

Hence we also need to generate prompts given solutions. To do that we train the model on two types of sequences, with additional tokens to indicate the direction.

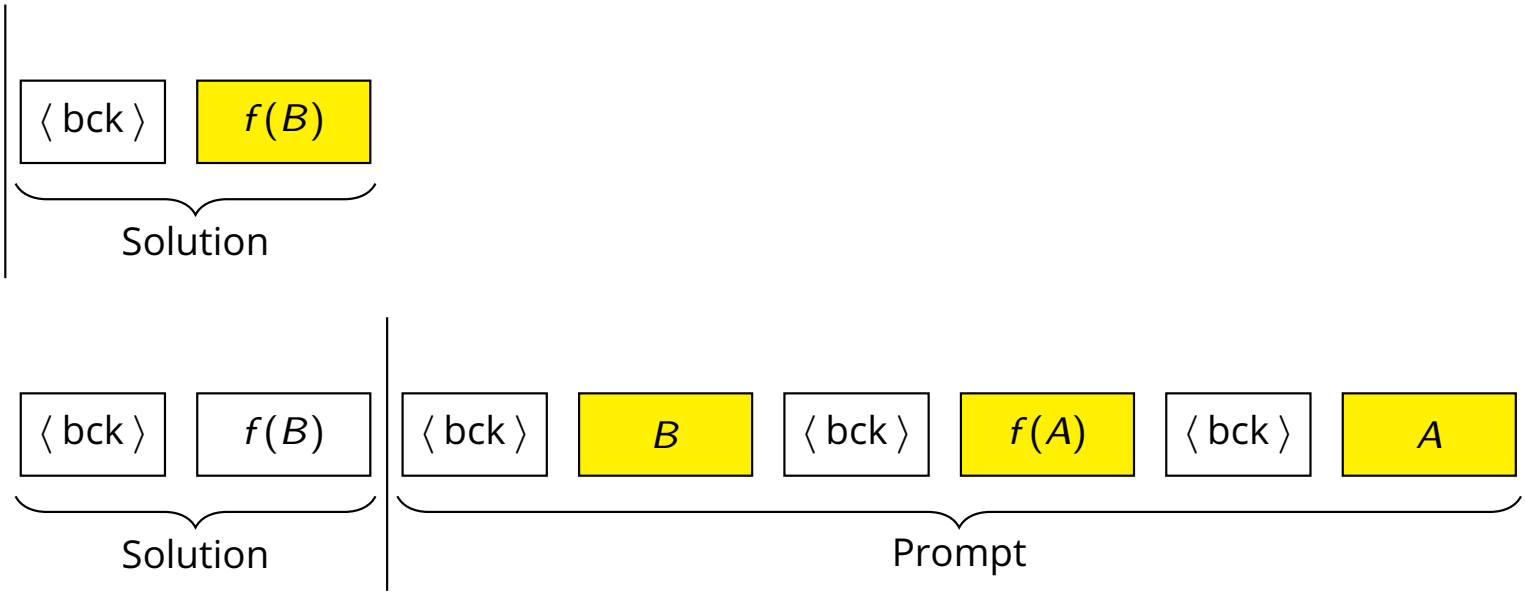


Given a model trained that way, we implement the sampling of new quizzes as follows:



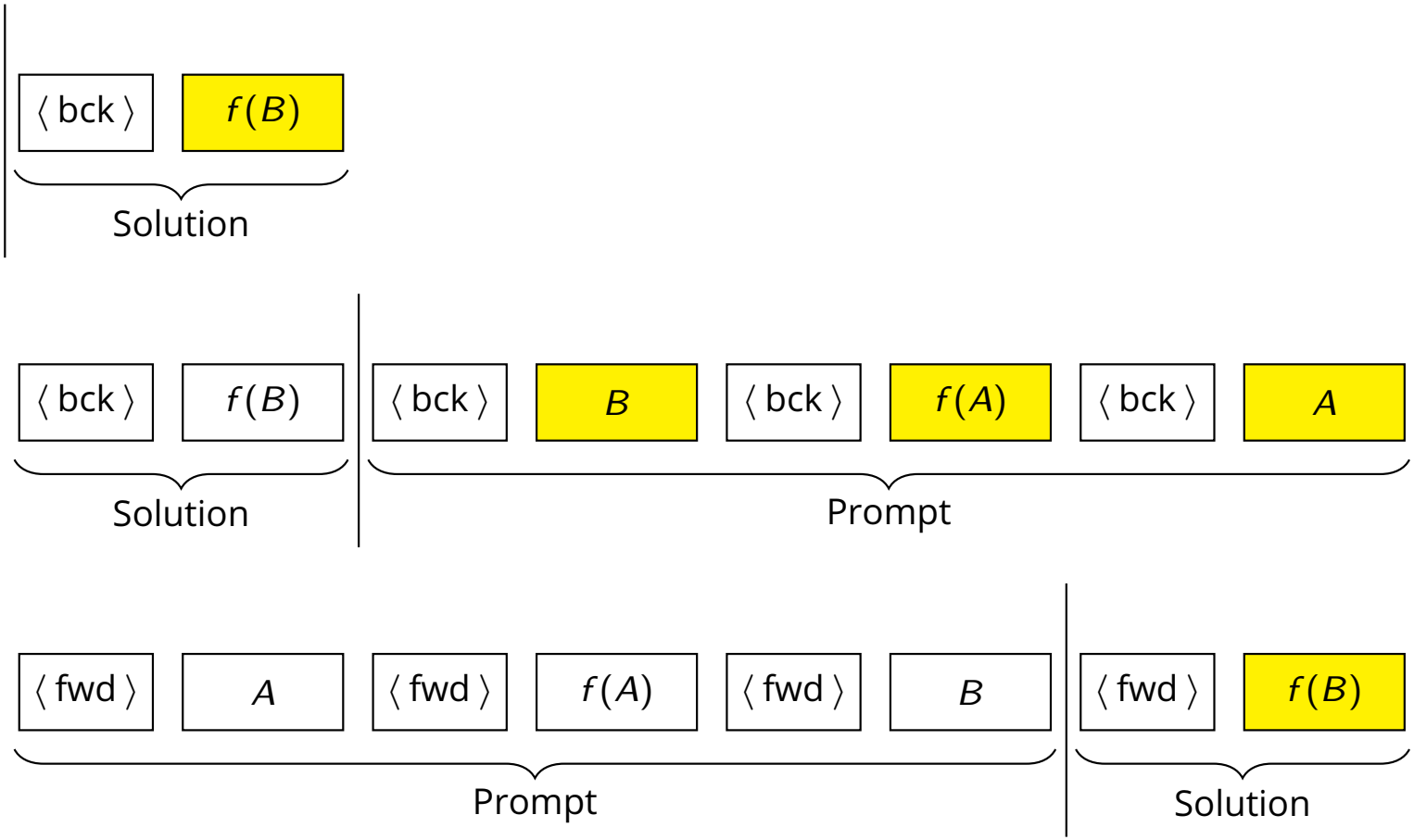
1. Generate a solution at high temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



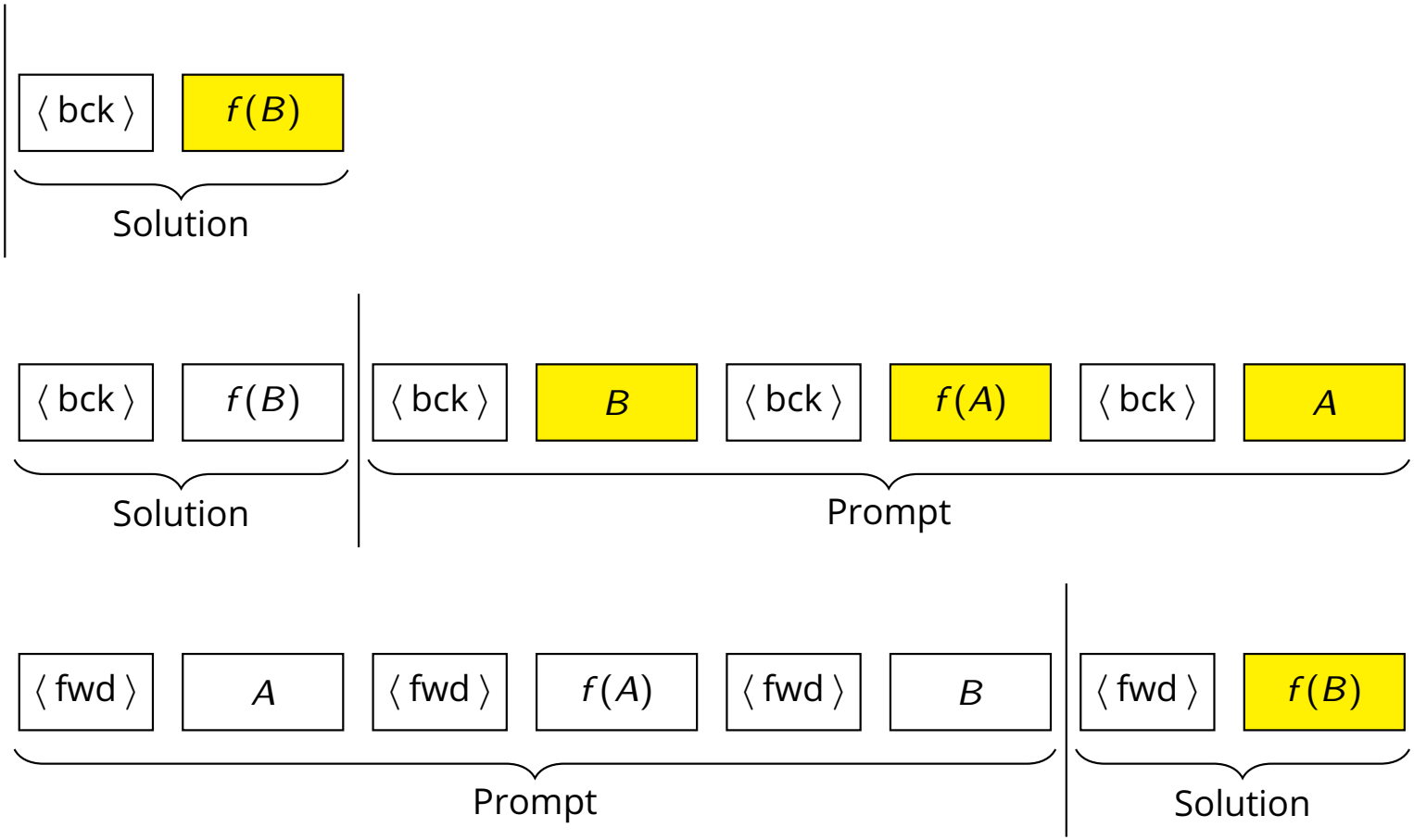
2. Generate a prompt at low temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



3. Re-generate the solution at low temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



We then evaluate the quiz with each of the N models and count the number of correct prediction M and "validate" the new quiz only if $M = N - 1$.

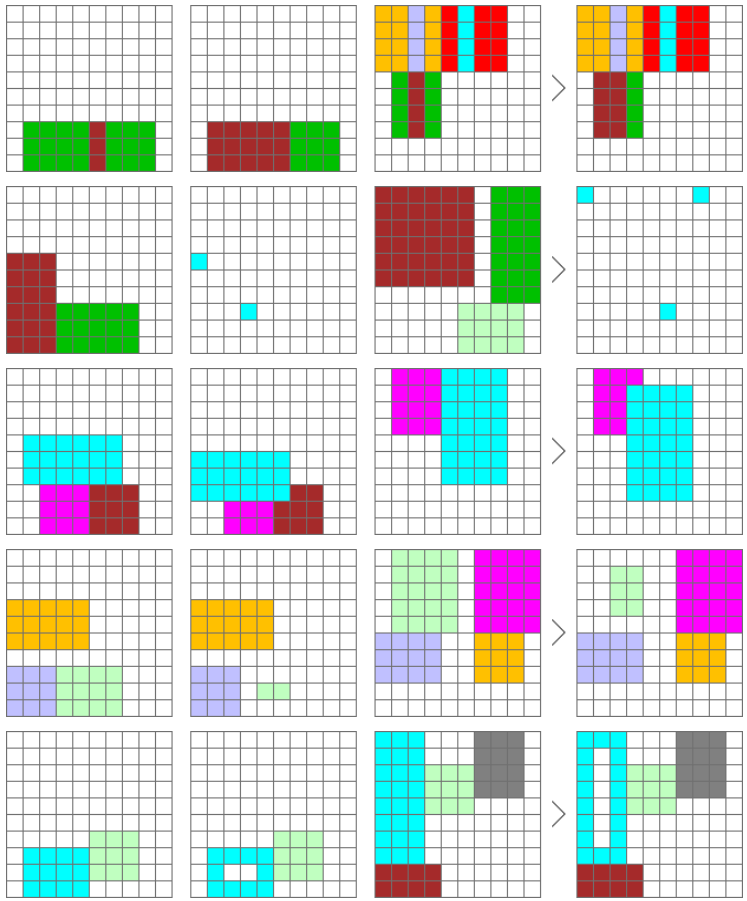
World and Culture quizzes

The global loop proceeds as follows at each iteration:

1. If all the models have an accuracy above a threshold (e.g. 0.95), generate culture quizzes and add them to the culture.

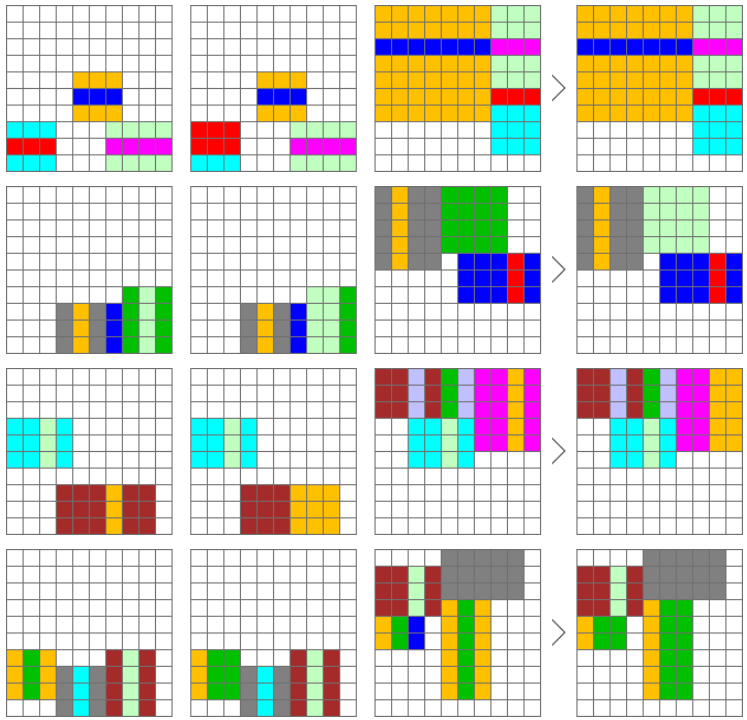
2. Select the model with the lowest accuracy, build a training set with up to 50% culture quizzes, and completed with fresh world quizzes, proceed with one epoch of training with this set.

Culture Quizzes



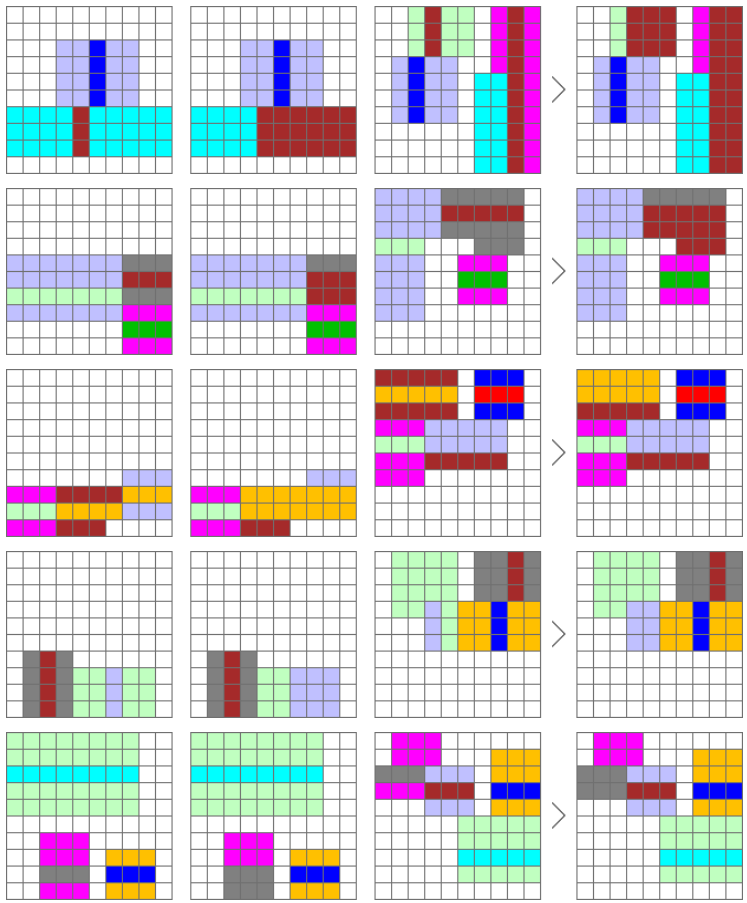
Various number of rectangles.

Culture Quizzes



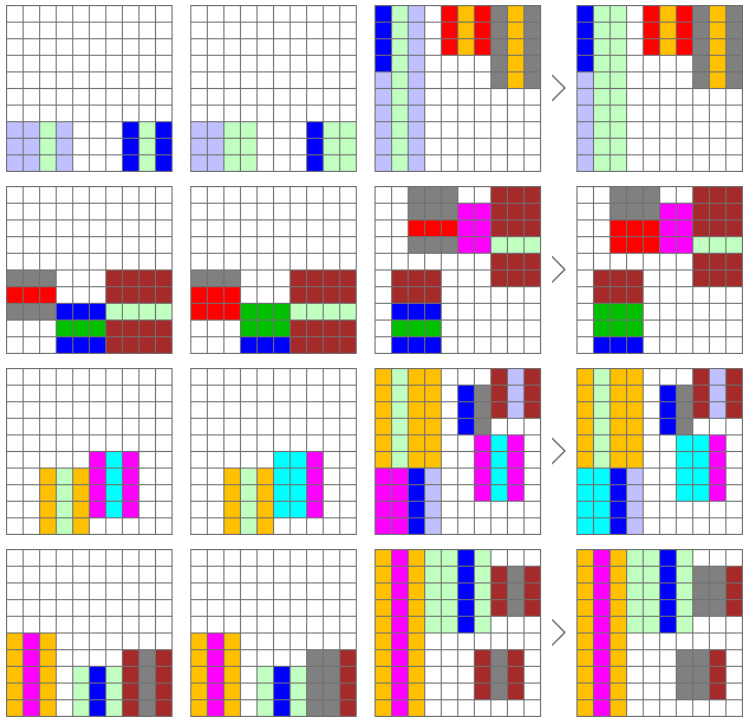
“Half Fill” with complex coloring.

Culture Quizzes



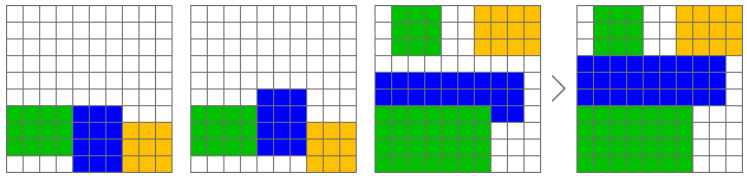
“Half Fill” with non-rectangular shapes.

Culture Quizzes



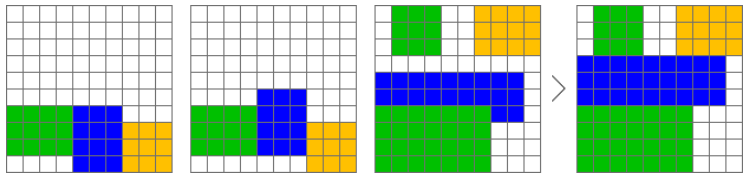
“Half Fill” with two colors or two rectangles to fill.

Culture Quizzes

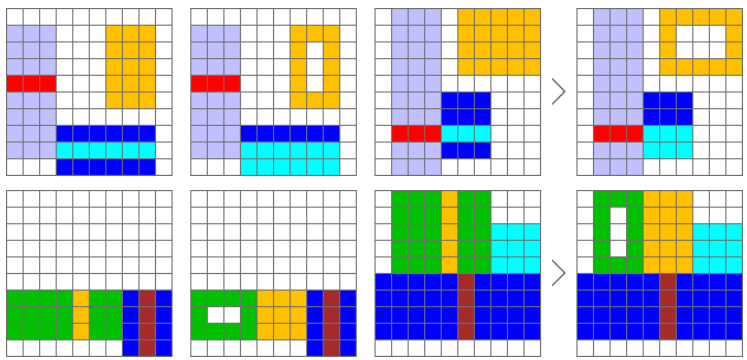


“Translate” with the moving part occluded.

Culture Quizzes

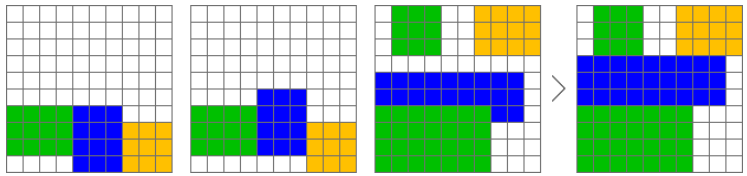


“Translate” with the moving part occluded.

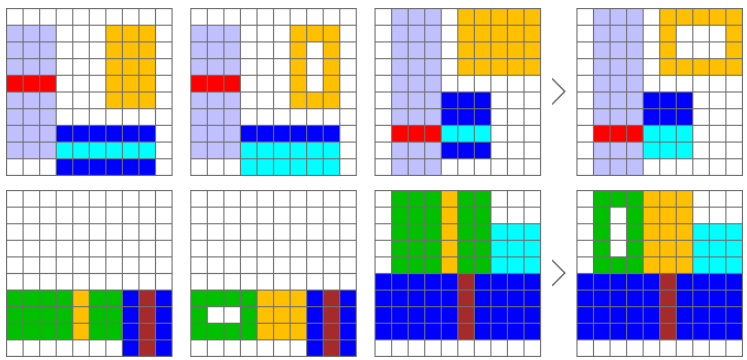


“Frame” and “Half fill” combined.

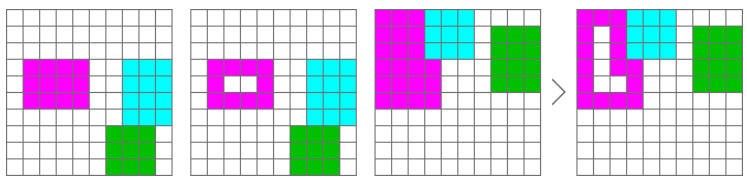
Culture Quizzes



“Translate” with the moving part occluded.

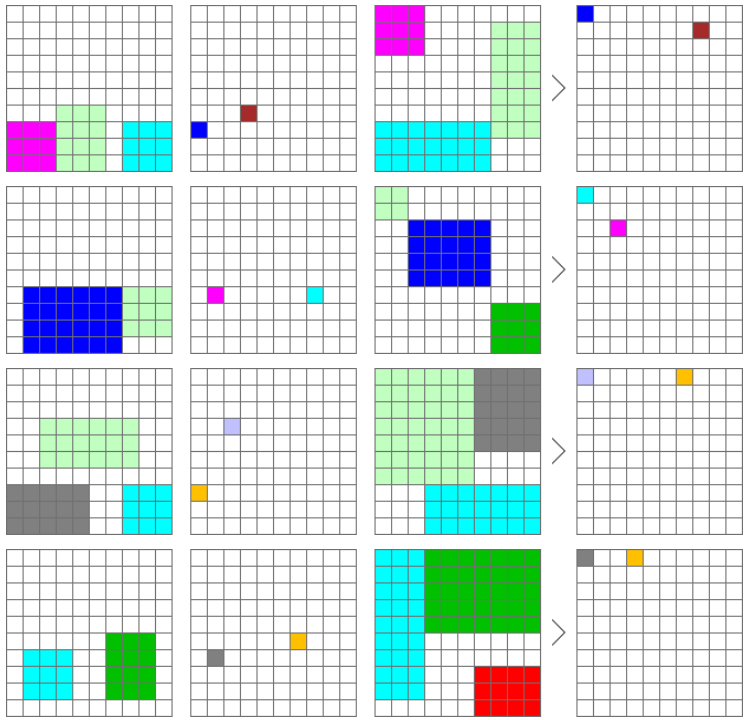


“Frame” and “Half fill” combined.



“Frame” generalized to non-rectangular shapes.

Culture Quizzes



“Detect” with location markers colored according to the rectangle colors.

Questions?

References

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. **Language models are few-shot learners**. *CoRR*, abs/2005.14165, 2020.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. **An image is worth 16x16 words: Transformers for image recognition at scale**. *CoRR*, abs/2010.11929, 2020.

K. Fukushima. **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**. *Biological Cybernetics*, 36(4):193–202, April 1980.

V. Micheli, E. Alonso, and F. Fleuret. **Transformers are sample efficient world models**. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. **Attention is all you need**. *CoRR*, abs/1706.03762, 2017.