NeurIPS 2025 Best Papers

Markov Lab Seminar
17 Dec 2025

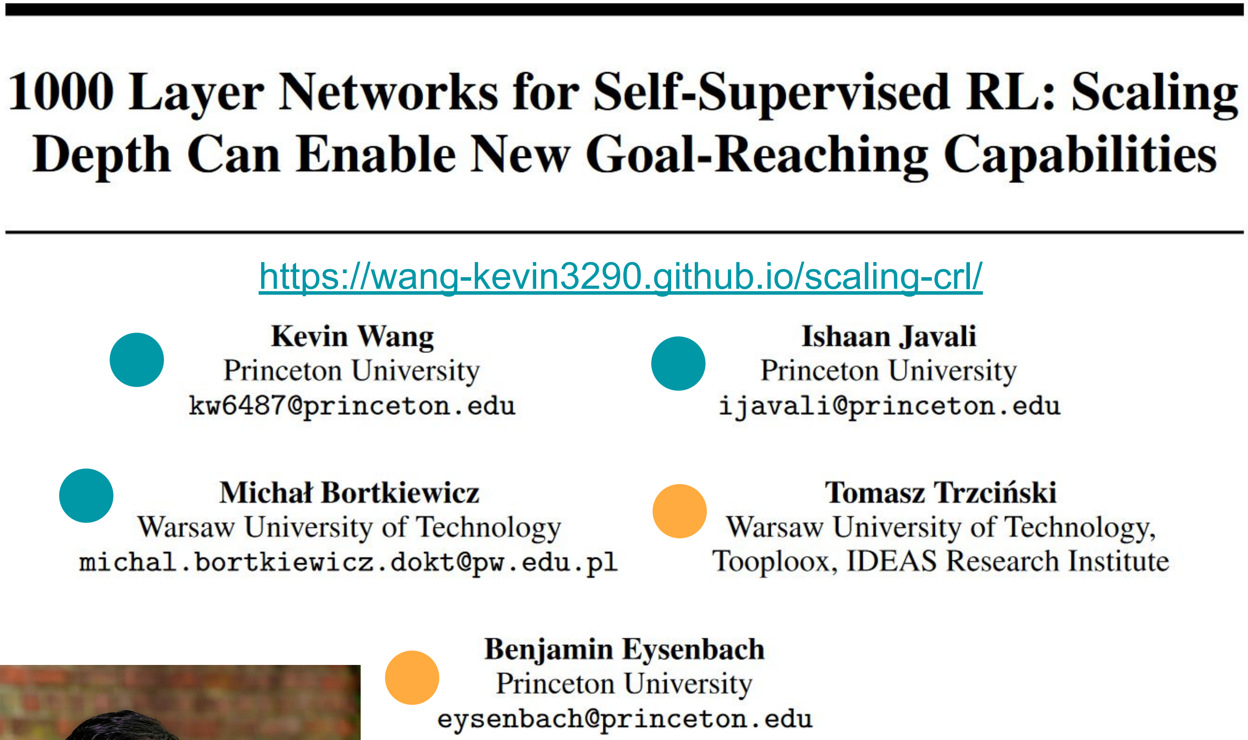
Санкт-Петербургский государственный университет
1724

M L

Факультет математики и компьютерных наук
• СПбГУ •

1000 Layer Networks for Self-Supervised RL: Scaling Depth Can Enable New Goal-Reaching Capabilities

Presenter: Konstantin Yakovlev



1000 Layer Networks for Self-Supervised RL: Scaling Depth Can Enable New Goal-Reaching Capabilities

https://wang-kevin3290.github.io/scaling-crl/

🔵 **Kevin Wang**
Princeton University
kw6487@princeton.edu

🔵 **Ishaan Javali**
Princeton University
ijavali@princeton.edu

🔵 **Michał Bortkiewicz**
Warsaw University of Technology
michal.bortkiewicz.dokt@pw.edu.pl

🟠 **Tomasz Trzciński**
Warsaw University of Technology,
Tooploox, IDEAS Research Institute

🟠 **Benjamin Eysenbach**
Princeton University
eysenbach@princeton.edu

Head of Princeton RL lab
Assistant Professor

Did PhD with S.Levine,
R.Salakhutdinov

🔵 BSc/MSc/PhD Student

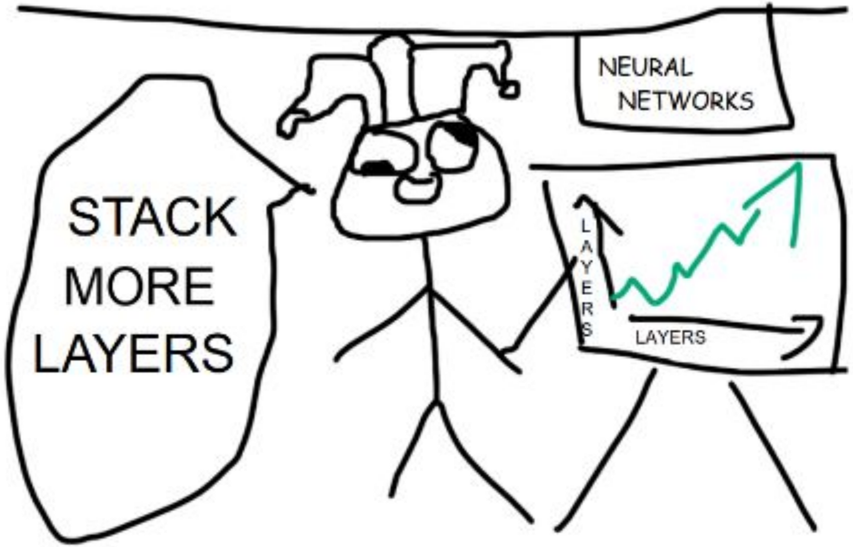
🟠 Senior Researcher

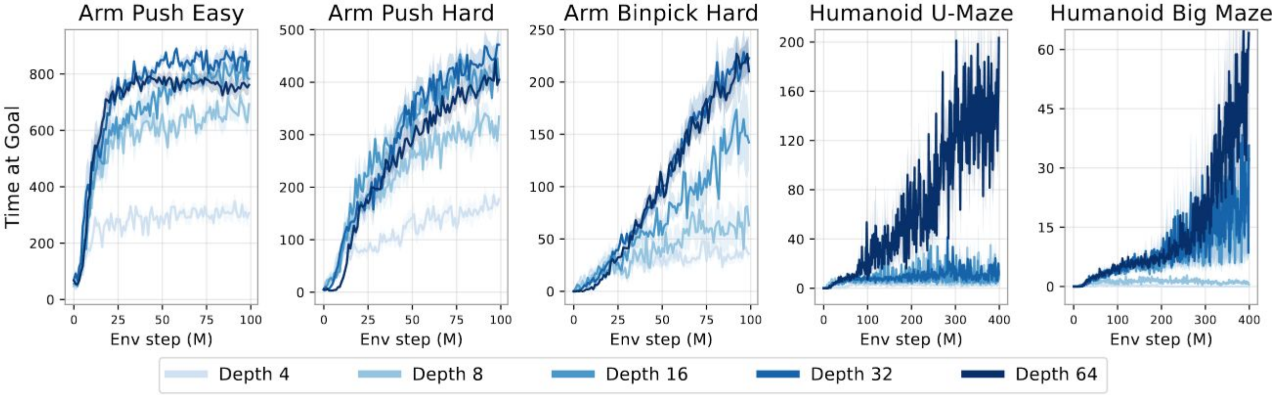
Full Professor, DSc,
h-index 43

CV guy, co-author of
BRIEF

TLDR

- Stack more layers
- Within a special type of RL methodology
  - Called Contrastive RL
- And you will get significant boost in performance





Literally, that's it

- No novel method/algorithm/model is proposed
- No novel challenging problem is stated
- No novel dataset/benchmark is proposed
- No theoretical analysis is conducted
- This is a pure empirical paper
    - Surprisingly the paper is not only accepted to NeurIPS as oral but is also chosen as a best paper (1 out 4 (+3 runner-ups))

1.Intro ~ 1 page

2. Related Work <1 page

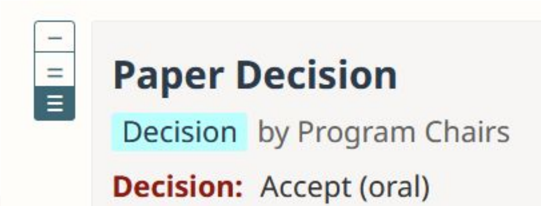
3. Preliminaries ~ 1 page

4. Experiments > 5 pages (+7 pages Appendix)

5. Conclusion ~ 0.5 page

Strengths

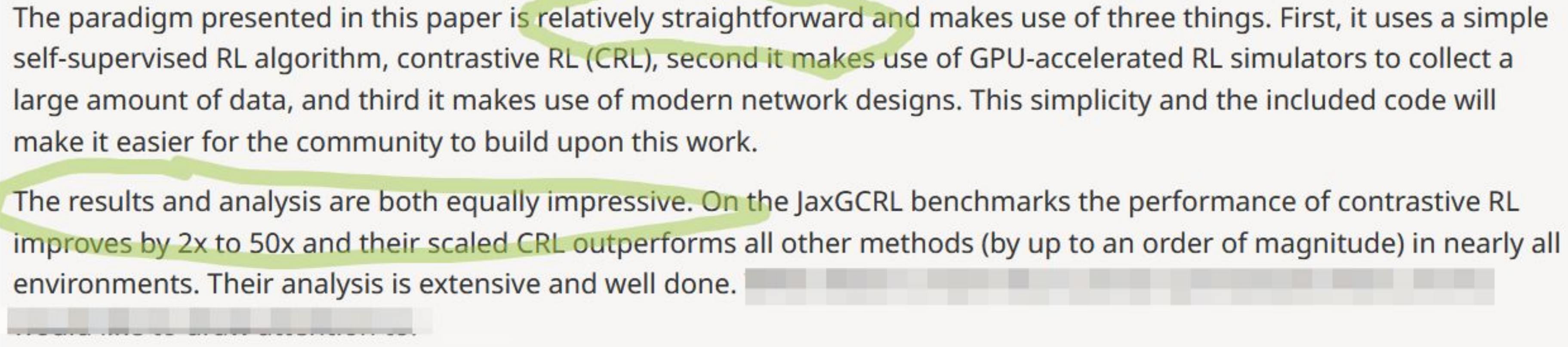
- The empirical analysis is really thorough
- Great results (up to x50 speed up)
    - Even if this only applies to certain setups and RL methodologies
- Simplicity + Code =>
  the community can build up on the work



Paper Decision

Decision by Program Chairs

Decision: Accept (oral)



The paradigm presented in this paper is relatively straightforward and makes use of three things. First, it uses a simple self-supervised RL algorithm, contrastive RL (CRL), second it makes use of GPU-accelerated RL simulators to collect a large amount of data, and third it makes use of modern network designs. This simplicity and the included code will make it easier for the community to build upon this work.

The results and analysis are both equally impressive. On the JaxGCRL benchmarks the performance of contrastive RL improves by 2x to 50x and their scaled CRL outperforms all other methods (by up to an order of magnitude) in nearly all environments. Their analysis is extensive and well done. [redacted]

What this paper is about

distribution of goals

- **Goal-conditioned** reinforcement learning

$$(\mathcal{S}, \mathcal{A}, p_0, p, p_g, r_g, \gamma),$$

$$r_g(s_t, a_t) \triangleq (1-\gamma)p(s_{t+1} = g \mid s_t, a_t)$$

Reward ~ (discounted) probability of reaching the goal

$$\pi(a \mid s, g)$$

the policy is conditioned on the goal

- **Contrastive** RL

RL ~ classification
whether current states and actions belong to the same or different trajectory.

Eysenbach, B., Zhang, T., Levine, S. and Salakhutdinov, R.R., Contrastive learning as goal-conditioned reinforcement learning. *NeurIPS 2022*

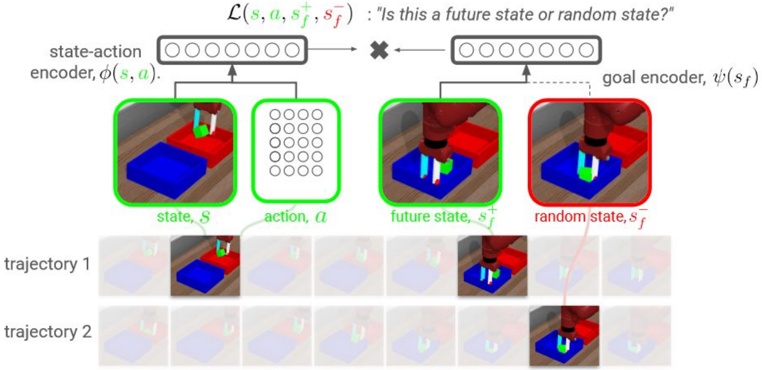
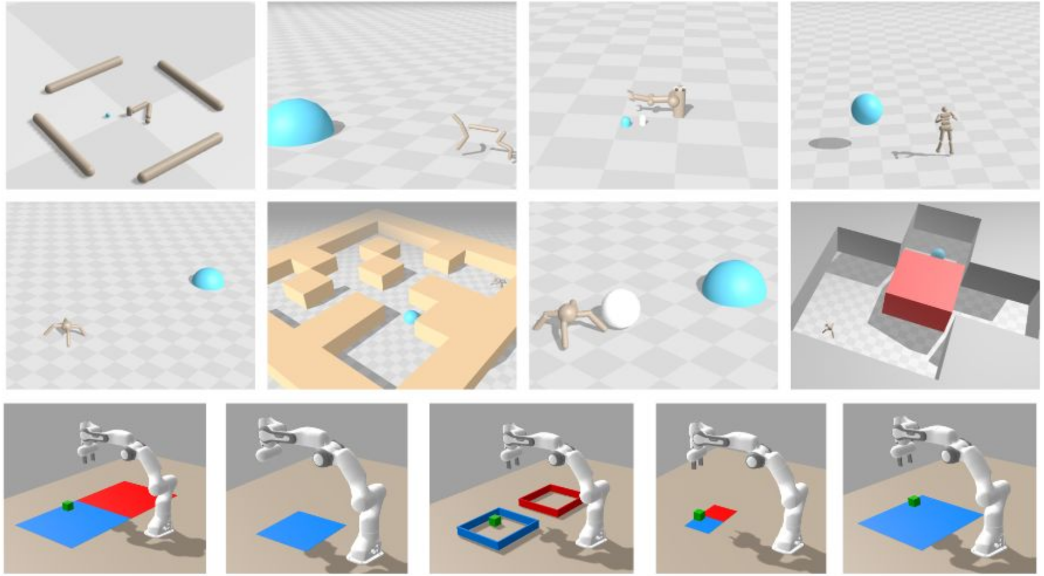

$\mathcal{L}(s, a, s_f^+, s_f^-)$: "Is this a future state or random state?"
state-action encoder, $\phi(s, a)$.
goal encoder, $\psi(s_f)$
state, s
action, a
future state, s_f^+
random state, s_f^-
trajectory 1
trajectory 2

Figure 1: **Reinforcement learning via contrastive learning.** Our method uses contrastive learning to acquire representations of state-action pairs ($\phi(s, a)$) and future states ($\psi(s_f)$), so that the representations of future states are closer than the representations of random states. We prove that learned representation corresponds to a value function for a certain reward function. To select actions for reaching goal s_g, the policy chooses the action where $\phi(s, a)$ is closest to $\psi(s_g)$.

Problem Suite

- JaxGCRL suite of GPU-accelerated environments (10 envs)

```
Ant_big_maze
ant_hardest_maze
arm_binpick_hard
Arm_push_easy
Arm_push_hard
Humanoid
humanoid_big_maze
humanoid_u_maze
Ant_u4_maze
ant_u5_maze
```



Results

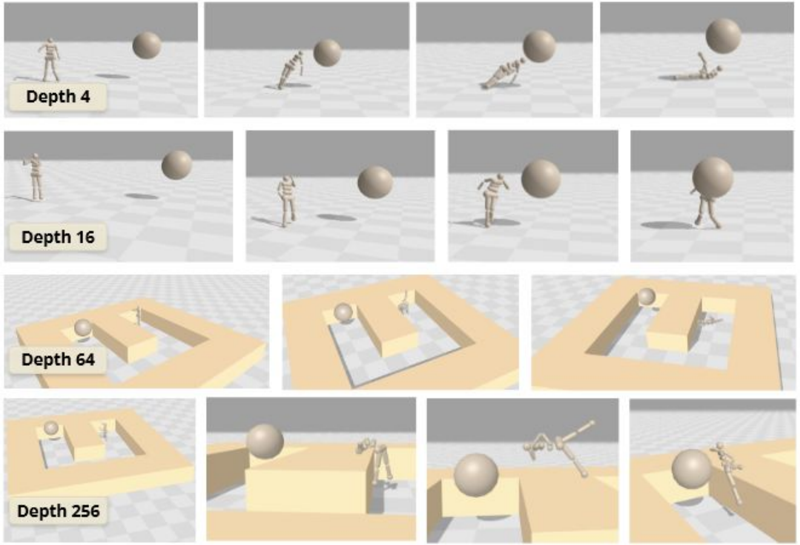


Figure 3: **Increasing depth results in new capabilities: Row 1**: A depth-4 agent collapses and throws itself toward the goal. **Row 2**: A depth-16 agent walks upright. **Row 3**: A depth-64 agent struggles and falls. **Row 4**: A depth-256 agent vaults the wall acrobatically.

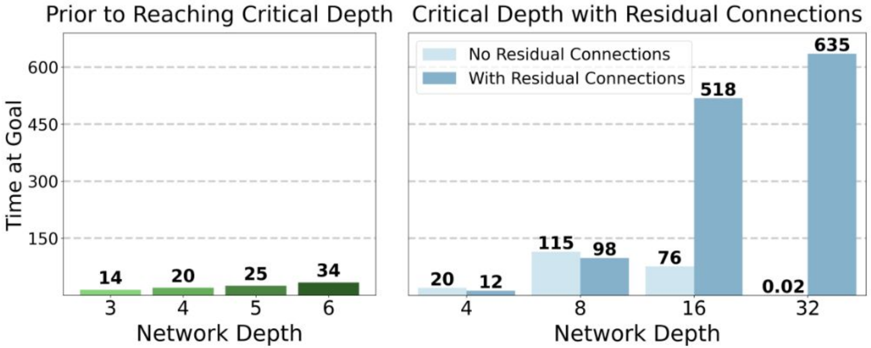


Figure 5: **Critical depth and residual connections.** Incrementally increasing depth results in marginal performance gains *(left)*. However, once a critical threshold is reached, performance improves dramatically *(right)* for networks with residual connections.

Results

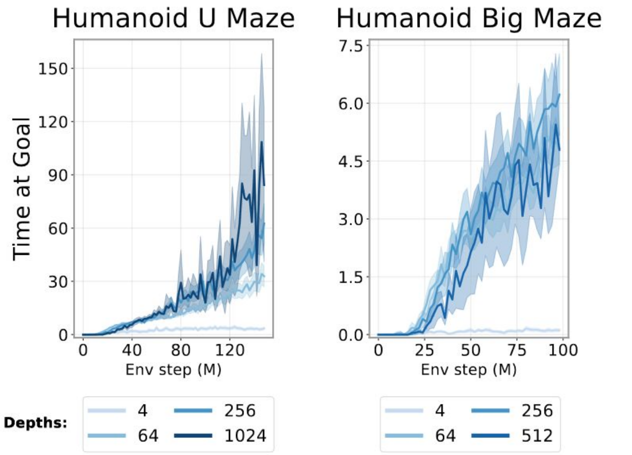


Figure 12: **Testing the limits of scale.** We extend the results from Figure 1 by scaling networks even further on the challenging Humanoid maze environments. We observe continued performance improvements with network depths of 256 and 1024 layers on Humanoid U-Maze. Note that for the 1024-layer networks, we observed the actor loss exploding at the onset of training, so we maintained the actor depth at 512 while using 1024-layer networks only for the two critic encoders.

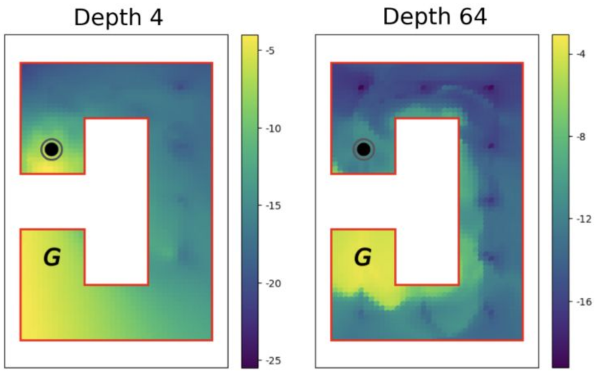


Figure 9: **Deeper Q-functions are qualitatively different.** In the U4-Maze, the start and goal positions are indicated by the ◉ and **G** symbols respectively, and the visualized Q values are computed via the L_2 distance in the learned representation space, i.e., $Q(s, a, g) = \|\phi(s, a) - \psi(g)\|_2$. The shallow depth 4 network *(left)* naively relies on Euclidean proximity, showing high Q values near the start despite a maze wall. In contrast, the depth 64 network *(right)* clusters high Q values at the goal, gradually tapering along the interior.

Key Take-aways

Simplicity may be the key

(to oral accepts, to winning the best-paper awards)

Empirical-only papers are ok

But

The results should be convincing

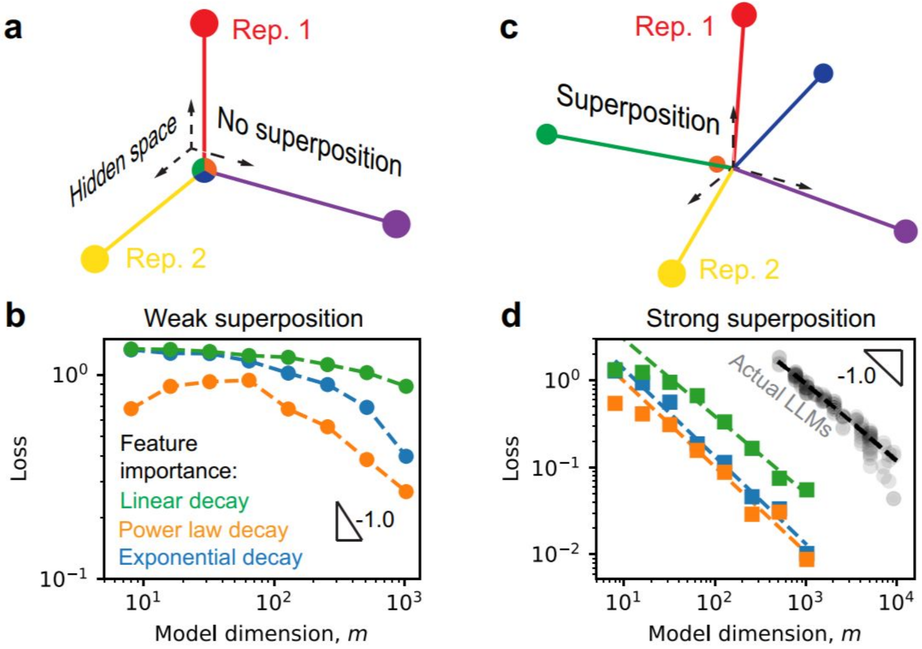
The analysis should be thorough

Yet Another Paper

Superposition Yields Robust Neural Scaling

Yizhou Liu and Ziming Liu and Jeff Gore, NeurIPS

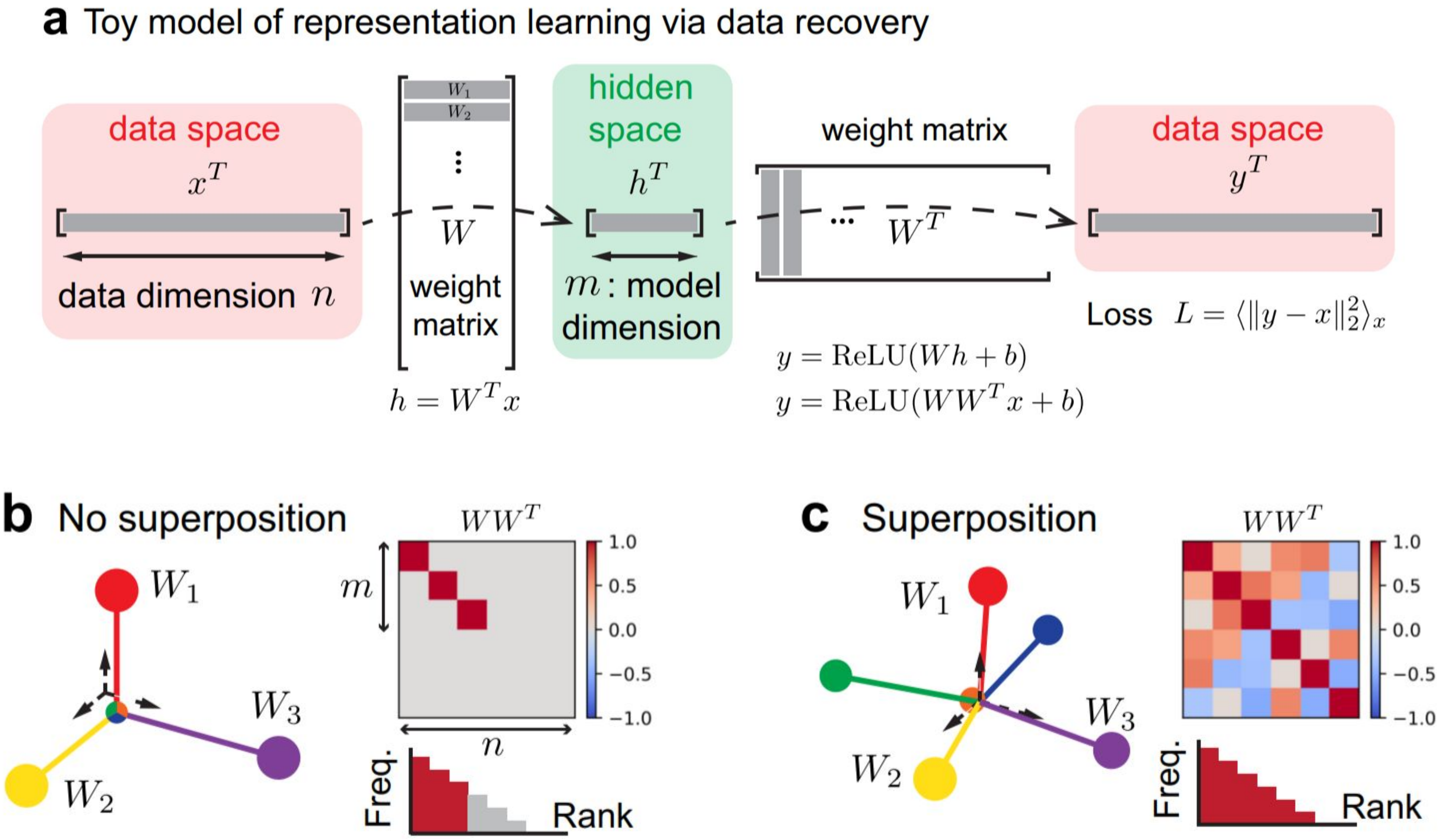
Superposition



Question: How will superposition influence the loss scaling with model dimension (width)?

Varying the degree of superposition and data structure, when is the loss a power law? And if the loss is a power law, what will the exponent be?

Toy model



a Toy model of representation learning via data recovery

b No superposition

c Superposition

$$h = W^T x$$

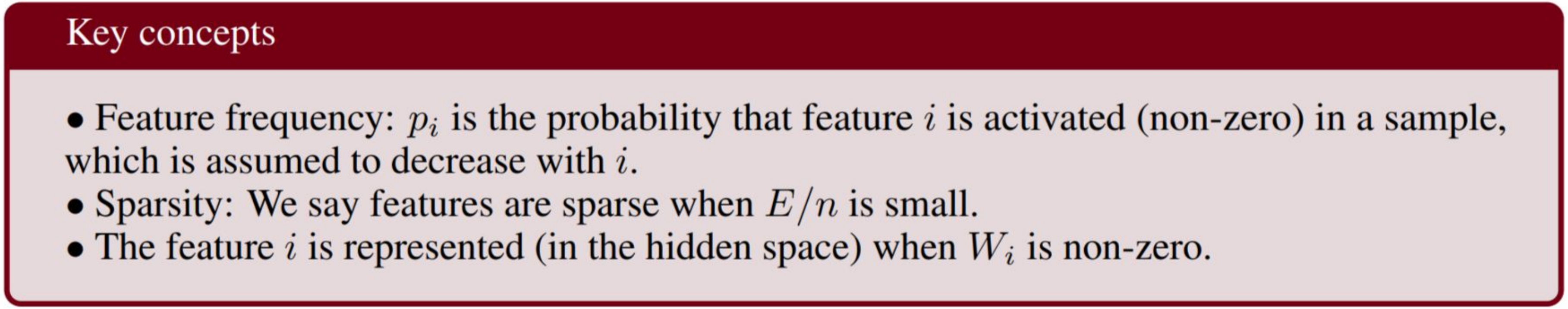
$$y = \mathrm{ReLU}(Wh + b)$$

$$y = \mathrm{ReLU}(WW^T x + b)$$

Loss $L = \langle \|y - x\|_2^2 \rangle_x$

Key concepts

$$x_i = u_i v_i,\ u_i \sim \text{Bernoulli}(p_i)\ \&\ v_i \sim U(0,2).$$



Key concepts

- Feature frequency: p_i is the probability that feature i is activated (non-zero) in a sample, which is assumed to decrease with i.
- Sparsity: We say features are sparse when E/n is small.
- The feature i is represented (in the hidden space) when W_i is non-zero.

$$W_{i,t+1} = \begin{cases} W_{i,t} - \eta_t \gamma W_{i,t},\ \gamma \geq 0, \\ W_{i,t} - \eta_t \gamma W_{i,t}(1/\|W_{i,t}\|_2 - 1),\ \gamma < 0, \end{cases}$$

Superposition tune

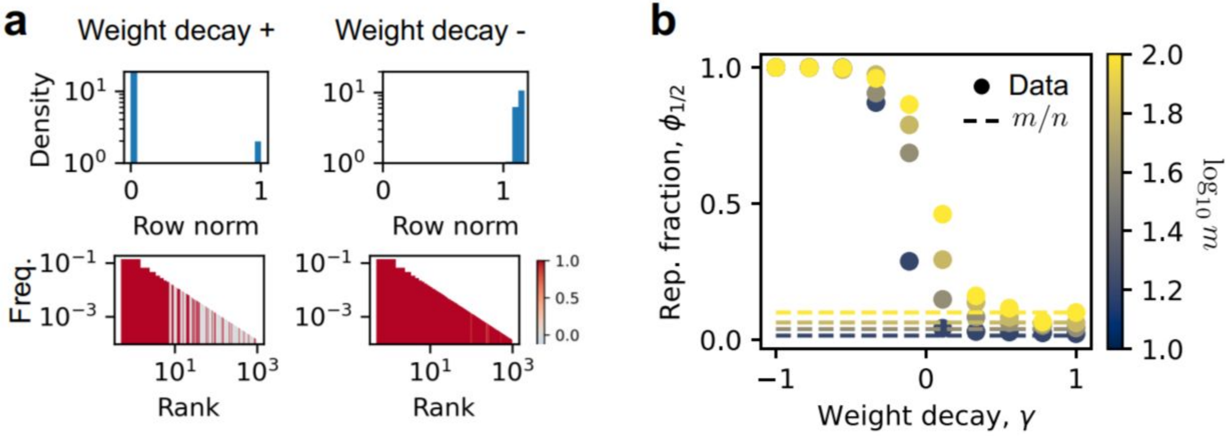


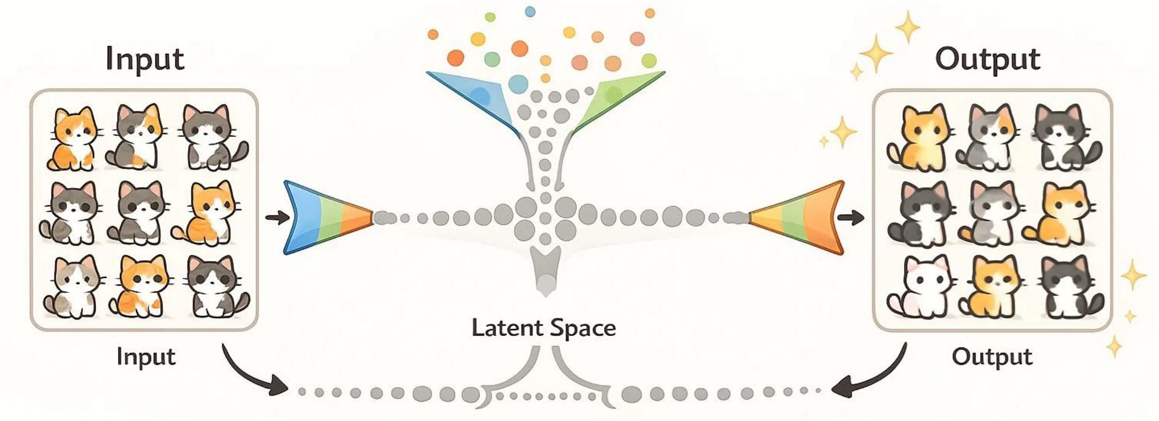
Figure 3: Weight decay can tune the degree of superposition. (a) Positive weight decay ($\gamma = 1$ in the figure) has $\|W_i\|_2$ near 0 or 1, with frequent features more likely to be represented (color means $\|W_i\|_2$ in frequency-rank plots). Negative weight decay ($\gamma = -1$) has $\|W_i\|_2$ around 1. We show results when $\alpha = 1$, $m = 100$, yet the claim is generally true. (b) For all models, small weight decays lead to strong superposition, and large weight decays lead to no superposition ($\phi_{1/2} \approx m/n$).

$$\phi_{1/2} = |\{i : \|W_i\|_2 > 1/2\}|/n,$$

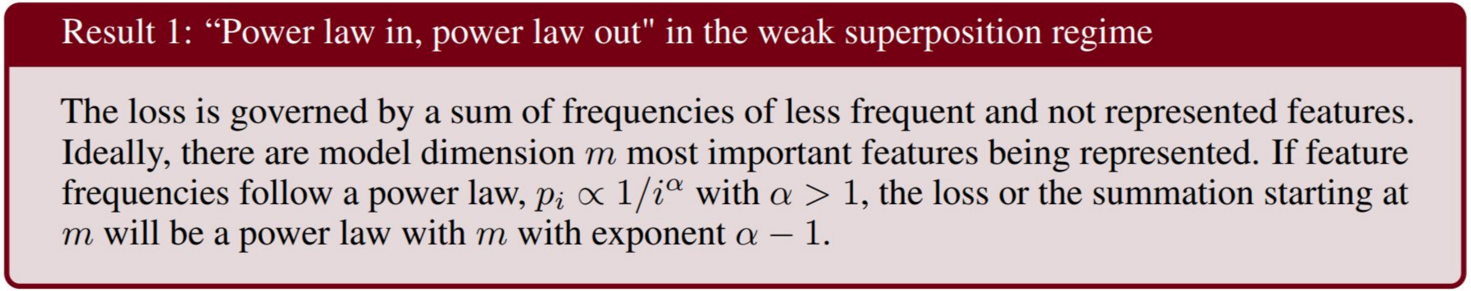
Weak superposition

$$L = \sum_{i>\phi_{1/2}n} <(x_i - <x_i>)^2> = \sum_{i>\phi_{1/2}n} \left(<v^2> p_i - <v>^2\, p_i^2\right) \approx <v^2> \sum_{i>\phi_{1/2}n} p_i$$

$$\int_m^n p_i\, di \propto m^{-\alpha+1}, \text{ when } n \gg m \text{ and } \alpha \gg 1$$



Weak superposition



Result 1: “Power law in, power law out" in the weak superposition regime

The loss is governed by a sum of frequencies of less frequent and not represented features. Ideally, there are model dimension m most important features being represented. If feature frequencies follow a power law, $p_i \propto 1/i^\alpha$ with $\alpha > 1$, the loss or the summation starting at m will be a power law with m with exponent $\alpha - 1$.

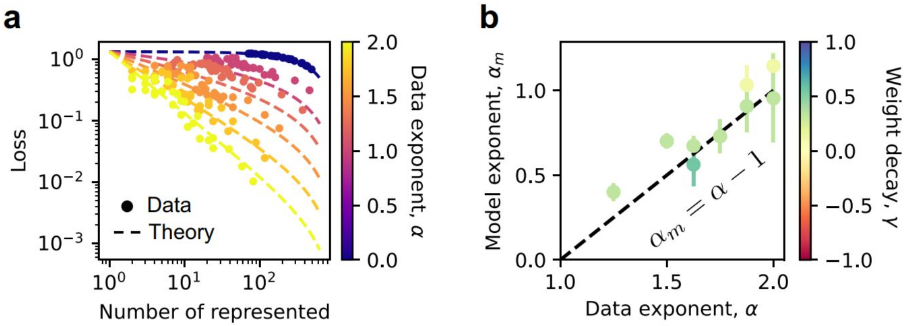
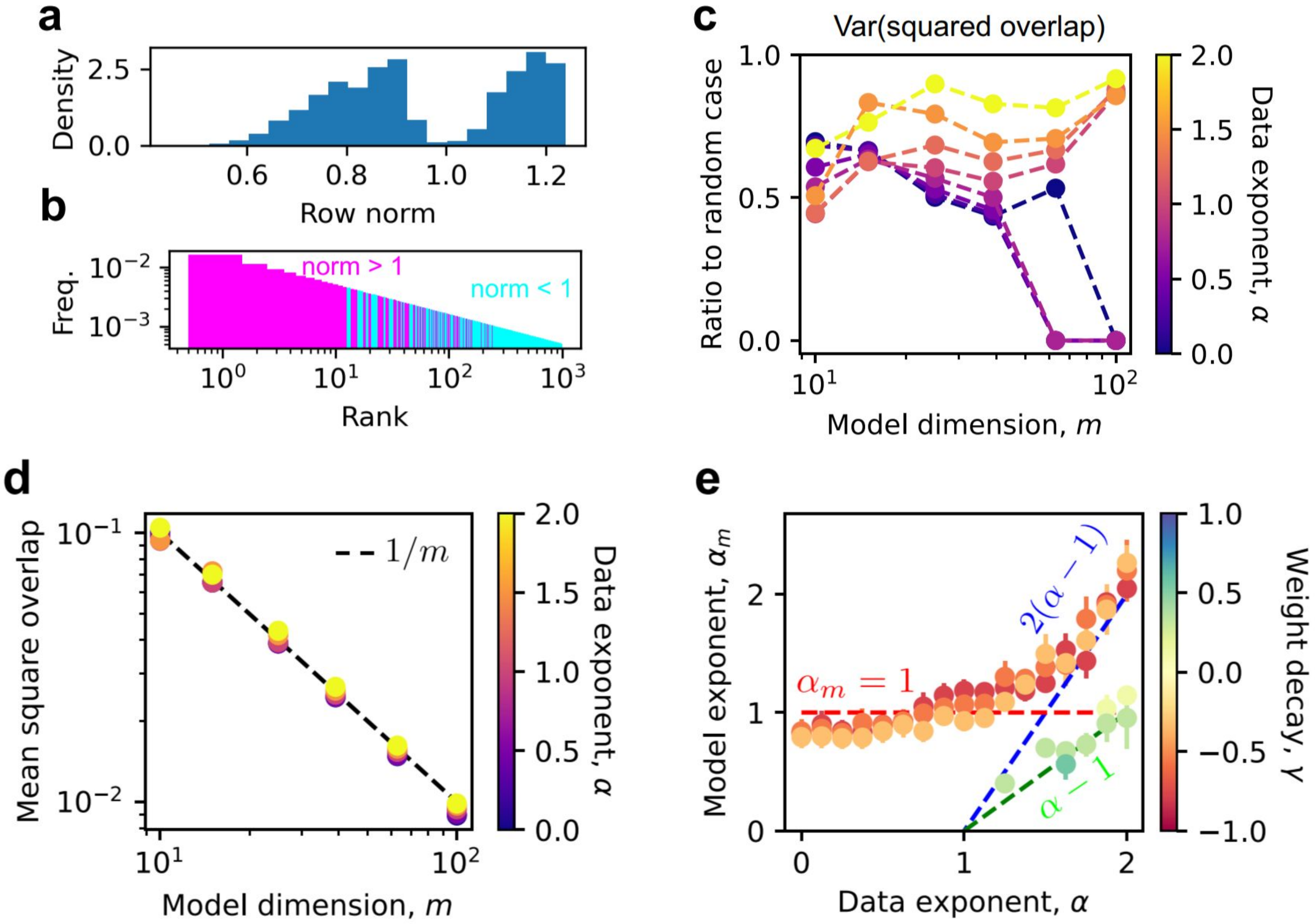
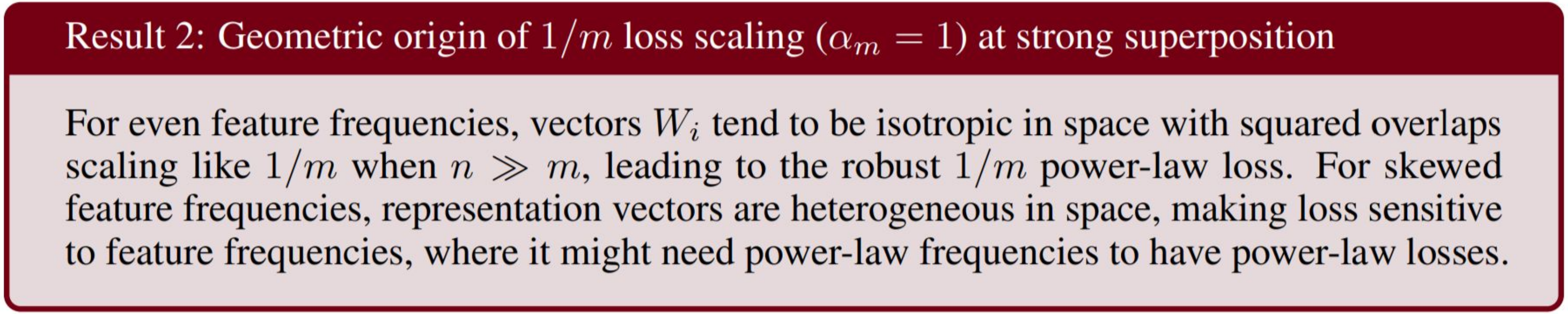


Figure 4: Loss at weak superposition can be well described by the frequency sum of ignored features. (a) Observation and theory at weak superposition (i.e., Equation (4) as a function of number of represented features, $\phi_{1/2}n$) agree when weight decay γ is positive. (b) For those closest to the ideal no superposition case, we expect $\alpha_m = \alpha - 1$, which is close to measured values. Error bars are standard errors. Details in Appendix D.5.

Strong superposition



Strong superposition



Result 2: Geometric origin of $1/m$ loss scaling ($\alpha_m = 1$) at strong superposition

For even feature frequencies, vectors W_i tend to be isotropic in space with squared overlaps scaling like $1/m$ when $n \gg m$, leading to the robust $1/m$ power-law loss. For skewed feature frequencies, representation vectors are heterogeneous in space, making loss sensitive to feature frequencies, where it might need power-law frequencies to have power-law losses.

LLMs

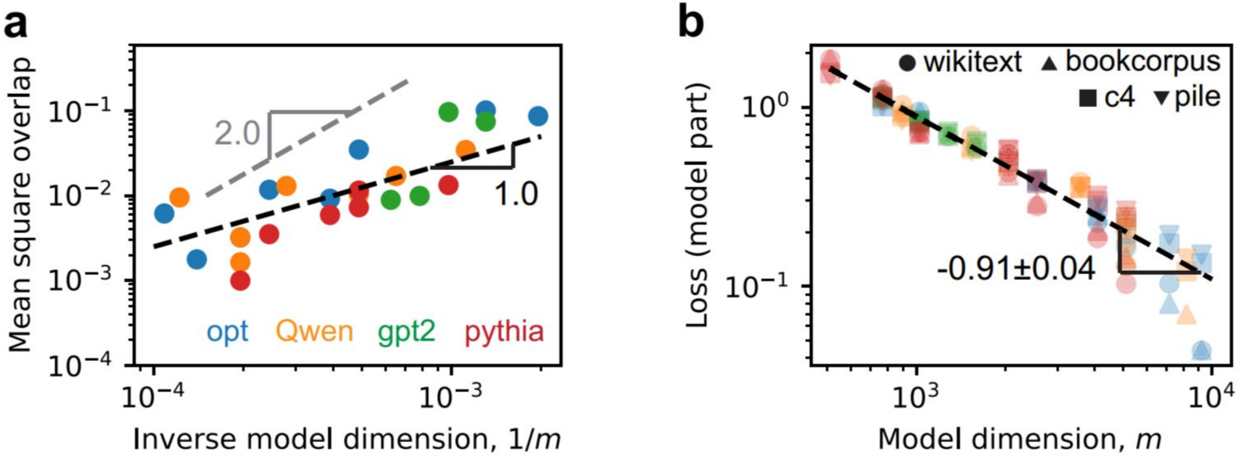


Figure 6: Superposition may explain the neural scaling law observed in actual LLMs. We evaluate four open-sourced model classes, Opt [39], GPT2 [40], Qwen [41], and Pythia [42], which have model sizes from around 100M to 70B (evaluation details in Appendix C). (a) We found the mean square overlaps of $W_i/\|W_i\|_2$ roughly follow $1/m$ scaling, where W is the language model head. (b) The model class is reflected by color as panel a, while we use shapes for evaluation datasets [43–46]. The loss related to model size is fitted as a power law, yielding empirical $\alpha_m = 0.91 \pm 0.04$ close to 1. More analysis in Appendix D.7.

Result 3: Superposition is an important mechanism behind LLM neural scaling laws

LLMs operate in the strong superposition regime. The squared overlaps of token representations scale as $1/m$, token frequencies are flat ($\alpha = 1$), and the model size relevant loss scales closely to $1/m$, agreeing with the toy model prediction.