Alternative Models

RWKV (Receptance Weighted Key Value)

ELL8299 · ELL881 · AIL861

Sourish Dasgupta
Associate Professor, DAU, Gandhinagar
https://daiict.ac.in/faculty/sourish-dasgupta

RWKV: Reinventing RNNs for the Transformer Era

Bo Peng[1,2*] Eric Alcaide[2,3,4*] Quentin Anthony[2,5*]
Alon Albalak[2,6] Samuel Arcadinho[2,7] Stella Biderman[2,8] Huanqi Cao[9] Xin Cheng[10]
Michael Chung[11] Xingjian Du[1] Matteo Grella[12] Kranthi Kiran GV[2,13] Xuzheng He[2]
Haowen Hou[14] Jiaju Lin[1] Przemysław Kazienko[15] Jan Kocoń[15] Jiaming Kong[16]
Bartłomiej Koptyra[15] Hayden Lau[2] Krishna Sri Ipsit Mantri[17] Ferdinand Mom[18,19]
Atsushi Saito[2,20] Guangyu Song[21] Xiangru Tang[22] Bolun Wang[23] Johan S. Wind[24]
Stanisław Woźniak[15] Ruichong Zhang[9] Zhenyuan Zhang[2] Qihang Zhao[25,26]
Peng Zhou[23] Qinghua Zhou[5] Jian Zhu[27] Rui-Jie Zhu[28,29]

[1]Generative AI Commons [2]EleutherAI [3]U. of Barcelona [4]Charm Therapeutics [5]Ohio State U. [6]U. of C., Santa Barbara
[7]Zendesk [8]Booz Allen Hamilton [9]Tsinghua University [10]Peking University [11]Storyteller.io [12]Crisis24 [13]New York U.
[14]National U. of Singapore [15]Wroclaw U. of Science and Technology [16]Databaker Technology [17]Purdue U. [18]Criteo AI Lab
[19]Epita [20]Nextremer [21]Moves [22]Yale U. [23]RuoxinTech [24]U. of Oslo [25]U. of Science and Technology of China
[26]Kuaishou Technology [27]U. of British Columbia [28]U. of C., Santa Cruz [29]U. of Electronic Science and Technology of China

Findings, EMNLP, 2023

Modeling Language over Time

Example: "John loves Mary who lives in New York City."

$$P(x_t \mid x_{1:t-1})$$

— the probability of the next token given all previous ones.

In our stream:

$$[\text{John, loves, Mary, who, lives, in, New, York, City}]$$

a good model must keep the referent *Mary* active across the relative clause "who lives. . . ", while letting older context fade gracefully.

n-gram LM: n-Order Markov Chains

$$P(x_t \mid x_{1:t-1}) \approx P(x_t \mid x_{t-n:t-1})$$

Limitation. Fixed window, no hidden structure, weak generalization beyond seen sequences.

Hidden Markov Models (HMM) as LMs

$$P(x_{1:T}) = \sum_{z_{1:T}} P(z_1) \prod_{t=2}^{T} P(z_t \mid z_{t-1}) \, P(x_t \mid z_t).$$

$$P(x_t \mid x_{1:t-1}) = \sum_{z_t} P(x_t \mid z_t) \, P(z_t \mid x_{1:t-1}).$$

Learning Memory Horizon via LSTM

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1}) && \text{(forget)} \\
i_t &= \sigma(W_i x_t + U_i h_{t-1}) && \text{(input)} \\
o_t &= \sigma(W_o x_t + U_o h_{t-1}) && \text{(output)} \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \qquad h_t = o_t \odot \tanh(c_t)
\end{aligned}
$$

Idea. The gates learn what to retain or overwrite — a learned Markov drift.

The Markovian Drift in LSTM

Example: "John loves Mary who lives in New York City."

Forget gate f_t implements exponential decay of older content; i_t injects new info. Effective order n becomes adaptive.

On our example. At "who", hidden state still carries "Mary" through the clause — if f_t keeps it.

Limitations. Sequential dependency, implicit retrieval, and long-range credit assignment remain challenging.

Self-Attention in Transformers – Generalizing skip-gram LM

$$q_t = W_q x_t, \qquad k_i = W_k x_i, \qquad v_i = W_v x_i,$$

$$\omega_{t,i} = \frac{e^{q_t^\top k_i}}{\sum_{j \le t} e^{q_t^\top k_j}}, \qquad z_t = \sum_{i \le t} \omega_{t,i} v_i.$$

Example: "John loves Mary who lives in New York City."

Our sentence. "who" attends to "Mary"; "City" aggregates "New", "York", "in", "lives".

$O(T^2)$ cost, no built-in recency prior, heavy KV-cache.

RWKV: Bridging RNN with Transformers

RWKV: Bridging RNN Efficiency with Attention Quality

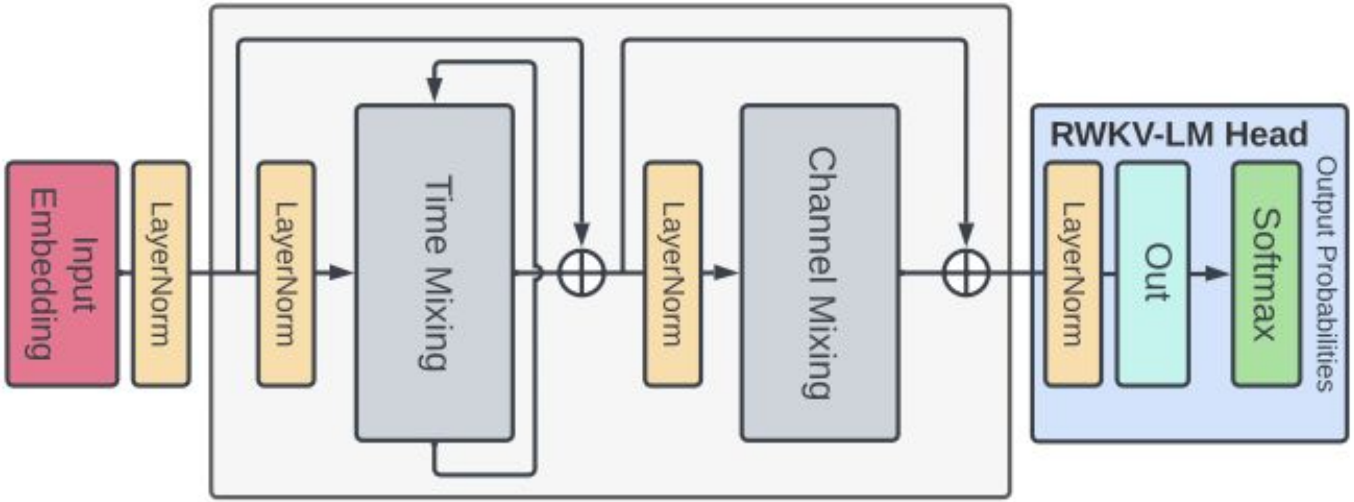
Goal:

- Global content mixing (like Transformers),
- Linear-time recurrent updates (like RNNs),

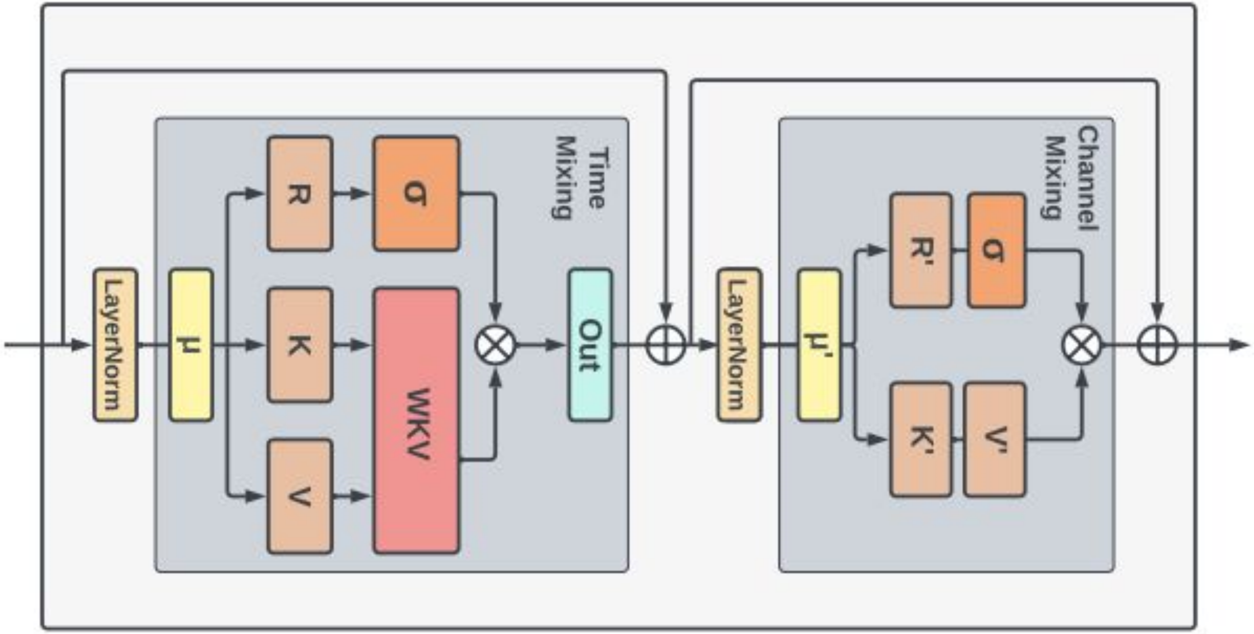
Interpretation. Content strength (β) $\times$ distance decay (α^{t-i}) $=$ soft, exponential skip-gram kernel.

$$s_t = \sum_{i \le t} \frac{\beta_i\, \alpha^{t-i}}{\sum_{j \le t} \beta_j\, \alpha^{t-j}} v_i, \quad \alpha = e^{-w} \in (0,1), \quad \beta_i = e^{k_i}.$$

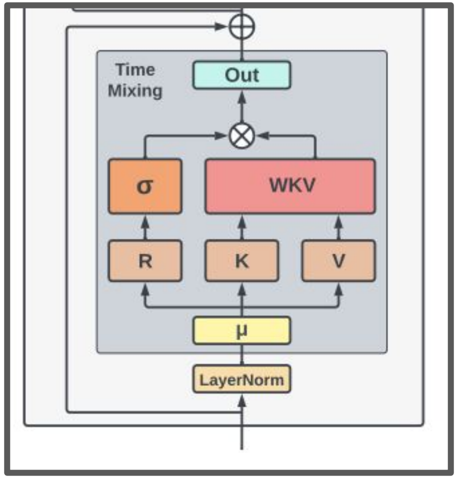
RWKV Architecture



RWKV Encoder



Step 1: Time-Mixing



$$x_t^{(k)} = \text{mix}_k \odot x_t + (1 - \text{mix}_k) \odot x_{t-1},$$
$$x_t^{(v)} = \text{mix}_v \odot x_t + (1 - \text{mix}_v) \odot x_{t-1},$$
$$x_t^{(r)} = \text{mix}_r \odot x_t + (1 - \text{mix}_r) \odot x_{t-1}.$$

Parallelization: x_{t-1} comes from a **vectorized shift** — a previous-index lookup across the whole batch (no sequential loop).

Time-Mixing via Token-Shift

Example: "John loves Mary who lives in New York City."

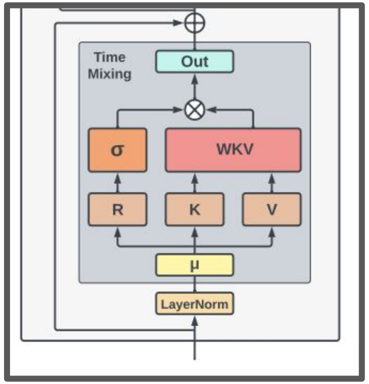
For [John, loves, Mary, who, ...], at $t = 3$ ("Mary"):

$$x_3^{(k)} = \mathrm{mix}_k \odot x_{\mathrm{Mary}} + (1 - \mathrm{mix}_k) \odot x_{\mathrm{loves}}.$$

Local bigram blending (a smooth contextual prior), done in parallel via:

$$X_{\mathrm{shift}}[t] = X[t-1], \quad X_{\mathrm{shift}}[1] = 0.$$

Step 2: Generating Key, Value, and *Receptance*



$$k_t = W_k\, x_t^{(k)}, \qquad v_t = W_v\, x_t^{(v)}$$

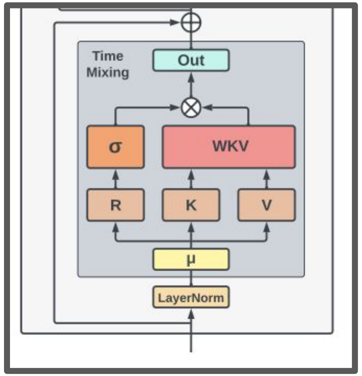
$$r_t = W_r\, x_t^{(r)}.$$

Gates. $\alpha = e^{-w},\ \beta_t = e^{k_t}.$

Step 3: Update the current State (*WKV Operation*)

Recurrence.

$$a_t = \alpha a_{t-1} + \beta_t v_t, \qquad b_t = \alpha b_{t-1} + \beta_t.$$



$$a_t = \sum_{i \leq t} \alpha^{t-i} \beta_i v_i, \qquad b_t = \sum_{i \leq t} \alpha^{t-i} \beta_i.$$

$$s_t = \frac{a_t}{b_t} = \sum_{i \leq t} \underbrace{\frac{\beta_i \alpha^{t-i}}{\sum_j \beta_j \alpha^{t-j}}}_{\omega_{t,i}} v_i, \quad \omega_{t,i} \geq 0, \ \sum_i \omega_{t,i} = 1.$$

Self-Attention vs. Exponential Decay Prior

$$\text{Transformer: } \omega_{t,i} \propto e^{q_t^\top k_i}$$

$$\text{RWKV: } \omega_{t,i} \propto e^{k_i}\, e^{-w(t-i)}$$

$\omega_{t,i} \propto \beta_i \alpha^{t-i}$ acts like a *decayed co-occurrence* weight. Nearby tokens contribute strongly; distant ones fade unless reinforced. Skip-gram uses fixed windows; RWKV uses a soft, infinite exponential window.

The inherent Markovian Drift

$$s_t = \frac{a_t}{b_t} = \sum_{i \le t} \underbrace{\frac{\beta_i \alpha^{t-i}}{\sum_j \beta_j \alpha^{t-j}}}_{\omega_{t,i}} v_i, \quad \omega_{t,i} \ge 0, \ \sum_i \omega_{t,i} = 1.$$

Example: "John loves Mary who lives in New York City."

With $0 < \alpha < 1$, older info decays exponentially; the center of mass $s_t = a_t / b_t$ *drifts* toward recent high-β tokens.

Our example. At "who", influence peaks at "Mary"; earlier "John" fades unless refreshed. As "lives in New York City" arrives, mass shifts toward the predicate phrase.

Step 4: What exactly is the Receptance for? (RWKV Operation)

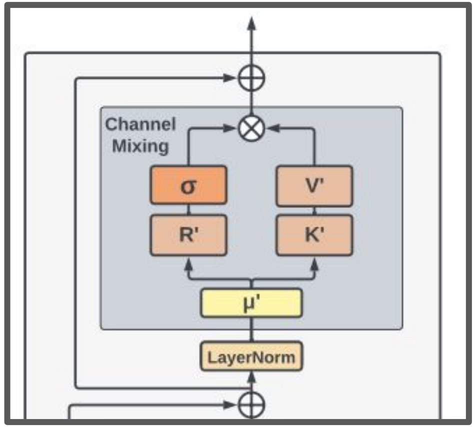
$$\boxed{r_t = W_r\, x_t^{(r)}\,.}$$

Gated Readout and Channel-Wise Receptivity

$$\gamma_t = \sigma(r_t), \qquad y_t = W_o(\underline{\gamma_t \odot \mathrm{LN}(s_t)})\,.$$

$\gamma_t \in (0,1)^d$ controls how much of the contextual mixture s_t is *accepted* into the output. Unlike the LSTM's output gate, it filters the normalized state s_t — hence "receptance".

Step 5: Channel Mixing: *Alternative to FFN between Transformer Layers*



$$y'_t = x_t + y_t.$$

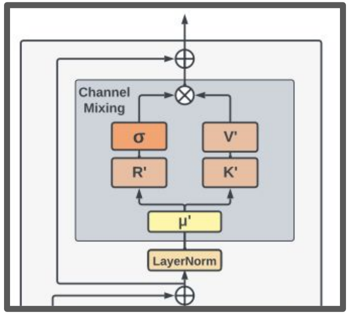
residual

$$x_t^{(f)} = \mathrm{mix}_f \odot y'_t + (1 - \mathrm{mix}_f) \odot y'_{t-1},$$

$$x_t^{(g)} = \mathrm{mix}_g \odot y'_t + (1 - \mathrm{mix}_g) \odot y'_{t-1},$$

$$u_t = W_f x_t^{(f)},$$

Normalized (squashed via ReLU²) & Gated Output



$$r'_t = \sigma(W_g x_t^{(g)}),$$

$$z_t = W_p(r'_t \odot \phi(u_t)).$$

$$\phi(u_t) = \max(u_t, 0)^2$$

Vectorized shift. y'_{t-1} is a tensor offset over the entire Y' (parallel in training).

The Final Bit – RKWV is stacked just like Transformers

$$y_t'' = y_t' + z_t, \qquad x_t^{(\ell+1)} = y_t''^{(\ell)}.$$

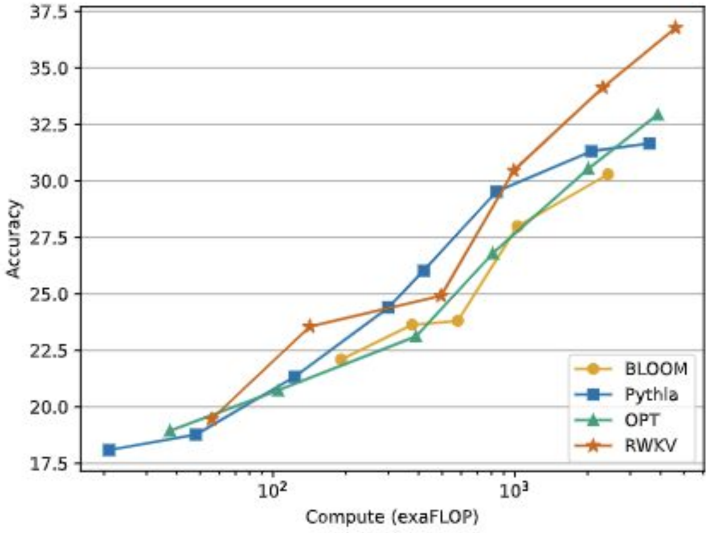
$$X \xrightarrow{\text{Time-Mix+WKV}} Y' \xrightarrow{\text{Channel-Mix}} Z \Rightarrow Y'' = Y' + Z.$$

Stack multiple RWKV blocks; LM head on top.

RWKV Evaluation Benchmark

ARC (AI2 Reasoning Challenge): Multiple-choice science questions requiring factual recall and reasoning.

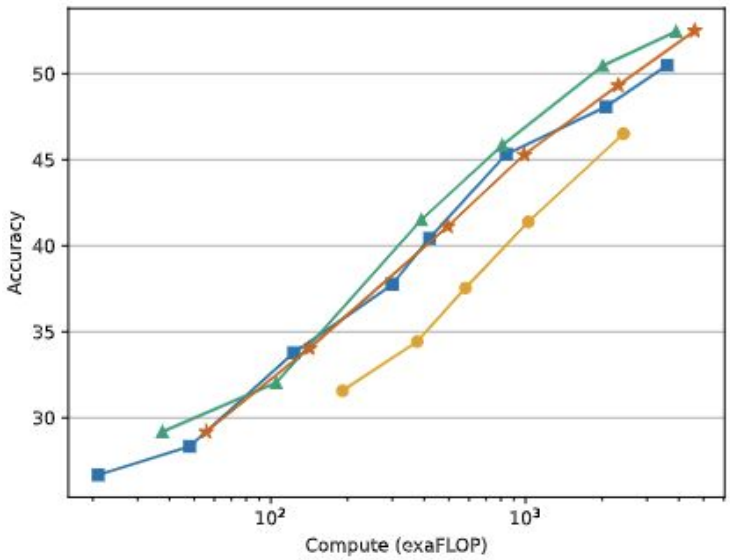
$\Rightarrow$ Tests general logical reasoning over knowledge.



(a) ARC (Challenge)

- **FLOP** = one floating-point operation (a single arithmetic step).
- **ExaFLOP** = 10^{18} such operations — a measure of total training work.
- **Compute (exaFLOPs)** = model size $\times$ dataset size $\times$ training steps.

HellaSwag: Pick the most plausible continuation of a short story.
$\Rightarrow$ Measures commonsense reasoning and event plausibility.

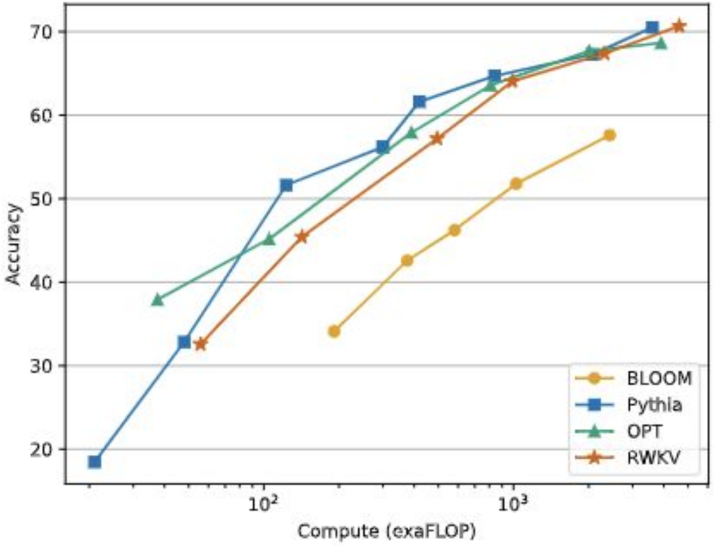


(b) HellaSwag

LAMBADA:

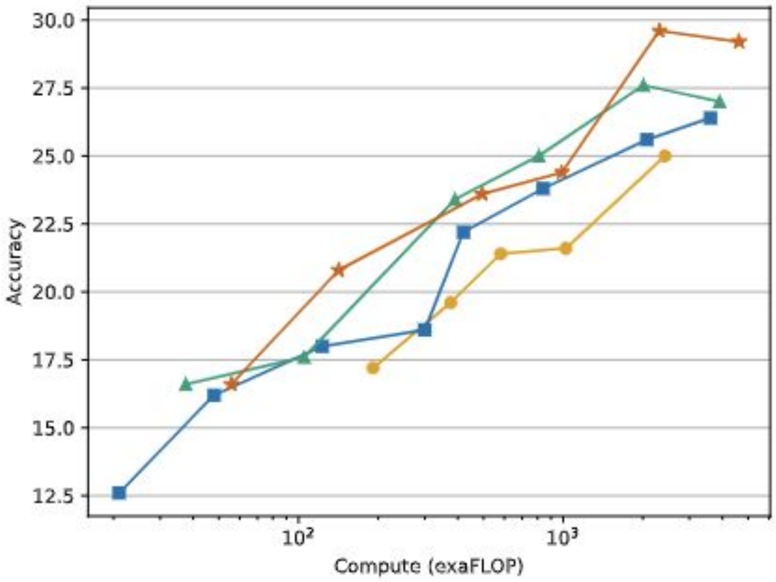
Predict the final word of a passage given full preceding context.

$\Rightarrow$ Pure long-range dependency test; requires global coherence.



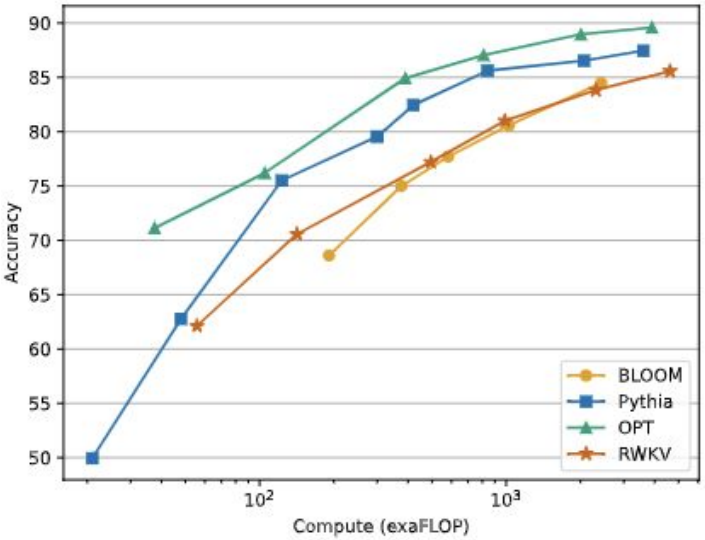
(c) LAMBADA (OpenAI)

OpenBookQA: small science reasoning tasks requiring commonsense and facts from an "open book" of basic science principles.



(d) OpenBookQA

ReCoRD (Reading Comprehension with Commonsense Reasoning):
Cloze-style question answering: fill in masked entities based on passage context.
$\Rightarrow$ Evaluates deep reading and entity tracking (*Mary who*).



(e) ReCoRD

Long Range Arena (LRA):

Synthetic + real datasets (text, math, images) with sequences up to 16K tokens.

$\Rightarrow$ Evaluates scalability of long-context modeling beyond 1K tokens.

Task	Concrete Analogy / Example
ListOps	Solving nested math expressions: `[MAX 2 9 [MIN 4 7] 3]` $\rightarrow$ 9.
Text	Reading a long article and deciding if it's about politics or science.
Retrieval	Finding where "Mary" first appears in a 2,000-word passage.
Image	Reading an image row-by-row like a sequence of pixel words.
Pathfinder	Checking if two dots in a maze are connected by a winding path.
Path-X	Navigating a huge, tangled map — extreme long-range reasoning.

Long Range Arena (LRA):

Synthetic + real datasets (text, math, images) with sequences up to 16K tokens.

$\Rightarrow$ Evaluates scalability of long-context modeling beyond 1K tokens.

MODEL	LISTOPS	TEXT	RETRIEVAL	IMAGE	PATHFINDER	PATH-X	AVG
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
Hrrformer	39.98	65.38	76.15	50.45	72.17	✗	60.83
S4	**59.60**	**86.82**	**90.90**	**88.65**	**94.20**	**96.35**	**86.09**
RWKV	55.88	86.04	88.34	70.53	58.42	✗	72.07

RWKV vs. LLMs

Tasks: Reasoning & Affective Understanding

Task	Type	What It Tests	Example (Simplified)
RTE	Textual entailment	Logical inference between two statements	*Premise:* John loves Mary. *Hypothesis:* John cares about Mary. (True/False)
WNLI	Coreference resolution	Understanding pronoun reference	*The trophy doesn't fit into the suitcase* because *it* is too small. → "it" = suitcase.
GoEmotions	Emotion classification	Detecting expressed emotions	*I can't believe this happened!* → surprise/anger.
PolEmo2	Aspect-based sentiment	Tracking multi-aspect polarity	*The food was delicious, but the service was slow.* → positive (food), negative (service).

Observation. These tasks require fine-grained reasoning — identifying subtle relations, emotional tone, and entity references — challenging for models with compressed or lossy memory.

RWKV-GPT = the *RWKV model family* (e.g., RWKV-4 Raven 14B) evaluated under **ChatGPT-style prompting**. No extra fine-tuning — just tested **zero-shot** with instruction-like inputs.

RWKV-adapted = RWKV that has been lightly **fine-tuned or instruction-adapted** on conversational and reasoning data (e.g., Alpaca, ShareGPT).

This alignment step teaches it how to interpret instructions and follow task structures.

Task Name	Measure	ChatGPT	GPT-4	RWKV-GPT	RWKV-adapted	SOTA
RTE	F1 Macro	88.1	**91.3**	44.2	74.8	92.1
WNLI	Accuracy	81.7	**91.6**	47.9	49.3	97.9
GoEmotions	F1 Macro	**25.6**	23.1	7.9	7.9	52.8
PolEmo2	F1 Macro	**44.1**	41.0	38.2	40.9	76.4

Table 6: ChatGPT, GPT-4 and RWKV-4-Raven-14B reasoning performance comparison in RTE (Wang et al., 2019), WNLI (Wang et al., 2018), GoEmotions (Demszky et al., 2020), and PolEmo2 (Kocoń et al., 2019) benchmarks. RWKV GPT prompts were primarily used for ChatGPT in (Kocoń et al., 2023). SOTA is provided as a supplementary reference.

Is there a fundamental limitation?

Example: "John loves Mary who lives in New York City."

RWKV keeps only two running summaries of all history:

$$a_t = \alpha a_{t-1} + \beta_t v_t, \quad b_t = \alpha b_{t-1} + \beta_t, \quad s_t = \frac{a_t}{b_t}.$$

Every past token—"John loves Mary who lives in New York City"—gets compressed into this single vector s_t.

Our running example at token "in":

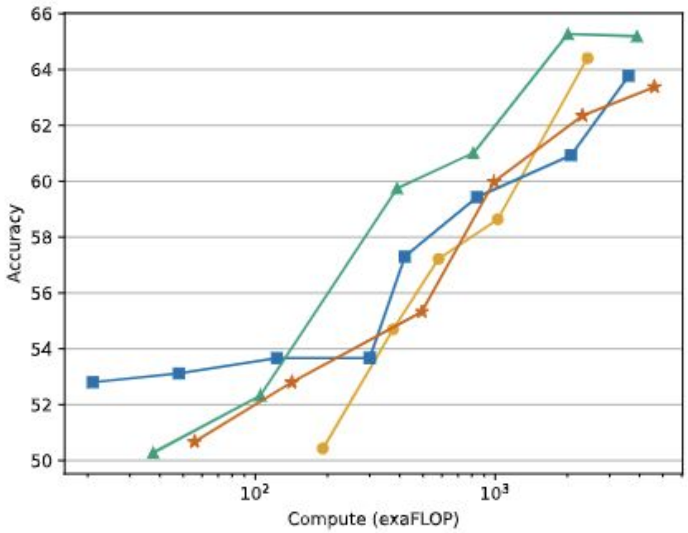
Word	Relative age	Weight (β)	Influence on current memory
John	very old (α^5)	small	nearly forgotten
[illegible]	[illegible]	[illegible]	[illegible] remains
[illegible]	[illegible]	[illegible]	[illegible] t
who	recent ($\alpha^{[illegible]}$)	moderate	syntactic link
lives	fresh (α^1)	high	main predicate focus
in	current (α^0)	strong	dominates state update

Information Bottleneck

Winogrande:

Pronoun resolution task—select which noun a pronoun refers to.

$\Rightarrow$ Tests fine-grained coreference and pragmatic reasoning.



(f) Winogrande

What else might be going wrong …

RWKV Updates Memory in *Discrete* Steps

$$a_t = \alpha a_{t-1} + \beta_t v_t.$$

- RWKV is elegant and efficient—at each step it decides how much of the past to retain (α) and how much new content to add ($\beta_t v_t$).

BUT, language meaning often *flows* between tokens rather than "jumping" at each step.

Language structure does not abruptly change

Example: "John loves Mary who lives in New York City."

"Mary" must smoothly shift from **object** of "loves" to **subject** of "lives."

- Memory should *evolve continuously*, not just be replaced at token boundaries.

So why study RWKV now?

The world is not just Transformers!

- Transformers dominate LLMs, but their vanilla versions scale quadratically with sequence length.
- RWKV offers a **linear-time alternative** — same expressiveness, far lighter compute.
- This matters for on-device, streaming, and long-context applications.

Bridging the world of recurrence with attention

- RWKV fuses RNN-style recurrence (memory efficiency) with Transformer-like attention (global context).
- It shows how **sequence models can be both causal and parallelizable** — a rare balance.
- Understanding RWKV builds intuition for next-generation architectures like **Mamba, RetNet, and Hyena**.

Opening Research Directions

- RWKV pushes the boundary on **memory dynamics, parallelization, and efficiency**.
- Studying it deepens understanding of **state-space models, kernel methods, and exponential recency**.
- Its principles now appear across cutting-edge research — from **selective SSMs** to **efficient LLM distillation**.

Questions?