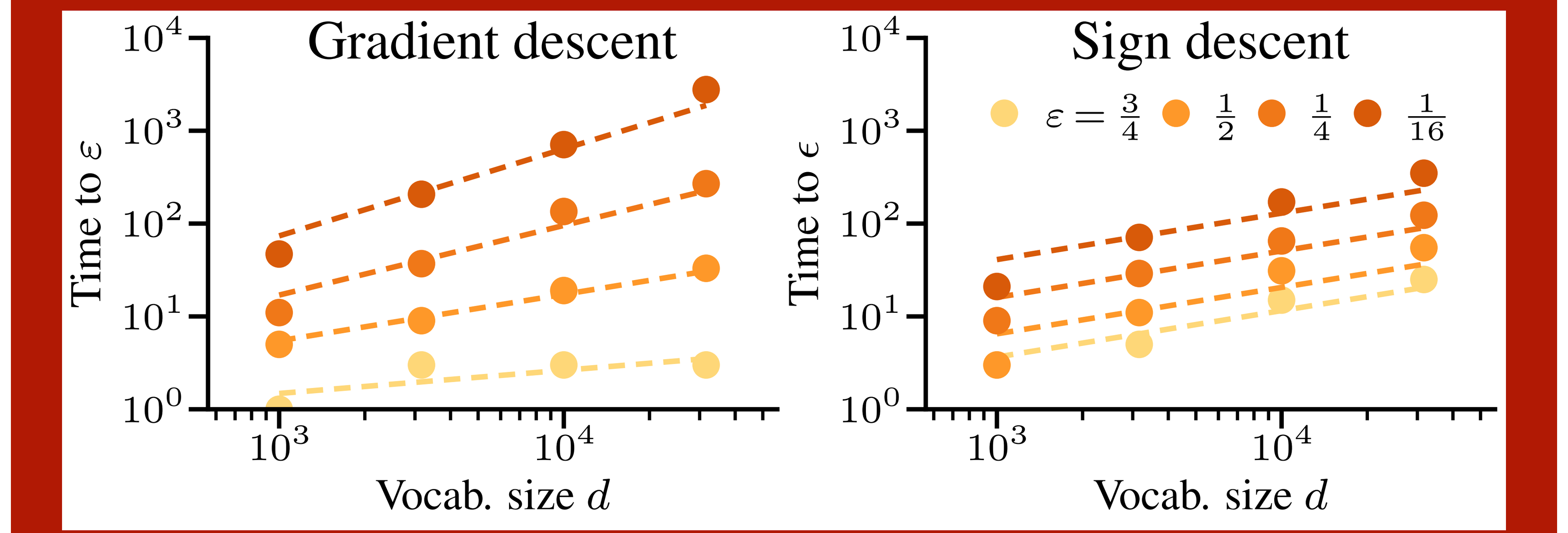
Scaling Laws for Next Token Prediction under Zipf's Law

Adam-like methods provably scale better to large vocabularies than gradient descent



Gradient descent: $t \asymp d$

Sign descent: $t \asymp \sqrt{d}$

Scaling Laws for Gradient Descent and Sign Descent for Linear Bigram Models under Zipf's Law

Frederik Kunstner
Francis Bach

arxiv.org/abs/2402.19449

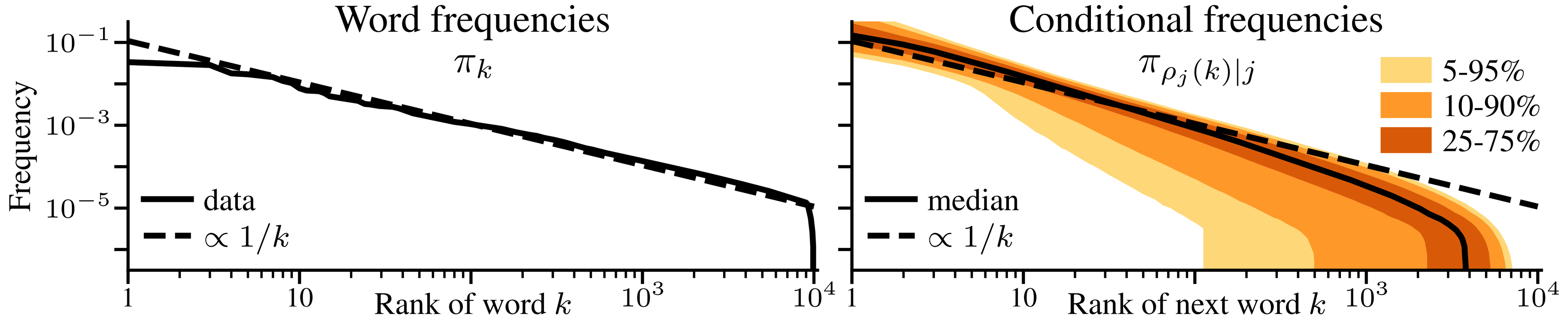
Simplified language task: Linear Bigram Model

Consider a linear model to predict the next token based on the current one, using the squared loss:

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}}\left[\frac{1}{2}\|\mathbf{W}\mathbf{x}-\mathbf{y}\|^2\right] \qquad \begin{array}{lll} \mathbf{x} = \mathbf{e}_x, & x \sim \boldsymbol{\pi}, & \pi_k \propto 1/k^\alpha \\ \mathbf{y} = \mathbf{e}_y, & y \sim \boldsymbol{\pi}_{\cdot\,|\,x} & \pi_{\rho_i(k)\,|\,i} \propto 1/k^\alpha \end{array}$$

where $\rho_i(k)$ gives the k-th most frequent token following token i, $\pi_{rho_i(k)\,|\,i} > \pi_{\rho_i(k+1)\,|\,i}$.

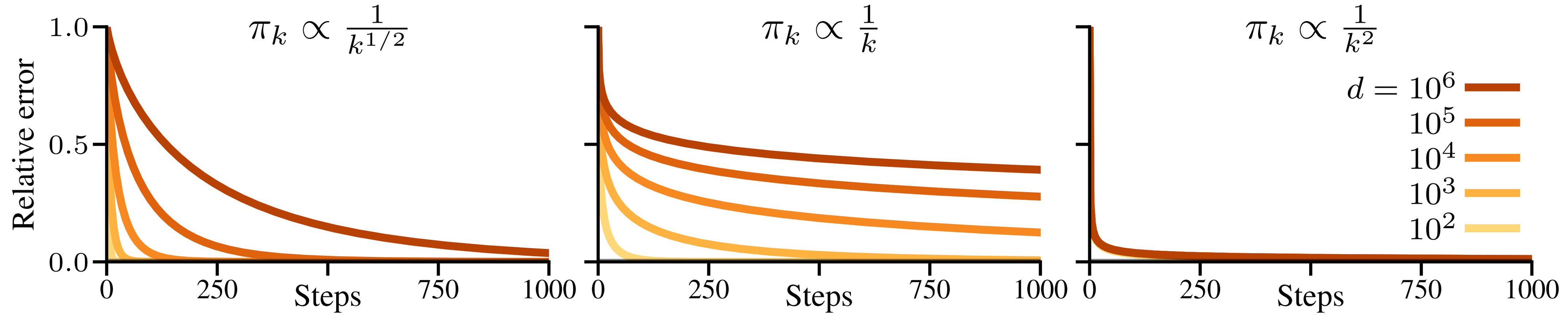
Approximate model of text data with – Zipf's law with $\alpha = 1$



Zipf's law is bad for gradient descent – optimization gets harder in high dimension

Performance depends heavily on algorithm and exponent α.

$\alpha \gg 1$	very imbalanced, rare words too rare to matter	$\to$ GD works even in high d
$\alpha \leq 1$	more uniform, but rare words matter more	$\to$ GD gets slower with d
$\alpha = 1$	worst case: rare words matter *and* learned slowly	$\to$ worst scaling in d



The scaling question: As d increases, do we need more iterations t?

to reach a relative error of ε $\qquad r_d(t) := \dfrac{\mathcal{L}_d(\mathbf{W}_t) - \mathcal{L}_d(\mathbf{W}^*)}{\mathcal{L}_d(\mathbf{W}_0) - \mathcal{L}_d(\mathbf{W}^*)} = \varepsilon$

Main result for Zipf's law: GD needs $t \asymp d$, but sign descent $t \asymp \sqrt{d}$

We obtain scaling laws with different functional forms depending on α. For GD

$$r_d(t) \asymp \frac{e^{-2t/d^\alpha}}{2t/d^\alpha} \qquad r_d(t) \asymp 1 - \frac{\log(t)}{\log(d)} \qquad r_d(t) \asymp t^{-p} \qquad \text{(joint limit in } d \text{ and } t\text{)}$$

Which algorithm dominates depends heavily on α. For $\alpha > 1$, GD scales better.

	$\alpha < 1$	$\alpha = 1$	$\alpha > 1$
Gradient descent	$t \asymp d^\alpha$	$t \asymp d^{1-\varepsilon}$	$t \asymp 1$

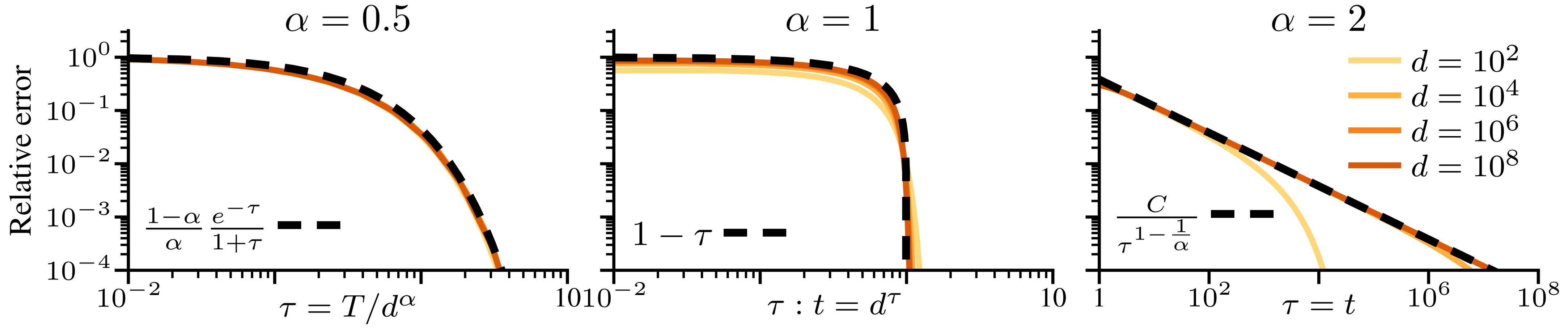
	$\alpha < 1/2$	$\alpha = 1/2$	$\alpha > 1/2$
Sign descent	$t \asymp 1$	$t \asymp \sqrt{d^{1-\varepsilon}}$	$t \asymp \sqrt{d}$

Scaling for Gradient Descent – Asymptotic convergence rate

Closed form expression for gradient descent on quadratics, as $d \to \infty$,

$$r_d(t) = \sum_{k=1}^{d} \underbrace{\pi_k}_{\text{Importance}} \underbrace{(1 - \eta\pi_k)^{2t}}_{\text{Speed}} \asymp I_d(t) := \frac{1}{H_{\alpha,d}} \int_1^d \frac{1}{k^\alpha}\left(1 - \frac{1}{k^\alpha}\right)^{2t} \mathrm{d}k$$

Scale $t(d)$ such that $\lim_{d\to\infty} I_d(t(d))$ is well-defined. Asymptotic, but good approximation for finite d

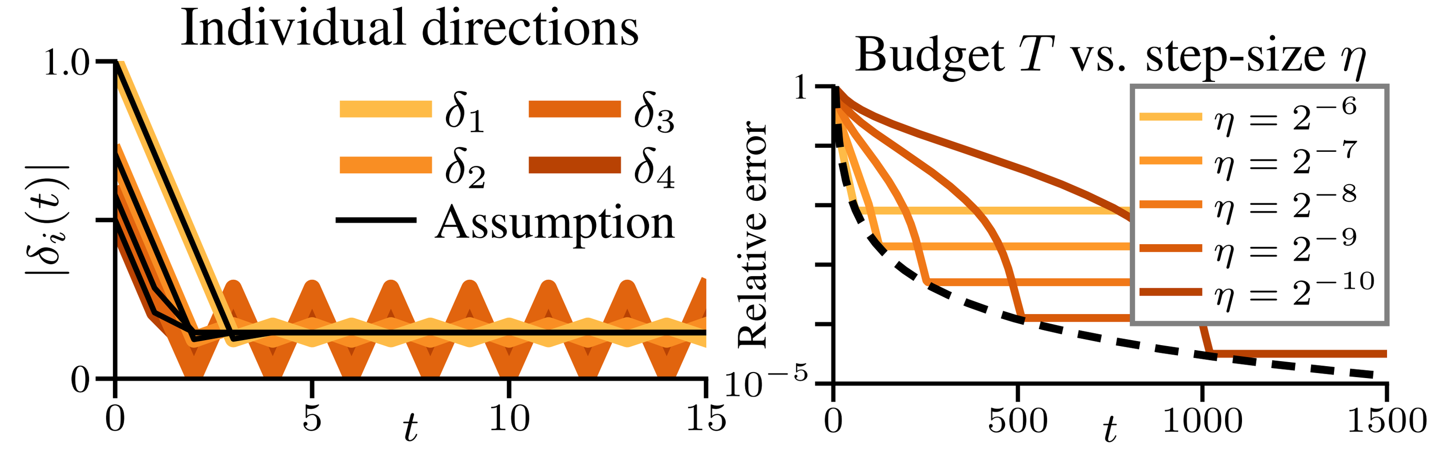


Improvement of Sign Descent – better scaling in high dimension

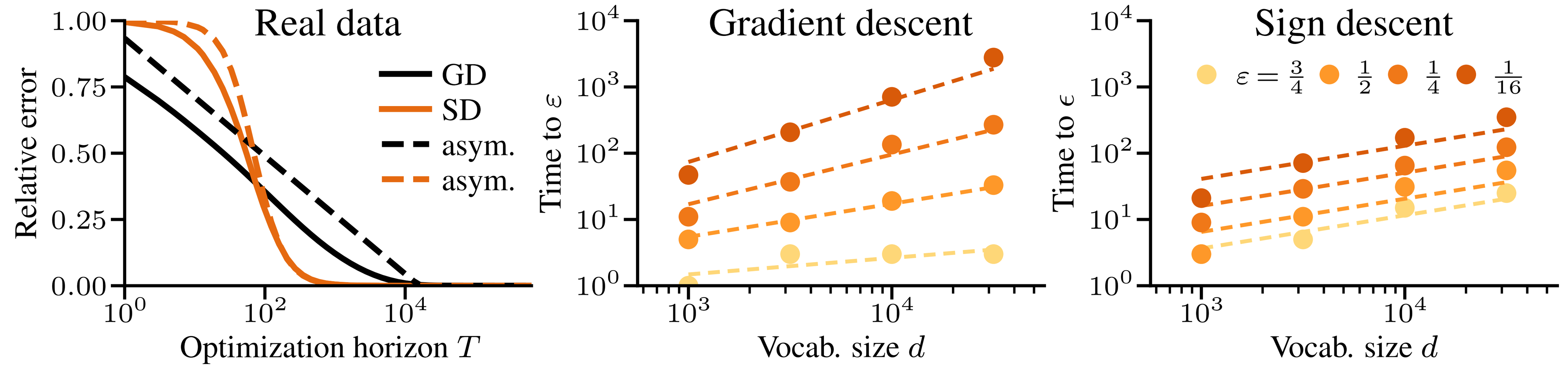
Sign descent uses a uniform update across all directions, going faster on rare words
But it also does not converge with a fixed step-size; instead it oscillates around the optimum.

Need to scale the step-size η as a function of dimension d and the total iteration budget T.

Despite small step-size, more progress than gradient descent



Real data – Linear model on OpenWebText with increasing vocabulary size



Beyond squared loss – More extreme difference observed for cross-entropy loss

Gradient descent still appears to need $t \asymp d$ steps, while sign descent needs $t \asymp 1$ steps.