A Survey of Continuous Time Dynamics of Transformers

Alan Chen*, Ivan Lee*, Yair Shenfeld

Spring 2025

1 Introduction

Transformers have recently emerged as the single dominant paradigm in language (and vision) [Vaswani et al., 2017, Achiam et al., 2023, Devlin et al., 2019, Dosovitskiy et al., 2020]. Models are generally masked, where the objective is to predict the masked out token, or autoregressive (causally masked), where the objective is instead to predict the next token after a sequence of tokens. Because performance has been observed to predictably increase with scale [Kaplan et al., 2020, Hoffmann et al., 2022], significant research is invested into understanding the empirical engineering tricks that allow for maximization of the available compute [Shazeer et al., 2017, Liu et al., 2024]. Simultaneously, out of a desire to understand the black boxes of models, theoretical perspectives have also seen a complementary rise in popularity, especially from the perspective of understanding training dynamics i.e. how the model parameters and behavior shifts under the gradient flow [Arora et al., 2018, Menon, 2024, inter alia].

Recently, a group at MIT under the supervision of Philippe Rigollet have pushed out a body of work describing a continuous time dynamical perspective on the forward pass of a transformer (as opposed to the training dynamics) [Geshkovski et al., 2023a,b, 2024a,b, Karagodin et al., 2024]. In this report we survey and expand on the existing findings with new numerical experiments. As an outline, our contributions are as follows:

1. We present an empirically grounded overview of a recent flurry of theoretical results modeling the continuous time infinite depth limit of transformer dynamics, beginning from empirical principles and reorganizing results suitably. (§2, 3, 4)
2. We replicate and extend numerical simulation of the continuous time model to situations outside of the theoretical assumptions (§5.1).
3. We characterize the ability of flow matching models (flows that depend only on time t) to model simple attention/transformer flows (flows that also depend on the measure μ) (§5.3).
4. We use the theoretical results and quantities to also analyze real language models (§5.4).

All code can be found at `https://github.com/IvanLee329/APMA-2822B`.

2 Preliminaries

2.1 Modern Transformer Overview

Attention. The fundamental building block in modern transformer based models is the **attention module** [Vaswani et al., 2017]. The attention module consists of 3 key matrices: the query, key, and value matrices. Intuitively, the attention module uses the queries to generate similarity (defined via an inner product) scores for the keys. The scores are then normalized to a probability distribution that weights a sum of the associated values.

Modern transformers largely utilize **multi-head attention**, which extends the standard attention operation to $H \geq 1$ heads in an attention layer. For head $h \in [H]$, after projecting the original input into separate head spaces via weight matrices $Q_h, K_h, V_h \in \mathbb{R}^{d_h \times d}$, the outputs of the individual heads are concatenated back together and projected via a single output matrix $W_O \in \mathbb{R}^{(d_h \times h) \times d}$. For simplicity, usually $d_h \times h = d$ so that W_O is square.

Let $\mathbf{x} = \{x_k\}_{k=1}^n \in \mathbb{R}_d^n$ be a sequence of inputs to the multi-head attention module. Then, the explicit form of the operation can be written as

$$\text{Attention}_h(\mathbf{x}) = \text{Softmax}_j\left(\beta(Q_h x_k)^T(K_h x_j)\right)(V_h x_k), \tag{1}$$

$$\text{MultiHeadAttention}(\mathbf{x}) = \text{Concat}_h(\text{Attention}_h(\mathbf{x}))W_O, \tag{2}$$

where for $v \in \mathbb{R}^n$

$$\text{Softmax}(\beta v) = \frac{e^{\beta v_k}}{\sum_{j=1}^n e^{\beta v_j}}. \tag{3}$$

The softmax operator is nothing new: in some sense, it is the "computer scientist's" revival of concepts known to physicists as the Boltzmann/Gibbs distributions. Work in interpreting and understanding the attention computation from a human-interpretable perspective has risen in popularity in recent years [Geshkovski et al., 2023b, i.a.]. Furthermore, there is much research in making attention more efficient as well, introducing modifications or empirical tweaks during inference to improve compute efficiency [Katharopoulos et al., 2020].

Causal modeling. Modern transformers also use **causal masking**: the attention module is modified such that the similarity scores are only computed between a query and keys that come *prior* to the query in the sequence.

$$\text{CausalAttention}_h(\mathbf{x}) = \text{Softmax}_{j \leq k}\left(\beta(Q_h x_k)^T(K_h x_j)\right)(V_h x_k). \tag{4}$$

This modification is the workhorse of *autoregressive* language generation models, which are trained to predict next-token probabilities (i.e. the distribution $p(x_{n+1}|\mathbf{x})$). The causal mask prevents the model from "cheating" during training by peeking ahead at future tokens to generate the next token.

The remaining vital components of the transformer are skip connections, layer normalization, and multi-layer perceptrons (MLPs).

Skip connections. A skip connection across a module is a simple idea that prevents information from being lost during the progression of the model. Let $\mathbf{x}_t$ be the token states after each layer $t \in L$: instead of generating $\mathbf{x}_{t+1}$ as

$$\mathbf{x}_{t+1} = \text{Module}(t, \mathbf{x}_t),$$

$\mathbf{x}_t$ is instead also added onto Module($\mathbf{x}_t$) in the "skip connection", giving the update equation

$$\mathbf{x}_{t+1} = \text{Module}(t, \mathbf{x}_t) + \mathbf{x}_t. \tag{5}$$

Intuitively, each progressive module generates a single additive update to the internal state, rather than overwriting the state completely. Notice the theoretical robustness of this idea: assuming Module is constant with respect to t, in the $\Delta t \to 0$ limit, this equation is actually just the discretization of the continuous time ODE

$$\dot{\mathbf{x}} = \text{Module}(\mathbf{x}). \tag{6}$$

This will be the foundation of the theoretical analysis we survey in this report and its validity depends fundamentally on skip connections.

LayerNorm. Because of the additive structure of skip connections, engineers found the norms of their internal states quickly diverging, leading to unstable gradient descent. Various normalization schemes were constructed to prevent this problem; the one most conventionally used in transformers is layer normalization, defined as

$$\text{LayerNorm}(x) = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}(x)}} \cdot \gamma + \beta, \tag{7}$$

where γ and β are learnable parameters and $\mathbb{E}[x]$ and $\text{Var}(x)$ are computed empirically for each x.

In contrast to the clean mathematical translation of skip connections, layer normalization seems to be the bane of existence for many theoretical and interpretability researchers because of its odd structure[1]. In the work we survey, projection onto the unit sphere in d dimensions (denoted $\mathbb{S}^{d-1}$) is used instead as a crude approximation of layer normalization [2].

MLPs. The multilayer perceptron is the quintessential neural network, consisting of an interleaving composition of affine functions and nonlinearities (activation functions). In transformers, it is usually just a single layer with dimension d_{MLP} (usually $4 \times d$), given as

$$\text{MLP}(x) = W_2\sigma(W_1 x + b_1) + b_2 \tag{8}$$

where $W_1 \in \mathbb{R}^{d_{\text{MLP}} \times d}$, $b_1 \in \mathbb{R}^{d_{\text{MLP}}}$, $W_2 \in \mathbb{R}^{d \times d_{\text{MLP}}}$, and $b_2 \in \mathbb{R}^d$. In the transformer, this module acts on each sequence position $x_i \in \mathbf{x}$ independently: as such, it is context independent and acts similarly to a lookup table.

Thus, in the complete form, the forward pass of a modern decoder-only transformer used in language modeling consists of an interleaving of skip connections, layer normalizations, attention operations, and MLPs, with all parameters dependent on the layer index ℓ. From layer ℓ to $\ell + 1$ the operation can be written as

$$\begin{aligned} \mathbf{y}^{(\ell)} &= \text{LayerNorm}\left(\mathbf{x}^{(\ell)} + \text{CausalAttention}^{(\ell)}(\mathbf{x}^{(\ell)})\right), \\ \mathbf{x}^{(\ell+1)} &= \text{LayerNorm}\left(\mathbf{y}^{(\ell)} + \text{MLP}^{(\ell)}(\mathbf{y}^{(\ell)})\right). \end{aligned} \tag{9}$$

[1] ... which is likely a result of its motivation being just for empirics.

[2] This toy normalization is a special case of root mean squared normalization (RMSNorm in modern programming libraries).

Article	Normalization	Time-dependent QKV	$V \neq I$	MHA	MLP	Causal	$d > 1$
Geshkovski et al. [2023a]	✗	✗	✓	✗	✗	✗	✗
Geshkovski et al. [2023b]	✓	✗	✗	✗	✗	✗	✓
Geshkovski et al. [2024b]	✓	✓	✓	✗	✓	✗	✓
Geshkovski et al. [2024a]	✓	✗	✗	✗	✗	✗	✓
Karagodin et al. [2024]	✓	✗	✗	✗	✗	✓	✓

Table 1: An overview of the assumptions we find being consistently used across the main works. Works may occasionally mention relaxing assumptions or provide numerical experiments that relax assumptions but we categorize the main theoretical contributions and model formalism.

2.2 Theoretical Simplifications

A flurry of work in recent years has been proposed to theoretically analyze the attention dynamics from the viewpoint of optimal transport and statistical physics, noting and proving the existence of theoretical phenomenon like clustering that emerge naturally from the transformer architecture Geshkovski et al. [2023b]. In Table 1, we outline the main assumptions that each work covers based on a brief categorization of the main theoretical contributions of each report.

We notice two consistencies: the most consistent modeling inclusion is modeling the layer normalization performed at each timestep. As aforementioned, layer normalization is approximated via a projection onto $\mathbb{S}^{d-1}$. On the other hand, the most consistent omission in modeling is multi-head attention: we always set $H = 1$ to avoid having to deal with understanding an additional matrix projection W_O and drop the subscripts on Q, K, and V for clarity. For future work, the theoretical results for multi-head attention could be quite rich - for example, one can consider taking limits like $H \to \infty$ to analogize with the results in expressivity and dynamics of infinite-width limit neural networks.

3 Continuous Model for Transformers

In order to more deeply study the dynamics of the transformer model, we consider the infinite depth limit that leads to a clean continuous time model for the forward pass of a transformer. We will work from the most general case to simple cases that we can study in depth. To begin, a direct translation of equation (9) using the skip connection limit in equation (6) is difficult because of the multiple applications of layer normalization. Instead, we simplify to just a single layer normalization applied after the MLP, giving the continuous time limit

$$\dot{\mathbf{x}} = \text{LayerNorm}(\text{CausalAttention}(t, \mathbf{x}) + \text{MLP}(t, \mathbf{x})). \tag{10}$$

Applying aforementioned assumptions such as replacing layer normalization with the projection onto the sphere and ignoring causal masking, for each i the model becomes simply,

$$\dot{x}_i(t) = \mathbf{P}^{\perp}_{x_i(t)} \left(\text{Attention}(t, \mathbf{x}) + \text{MLP}(t, x_i)\right). \tag{11}$$

In this very general form, we can define

$$\theta : [0, T] \to \Theta \tag{12}$$

as the set of parameters for determining $\text{Attention}(t, \cdot)$ and $\text{MLP}(t, \cdot)$, where Θ is some large, real Euclidean parameter space depending on architecture. Under this setting, Geshkovski et al. [2024b] prove a crucial expressivity result.

Theorem 3.1 (Measure-to-Measure Interpolation, Geshkovski et al. [2024b]). *Let $(\mu^i, \nu^i) \in \mathcal{P}(\mathbb{S}^{d-1}) \times \mathcal{P}(\mathbb{S}^{d-1})$ be initial and target measures. Under suitable assumptions on μ^i and ν^i, for any $T > 0$, there exists a set of parameters $\theta \in L^\infty((0,T);\Theta)$ such that the unique solution μ^i_* for equation 11 converges to ν^i at time T in W_2 distance. Furthermore, θ can be chosen to be piecewise constant.*

The proof itself is quite technical so we focus on the key takeaway. In particular, in practical applications, ν^i is often a point mass measure (for example on the next token) - this result shows that transformers are architecturally well-suited for this task as there exists a continuous-time parameter set such that the initial measure (also discrete) is pushed forward to ν.

The details of the transformer become clear from equation 11. At this point, we simplify even further and remove the MLP. We reinclude it in the numerical simulations, but focus on the study of the attention mechanism for now. Because of the normalization, we note the transformer is a flow map on $\left(\mathbb{S}^{d-1}\right)^n$, where n is the number of tokens. Indeed, after all of these simplifications, the transformer evolves a sequence of token inputs $x(0) = (x_1(0), \dots, x_n(0)) \in \left(\mathbb{S}^{d-1}\right)^n$ according to the following dynamics:

$$\dot{x}_i(t) = \boldsymbol{P}^{\perp}_{x_i(t)} \left(\frac{1}{Z_{\beta_i}(t)} \sum_{j=1}^{n} e^{\beta \langle Q(t)x_i(t), K(t)x_j(t) \rangle} V(t) x_j(t) \right), \ i \in [n], t \geq 0 \tag{13}$$

where

$$\boldsymbol{P}^{\perp}_x(y) = y - \langle x, y \rangle x$$

is the projection of $y \in \mathbb{R}^d$ onto $T_x\mathbb{S}^{d-1}$ and the partition function $Z_{\beta,i}(t)$ is given by

$$Z_{\beta,i}(t) = \sum_{k=1}^{n} e^{\beta \langle Q(t)x_i, K(t)x_k(t) \rangle}. \tag{14}$$

We begin with the simplest model: we consider Q, K, and V **time-independent**. Under these simplifying assumptions, (13) reduces to

$$\dot{x}_i(t) = \boldsymbol{P}^{\perp}_{x_i(t)} \left(\frac{1}{Z_{\beta,i}(t)} \sum_{j=1}^{n} e^{\beta \langle Qx_i(t), Kx_j(t) \rangle} V x_j(t) \right), \ i \in [n], t \geq 0 \tag{15}$$

and (14) reduces to

$$Z_{\beta,i}(t) = \sum_{k=1}^{n} e^{\beta \langle Qx_i, Kx_k(t) \rangle}$$

To make the structure more apparent, we rewrite (15) in terms of empirical measures. We will also reduce to $Q = K = V = I_d$ for simplicity but will mention when results hold for wider classes of Q, K, and V. Let $\mu(t, \cdot) \in \mathcal{P}(\mathbb{S}^{d-1})$ be an empirical measure given by

$$\mu(t, \cdot) = \sum_{i=1}^{n} \delta_{x_i(t)}(\cdot). \tag{16}$$

Then, we can define the vector field $\mathcal{X}[\mu] : \mathbb{S}^{d-1} \to T_x\mathbb{S}^{d-1}$ given as

$$\mathcal{X}[\mu](x) = \boldsymbol{P}^{\perp}_x \left(\frac{1}{Z_{\beta,\mu}} \int e^{\beta \langle x, y \rangle} y \, d\mu(y) \right) \tag{17}$$

where

$$Z_{\beta,\mu} = \int e^{\beta\langle x,y\rangle}\ d\mu(y).$$

Then the dynamics (15) is equivalently

$$\dot{x}_i(t) = \mathcal{X}[\mu(t)](x_i(t)). \tag{18}$$

By the equivalence of the Eulerian and Lagrangian descriptions of fluid flow, we see that the evolution of $\mu(t)$ satisfies the continuity equations:

$$\begin{cases} \partial_t \mu + \operatorname{div}\left(\mathcal{X}[\mu]\mu\right) = 0 \\ \mu(0,\cdot) = \mu(0) \end{cases} \tag{19}$$

Remark 1. *The fact that θ can be piecewise constant in Theorem 3.1 is quite noteworthy. Let θ have K time intervals*

$$\{(0,t_1),(t_1,t_2),\ldots,(t_{K-1},T)\} \tag{20}$$

on which it is constant. Then, it is not hard to see that there exists (since the value of the flow depends smoothly on V) $\tilde{\theta}:\{0,1,\ldots,K-1\}\to\Theta$ and $\tilde{\mu}_t$ as

$$\begin{cases} \tilde{\mu}_t = \tilde{\mu}_{t-1} + \operatorname{div}\left(\mathcal{X}_{\tilde{\theta}(t-1)}[\tilde{\mu}_{t-1}]\tilde{\mu}_{t-1}\right), \\ \tilde{\mu}_0 = \mu. \end{cases} \tag{21}$$

such that

$$\mu_T = \int_0^T \operatorname{div}(\mathcal{X}_\theta[\mu]\mu)dt = \tilde{\mu}_K. \tag{22}$$

In particular, $\tilde{\theta}$ exactly represents the parameters of a discrete time transformer with K layers. One can ask how many switches we need (i.e. how big K is). Let n be the sequence length, in which case we have a specific result and a general one:

1. *In the specific case of ν^i being a point mass measure, $O(d\cdot N)$ switches are needed in θ.*
2. *In general, $O(2^d)$ switches are needed.*

3.1 Monotonic Quantities

As with many other flows, we would like to see if equation (19) nets any monotonic quantities. Although entropy is not monotonic, a related quantity is.

Definition 3.2. *Given $\beta,\mu\in\mathcal{P}(\mathbb{S}^{d-1})$, the interaction energy $E_\beta[\mu]:\mathcal{P}(\mathbb{S}^{d-1})\to\mathbb{R}$ of a measure μ is defined by*

$$\boldsymbol{E}_\beta[\mu] = \frac{1}{2\beta}\int\int e^{\beta\langle x,x'\rangle}d\mu(x)d\mu(x'). \tag{23}$$

For $\beta>0, d\geq 2$, the global minimum of $\boldsymbol{E}_\beta$ over $\mathcal{P}(\mathbb{S}^{d-1})$ is the uniform measure. Additionally, $\boldsymbol{E}_\beta$ is maximized at any point mass measure. Thus, in some sense, the interaction energy can be thought of as an "inverse" entropy which is large when the distribution is "condensed" and small when it is "spread out".

Lemma 3.3 (Geshkovski et al. [2023b])**.** *The interaction energy (definition 3.2) is monotonic along the flow in equation* (19)*.*

Proof. By product rule, we can see that

$$\frac{d}{dt}E_\beta[\mu(t)] = \frac{1}{\beta}\iint_{(\mathbb{S}^{d-1})^2} e^{\beta\langle x,x'\rangle} d\partial_t\mu(t,x)d\mu(t,x'). \tag{24}$$

Now, using that $\mu(t,x)$ solves the flow in equation (19), we know that

$$\partial_t\mu(t,x) = -\mathrm{div}(\mathcal{X}[\mu(t)](\mu(t))).$$

Thus, we see that

$$= -\frac{1}{\beta}\iint_{(\mathbb{S}^{d-1})^2} e^{\beta\langle x,x'\rangle}\mathrm{div}_x(\mathcal{X}[\mu(t)](x))d\mu(t,x)d\mu(t,x') \tag{25}$$

$$= -\frac{1}{\beta}\iint_{(\mathbb{S}^{d-1})^2} e^{\beta\langle x,x'\rangle}\mathrm{div}_x(\mathcal{X}[\mu(t)](x))d\mu(t,x')d\mu(t,x). \tag{26}$$

By integration by parts, this simplifies to

$$\begin{aligned} &= \int_{\mathbb{S}^{d-1}} \mathcal{X}[\mu(t)](x)\cdot\int_{\mathbb{S}^{d-1}} \nabla_x\left(\beta^{-1}e^{\beta\langle x,x'\rangle}\right)d\mu(t,x')d\mu(t,x) \\ &= \int_{\mathbb{S}^{d-1}} \mathcal{X}[\mu(t)](x)\cdot\int_{\mathbb{S}^{d-1}} e^{\beta\langle x,x'\rangle}x'd\mu(t,x')d\mu(t,x). \end{aligned} \tag{27}$$

The interior integral is exactly the flow in equation (17) but multiplied by the partition function at x. Thus, this reveals the positivity of the time derivative:

$$\frac{d}{dt}E_\beta[\mu(t)] = \int_{\mathbb{S}^{d-1}} \|\mathcal{X}[\mu(t)](x)\|^2 Z_{\beta,\mu(t)}(x)d\mu(t,x). \tag{28}$$

Since $Z_{\beta,\mu(t)}(x) \geq 0$, we conclude that $\frac{d}{dt}E_\beta[\mu(t)] \geq 0$ for all $t \geq 0$. □

Remark 2. *When $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ is an empirical measure, the interaction energy simplifies to the following quantity:*

$$E_\beta[\mu] = \frac{1}{2\beta n^2}\sum_{i,j=1}^n e^{\beta\langle x_i,x_j\rangle}. \tag{29}$$

We notice that this quantity has experimental value other than being a theoretically monotonic quantity on this particular flow: we verify that in real transformers it remains monotonic and is correlated with phases of training in Section 5.4.

3.2 Interpretation of Dynamics as Wasserstein Gradient Flow

Because interaction energy is monotonic, we are interested in studying whether the dynamics (18) is a Wasserstein gradient flow of the interaction energy. In other words, when does it hold that $\mathcal{X}[\mu] = \nabla E_\beta[\mu]$? This is false in general, but under slightly modified flows it is true. In particular:

1. A simple method is to replace the partition function. If we take the vector field $\mathcal{X}[\mu]$ to be defined alternatively by

$$\mathcal{X}[\mu](x) = \boldsymbol{P}^{\perp}_{x(t)} \left(\int e^{\beta \langle x, x' \rangle} x' \, d\mu(x') \right) \tag{30}$$

then one has

$$\mathcal{X}[\mu](x) = \nabla \delta E_{\beta}[\mu](x) \tag{31}$$

where in this case the gradient is taken with respect to the standard Riemannian metric on $\left(\mathbb{S}^{d-1}\right)^n$, which we recall is i copies of the projection of the Euclidean gradient onto the tangent space of the sphere (i.e. via $\boldsymbol{P}$). The argument is quite straightforward in the empirical measure case.

Proof. When $\mu(t)$ is a discrete empirical measure, we can write it as

$$\mu(t) = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i(t)} \tag{32}$$

and the dynamics of $\mathbf{x}$ are given by

$$\dot{x}_i(t) = \boldsymbol{P}^{\perp}_{x_i(t)} \left(\frac{1}{n} \sum_{i=1}^{n} e^{\beta \langle x_i(t), x_j(t) \rangle} x_j(t) \right). \tag{33}$$

In spirit, we have simplified the flow by replacing the partition function with just $\frac{1}{n}$. The interaction energy is also simple in this form and given in equation 29. Now, we can directly compute ∂_i (gradient with respect to ith particle lying on the ith sphere) as

$$\partial_i E_{\beta}(\mathbf{x}) = \frac{1}{2\beta n^2} \left(\sum_{j=1}^{n} \partial_i e^{\beta \langle x_i(t), x_j(t) \rangle} + \sum_{j=1}^{n} \partial_i e^{\beta \langle x_j(t), x_i(t) \rangle} \right) \tag{34}$$

$$= \frac{1}{\beta n^2} \sum_{j=1}^{n} \boldsymbol{P}^{\perp}_{x_i(t)} \left(\beta e^{\beta \langle x_i(t), x_j(t) \rangle} x_j(t) \right). \tag{35}$$

Using equation (33), we see that the RHS is exactly $\frac{1}{n} \dot{x}_i(t)$, giving the Wasserstein gradient flow. □

Thus, the dynamics equation (18) with the modified vector field $\mathcal{X}[\mu]$ with the replaced partition function is a Wasserstein gradient flow.

2. A more subtle modification is to alter the metric on the sphere in a specific case so that the original dynamics (18) is a gradient flow. In particular, begin by assuming that $Q^T K$ is symmetric and $V = Q^T K$. For $X = (x_1, x_2 \ldots, x_n) \in (\mathbb{S}^{d-1})^n$, define a new metric on $T_X(\mathbb{S}^{d-1})^n$ by

$$\langle (a_1, \ldots, a_n), (b_1, \ldots, b_n) \rangle_X = \sum_{i=1}^{n} Z_{\beta,i}(X) \langle a_i, b_i \rangle,$$

where $a_i, b_i \in T_{x_i}\mathbb{S}^{d-1}$ and the partition function becomes

$$Z_{\beta,i}(X) = \sum_{k=1}^{n} e^{\beta\langle Vx_i, x_j\rangle}.$$

This inner product is clearly properly defined, as it is (1) linear, (2) real valued and symmetric, and (3) always positive because $Z_{\beta,i}$ is always positive. Under this metric, the dynamics are also a Wasserstein gradient flow (we refer readers interested in the proof to Geshkovski et al. [2023b], Section 3.4.1).

4 Clustering

There exists a wide variety of long-time limit results, but they all share the same flavor: as $t \to \infty$, the states $\mathbf{x}$ almost surely exhibit some predictable behavior in $\left(\mathbb{S}^{d-1}\right)^n$. We explore a few different versions of this and analyze the key assumptions underlying all. We will then address the assumptions and attempt to extend results beyond them (such as considering non identity value matrices) via numerical simulation.

Single Point Clustering. The simplest phenomenon is a collapse of all points to a single point $x^* \in \mathbb{S}^{d-1}$. The results in this area are plentiful when $V = I$.

Definition 4.1 (Single-Point Clustering)**.** *A flow $\mathcal{F} = \dot{\mathbf{x}}$ **single-point clusters** in the limit $t \to \infty$ iff for Lebesgue almost any initial conditions $\mathcal{I} = \{x_i(0)\}_{i=1}^n$, there exists a point x^* such that the solution $\{x_i(\cdot)\}_{i=1}^n$ to $\mathcal{F}$ with initial conditions $\mathcal{I}$ satisfies*

$$\lim_{t\to\infty} x_i(t) = x^* \tag{36}$$

for all $1 \le i \le n$.

It turns out that in the high-dimensional case the results are simpler than when $d = 2$.

Theorem 4.2 (Various Cases of SPC, Geshkovski et al. [2023b])**.** *If any of the following is true:*

1. *$d \ge 3$, $n \ge 2$, $\beta \ge 0$,*
2. *$d = 2$, $\beta \le \max(1, Cn^{-1})$ for specific C,*
3. *$d = 2$, $\beta \ge n^2/\pi^2$ for different C,*

then it follows that the flow in 15 single-point clusters when $Q = K = V = I$. If $\beta \ll 1$, the flow also single-point clusters for Q, K arbitrary and $V = I$ in all dimensions.

Notably, there is no result when $d = 2$ and $\beta \in (1, n^2/\pi^2)$ nor when $V \neq I$. We will explore this numerically.

Remark 3 (Cone Collapses Exponentially)**.** *It can be shown that for any d, arbitrary Q, K and $V = I$, if there exists $w \in \mathbb{S}^{d-1}$ such that $\langle w, x_i(0)\rangle > 0$ for all i, then the points converge exponentially to x^*.*

Recall that when $d \ge n$, since a set of n points cannot form a d dimensional convex hull, for uniformly randomly selected $x_i(0)$ there will almost surely exist a w that satisfies the above property. Thus, when $d \ge n$, we have almost sure exponential convergence to x^*.

Causal Dynamics. The only main result in causal dynamics comes from Karagodin et al. [2024] and considers it in the extended case that the dynamics are causal i.e. $x_i(t)$ only depends on the states of $x_1(t), x_2(t), \ldots, x_{i-1}(t)$. The main application of causal dynamics is autoregressive language modeling.

Theorem 4.3 (Single Point Clustering in Causal Models, Karagodin et al. [2024])**.** *Assume $V = I_d$ and Q, K arbitrary but constant. Then, for any i,*

$$\lim_{t\to\infty} x_i(t) = x_1(0). \tag{37}$$

In other words, all points cluster around the initial value of the first particle.

In particular, the causal dynamics let us specify the *location* of x^* - around the initial value of the first point. An important part of this argument comes from noticing the dynamics of the first token (since it can only attend to itself) under the $V = I$ assumption are trivial:

$$\dot{x}_1(t) = \mathbf{P}^{\perp}_{x_1(t)}(x_1(t)) = 0. \tag{38}$$

However, clearly these assumptions are not actually practical. Often in causal models $x_1(0)$ is the `<bos>` or beginning of sequence token which is appended to the start of every sequence but never appears in an actual context. If all attention heads in a transformer model satisfied these assumptions, then all predicted tokens would be the `<bos>` token, which would be a terrible language model!

Other Forms of Clustering. When layer normalization is omitted, other forms of clustering have been observed. Because this model is not that indicative of how transformers are used in practice, we only briefly skim the results and direct interested readers to Geshkovski et al. [2023a].

Theorem 4.4 (Convergence without LayerNorm, Geshkovski et al. [2023a])**.** *Consider the flow*

$$\dot{x}_i(t) = \frac{1}{Z_{\beta,i}} \sum_{j=1}^{n} e^{\beta\langle Qx_i(t), Kx_j(t)\rangle} V x_j(t). \tag{39}$$

Then, the following convergence behaviors have been shown under positivity results:

1. *$V = I$, $Q^T K > 0$: vertices of a convex polytope,*
2. *V's largest eigenvalue is real and positive, $\langle Q\varphi_1, K\varphi_1\rangle > 0$ for all $\varphi_1 \in span(v_1)$, where v_1 is the first eigenvector of V: 3 parallel hyperplanes,*
3. *V is paranormal (Definition 5.1, Geshkovski et al. [2023a]) and $Q^T K > 0$: union of polytopes and subspaces.*

In general, these positivity assumptions are not at all guaranteed to hold. The lack of consideration of layer normalization further weakens the applicability of these results. However, there is an interesting lemma in the $d = 1$ case used to prove these results.

Lemma 4.5 (Attention Pattern Becomes Low Rank, Geshkovski et al. [2023a])**.** *Let $d = 1$. Define the **attention pattern** as the matrix*

$$P_{ij}(t) = \frac{\sum_{j=1}^{n} e^{\beta\langle x_i(t), x_j(t)\rangle}}{Z_{\beta,i}}. \tag{40}$$

so that the unnormalized flow is

$$\dot{x}_i(t) = \sum_{j=1}^{n} P_{ij}(t) V x_j(t). \tag{41}$$

Then, if scalars $V > 0$ and $QK > 0$,

$$\lim_{t\to\infty} P(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ & & \vdots & & \\ \nu_1 & \nu_2 & \cdots & \nu_{n-1} & \nu_n \\ 0 & 0 & \cdots & 0 & 1 \\ & & \vdots & & \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \tag{42}$$

where $\sum_{i=1}^{n} \nu_i = 1$ i.e. a low rank boolean matrix.

The reason why this result is so interesting despite its limited scope is that in practice empirical researchers do indeed study the P matrix of attention heads - this proves to be a valuable method to *interpret* the behavior of various weights in the models, especially causal language models cite. Notice that $P_{ij} = 0$ if $j > i$ in causal models due to the masking. A result similar to this in a more general case would be of great interest to many practitioners and is an open problem.

For example, the following forms of heads have been discovered in real models by analyzing the attention patterns:

1. **Previous Token Heads** - attention heads where the attention pattern is always approximately

$$P_{ij} \approx \begin{cases} 1 & i = j + 1 \\ 0 & \text{otherwise} \end{cases}, \tag{43}$$

   i.e. $x_i(t)$ is only influenced by $Vx_j(t)$ if $x_j(t)$ is the token directly before x_i.

2. **Induction Heads** - indicated by attention patterns that place high attention on the token directly following the last occurrence of the current token.

These heads have also been found to directly *compose* across t. As a concrete example, induction heads are reliant on previous token heads existing at a previous layer. Together, they form the induction circuit: previous token heads copy information across the input sequence, while induction heads read from this information to identify the last occurrence of the current token and copy the token directly after it. This mechanism is responsible for predicting the token `b` as the continuation of the sequence

`a b c a b c a ...`

and is fundamental to the emergence of in-context learning, or a language model's ability to perform unseen tasks after just a few examples provided in the prompt.

4.1 Metastability

While single point clustering is interesting theoretically, it is not actually that interesting in practice as it could be interpreted as a lack of expressivity in the infinite depth limit. However, a form of *metastability* has also been noted to appear in these systems: under certain initial conditions, particles cluster together into $1 < k < n$ clusters for a "long" time before eventually collapsing into 1 cluster.

Let

$$\mathcal{S}(x, \varepsilon) = \{y \in \mathbb{S}^{d-1} : \langle y, x \rangle \geq 1 - \varepsilon\} \tag{44}$$

be the spherical cap centered at x.

Definition 4.6 ((β, ϵ) Separated, Geshkovski et al. [2024a])**.** *Let $d, n \geq 2$, $\beta > 1$, and $\varepsilon \in \left(0, \frac{1}{16}\right)$. $\{x_i\}_{i=1}^n$ is (β, ϵ) separated if there exists $k \leq n$ points $\{w_q\}_{q=1}^k \subseteq \mathbb{S}^{d-1}$ such that*

1. *(Covering) $x_i(0) \in \bigcup_{q=1}^k \mathcal{S}(w_q, \varepsilon)$ for all i.*
2. *(w_q Separated) $\gamma(\beta) = 1 - \alpha - C(\varepsilon, \beta) > 0$ and $\gamma(\beta) = \Omega(1)$ where*

$$\alpha = \max_{(x,y) \in \mathcal{S}(w_i, \varepsilon) \times \mathcal{S}(w_j, \varepsilon), i \neq j} \langle x, y \rangle. \tag{45}$$

Theorem 4.7 (Metastability)**.** *Assume $\{x_i(0)\}_{i=1}^n$ is (β, ε) separated. Then, there exists $0 < T_1 < T_2$ such that the solution to equation (15) ($Q = K = V = I$) satisfies*

1. *x_i remain in the initial spherical caps up to T_2: if $x_i(0) \in \mathcal{S}(w_q, \varepsilon)$, then $x_i(t) \in \mathcal{S}(w_q, 2\varepsilon)$ for $t \leq T_2$.*
2. *x_i are exponentially close within caps between $[T_1, T_2]$.*

Furthermore, T_2 is exponentially large in terms of $1 - \alpha$.

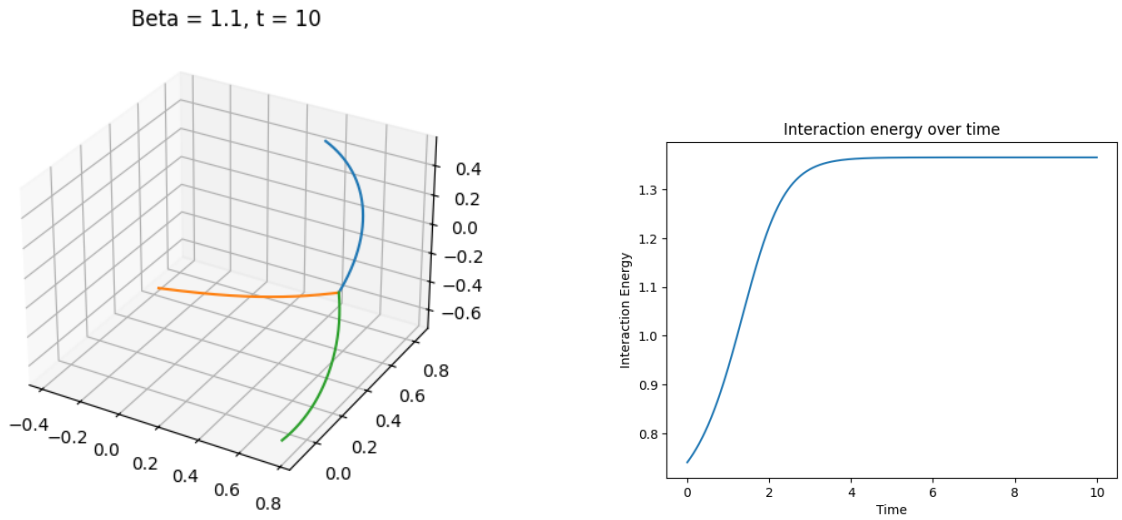
We have omitted many constants and technicalities in the presentation of the results, but the intuition is present: after T_1 and up to T_2, the points are exponentially close in k spherical caps *separated* around the sphere, which is exactly metastability. We numerically verify the existence of this phenomenon in Section 5.1.

5 Numerical Experiments

We can simulate the dynamics all versions of the above flows through a simple first-order Euler scheme. Specifically, given a step size Δt, we compute

$$x_i(t + \Delta t) = x_i(t) + f(t, \mathbf{x}(t); \theta(t)), \tag{46}$$

where f is given by the transformer parameterized by $\theta(t)$ and depends on the location of all other tokens $\mathbf{x}(t)$. Due to numerical precision, even if we project f onto the tangent space at $x_i(t)$, we still observe deviations as the step size increases from $\mathbb{S}^{d-1}$. To correct for this we simply normalize to $\mathbb{S}^{d-1}$ at every step. Likewise, unless otherwise mentioned, we select our initial tokens uniformly on $\mathbb{S}^{d-1}$ by sampling a Gaussian and then normalizing so the vectors are unit norm.



(a) Example trajectory for 3 tokens in $d = 3$.

(b) Interaction energy over time for $d = 3$.

Figure 1: (a) Three particles following the dynamics in (15) converge into a single cluster. (b) Interaction energy is monotone increasing over time for the dynamics shown in figure 1a.

For estimating quantities such as interaction energy (Equation (23)) that depend on the measure μ, we estimate their evolution through Monte Carlo simulation. We sample numerous initial conditions, evolve the flow, and calculate the empirical average at various time steps. For example, formally, with N simulations we estimate $E_\beta[\mu(t)]$ as

$$E_\beta[\mu(t)] \approx \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2\beta n^2}\sum_{i,j=1}^{n}\exp\left(\beta\left\langle x_i^{(k)}(t), x_j^{(k)}(t)\right\rangle\right), \tag{47}$$

where $x_i^{(k)}(t)$ is the value of the ith particle at time t starting from the kth sampled initial condition.

5.1 Basic Replication and Extensions

We begin by verifying existing theoretical results. In particular we simulated the dynamics in equation (15) with $Q = K = V = I$ for various initial conditions and temperatures and consistently observed the clustering behavior described in Geshkovski et al. [2023b] - see Figure 1a for an example in $d = 3$ with temperature $\beta = 1.1$. Furthermore, estimating the interaction energy using equation 47 verifies Lemma 3.3, confirming that the interaction energy is also monotonically increasing in this simple flow (Figure 1b). We also observe the *metastability* phenomenon described in Geshkovski et al. [2024a]: particles tend to quickly form into a few clusters and stay for a very long time before collapsing into a single cluster.

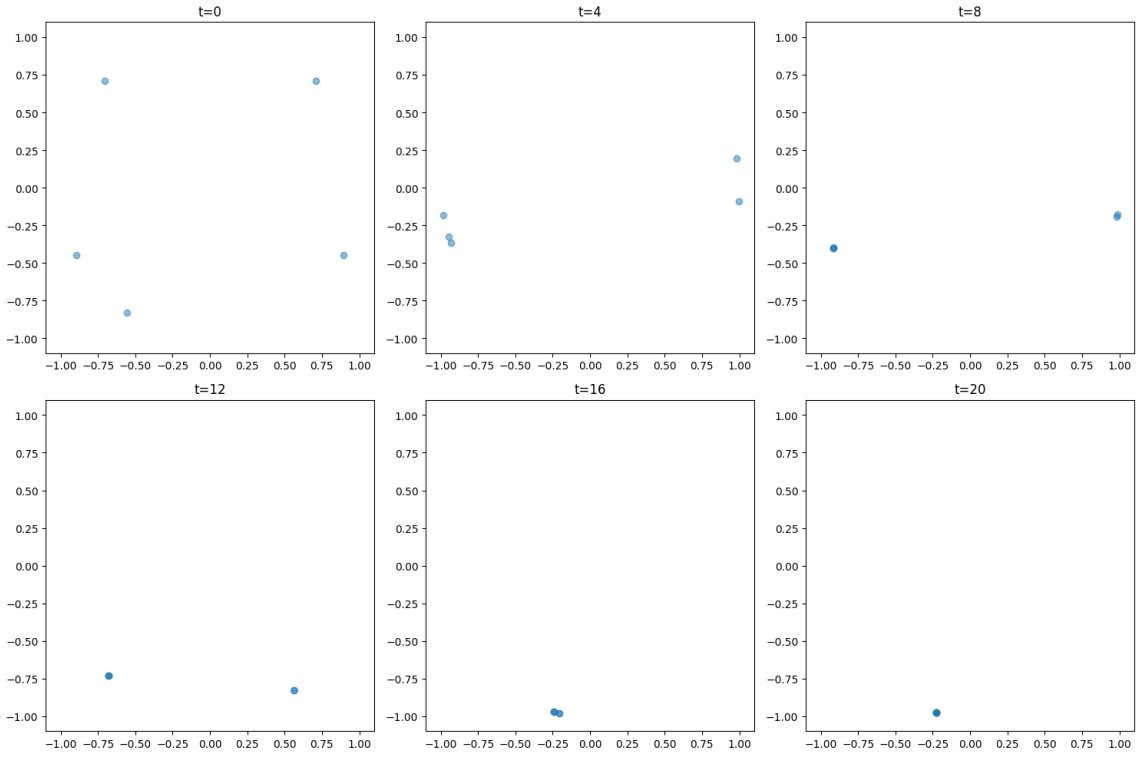


Figure 2: The particles converge into two clusters and stay for a long period of time before the two clusters eventually merge.

5.2 Initial Experiments in Time Dependence.

We also find evidence that Q and K can be fully random at all times - as long as V is the identity clustering is always exhibited (Figure 3).

Non Identity V - Bouncing. As a result of the time dependence, we can construct Q, K, and V matrices that push tokens into unusual behaviors. As an illustrative example, assume we choose Q and K random, but they change at $t = 5$. V will be chosen so that it "flips" between attraction and repulsion on the particles at $t = 5$ as well:

$$V(t) = \begin{cases} \text{diag}(1, \ldots, 1, -1, \ldots, -1) & 0 \leq t \leq 5 \\ -V(0) & t > 5. \end{cases} \tag{48}$$

We plot the dynamics of particles in Figure 4, visualizing by projecting the number of dimensions down using the first two components of PCA. As expected, the flip in the value matrix results in a unique particle trajectory where the points first cluster in one area, then "bounce" and begin clustering in a different region.

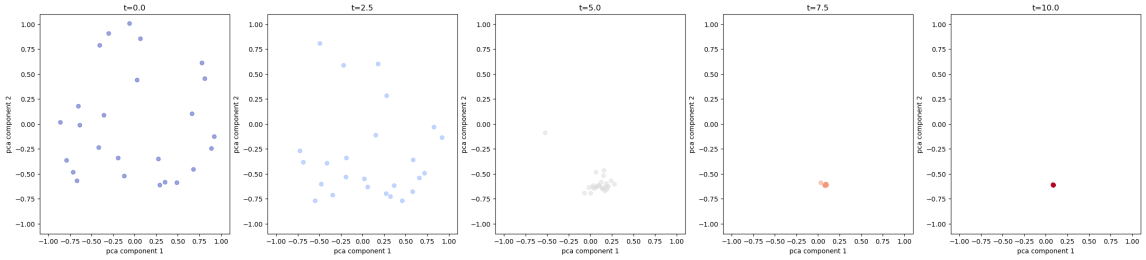


Figure 3: Random Q, K matrices still cluster as long as $V = I$. $d = 4$ with $n = 25$ tokens.

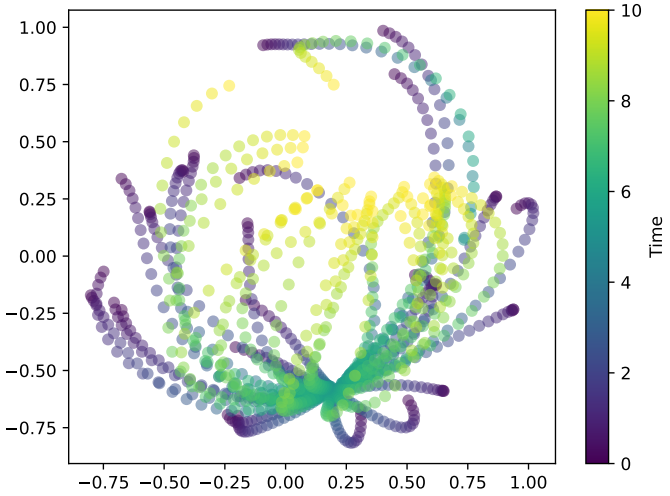


Figure 4: Example of "bouncing" resulting from parameters changing at $t = 5$. The points are evolved in $d = 5$ but visualized along the first two principal components fitted on the initialization data. The points begin spread out across the sphere, cluster together from $t = 0$ to $t = 5$, then repel rapidly and begin clustering to another point.

5.2.1 Reverse Engineering a Simple Algorithm

In this section, we reverse engineer a simple algorithm that predicts the next token based on whether a specified token exists in the sentence. All tokens use one-hot token embeddings. The composition steps are simple for our algorithm:

1. Attend to token 0 if it exists in the sequence. If it does, push the current token (either 1 or 2) to the embedding for 0. If it doesn't, leave the current token alone.

2. Generate the result: push token 0 to 3 (since that means 0 was in the sequence) and push token 1 and 2 to 4 (0 was not in the sequence).

These operations can be encoded quite simply by the following parameterization of matrices. Let 1_{ij} be the indicator matrix on the ijth element of a matrix.

$$Q = K = 0 \text{ for all } t, \tag{49}$$

$$V = \begin{cases} 1_{00} & t < 5, \\ 1_{30} + 1_{41} + 1_{42} & t \geq 5. \end{cases} \tag{50}$$

The Q and K being 0 results in uniform attention being placed on all tokens. V's first stage promotes the 0 token if it is spotted in the sequence. The second stage aligns the representations with the corresponding tokens. In Figure 5, we map each point on the sphere to a probability distribution over tokens (using a softmax and with temperature $\beta = 2$) and plot the induced distribution for each token in a causal setting. We can see that in stage 1, the distribution flips at the moment token 0 is seen in the input sequence, marking the end of stage 1. At the end of stage 2, we then see a similar plot, showing that the moment 0 appears in the input sequence the distribution shifts weight onto token 3. This is an example of how the causal dynamics system can cluster to two different clusters in a nontrivial way when $V \neq I$.

Token Index	**Purpose**
0	Special Token
1, 2	Filler Tokens
3	YES (0 in sequence)
4	NO (0 not in sequence)

Table 2: Token roles in simple toy setup from §5.2.1.

We also attempted to reverse engineer other compositions for simple tasks, such as an previous token head and induction head in order to form an induction circuit, as discussed in §4. However, reverse engineering continuous time weights proved to pose a significant difficulty, especially when positional encodings are introduced.

We attempted to use a very simple one-hot positional encoding, resulting in an embedding vector of shape $\mathbb{R}^{d+n}$ where n is the fixed input sequence length. Anything more complex (such as sinusoidal or rotary positional encodings) would be nearly impossible to manually encode. In this setting, we are able to approximately recover a previous token head. Suppose we take 8 tokens and sequence length 24. Then, at $t = 1.5$, if we take the argmax component for each token (i.e. which token it has highest cosine simliarity to), we see that 19 out of the possible 22 tokens (ignoring end

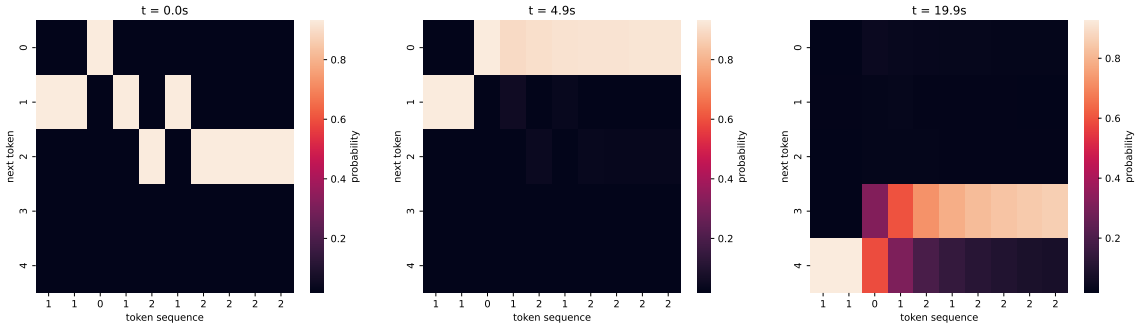


Figure 5: Results of reverse engineering the token detection algorithm using time-dependent matrices. Step 1 (detection) ends at $t = 5$. Step 2 (prediction) continues until $t = 20$. We can see that because of the causal masking, 0 is only detected in the sequence *after* it appears in the third token position. How strongly confident our simple model is that the 0 has appeared increases as the sequence extends in length. The probabilities are generated by taking the embedding weights and applying a softmax with temperature 2.

tokens) have the most weight in their embedding on the previous token (Figure 6). However, we find the second half of the circuit much noisier and we are unable to generate the desired induction behavior.

The hardest difficulty comes as a direct result of the continuous time dynamics. Suppose the state is currently $\mathbf{x}$ and we would like to end up at $\mathbf{c}$. In discrete time dynamics, we just have one timestep, which means that a desired behavior can be induced by taking one large step in the direction of the desired token and then normalizing, because

$$\mathbf{x} + \mathbf{P}_{\mathbf{x}}^{\perp}(\mathbf{c}) \approx \mathbf{c} \tag{51}$$

if $\|\mathbf{c}\| \gg 1$. However, the integration and identification of the tangent vector at each timestep that would lead to approximate ending up at $\mathbf{c}$ is much more difficult because there are multiple particles that, on their continuous time paths to $\mathbf{c}$, can incur unintended interaction effects, leading particles to deviate from the intended direction.

5.3 Flow Matching Expressivity

In this section, we consider using flow matching to approximate the transformer flow. Taking tokens in $\mathbb{R}^{N\times d}$ sampled from some initial distribution μ and then applying the transformer flow on it, we get vectors in $\mathbb{R}^{N\times d}$ with some distribution ν. Given the samples $X_0 \sim \mu$ and $X_1 \sim \nu$, we want to learn a vector field $v : \mathbb{R}^{N\times d} \times \mathbb{R}_+ \to \mathbb{R}^{N\times d}$ such that the flow $\rho_t : \mathbb{R}^{N\times d} \times \mathbb{R}_+ \to \mathbb{R}$ that satisfies the continuity equation

$$\partial_t \rho_t + \operatorname{div}(\rho_t v) = 0$$

transports μ to ν. In contrast to the continuity equation for the the transformer flow given in (19), the vector field v we seek does not depend on the flow ρ_t. Indeed, our objective is determine to what extend flow matching can approximate the transformer flow with a simpler flow.

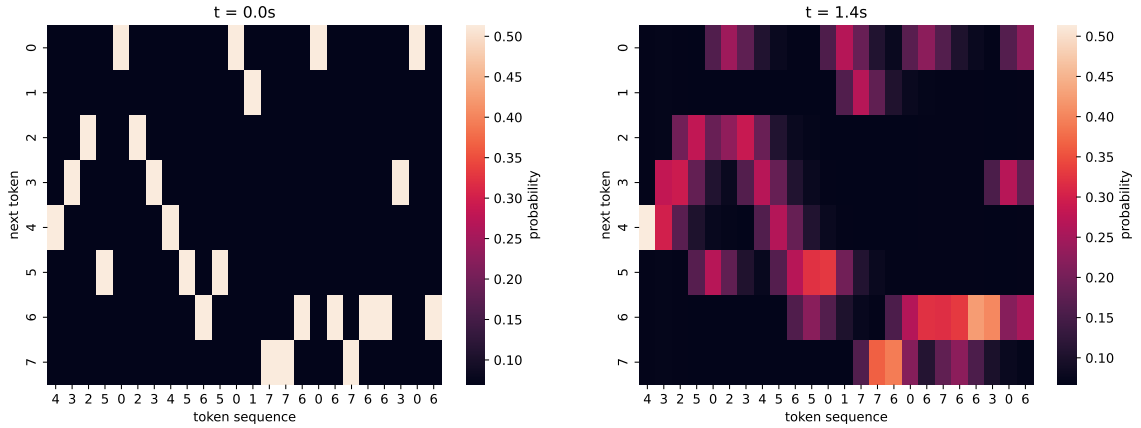


Figure 6: Example of approximate previous token head using simple one hot positional encodings. The values of the tokens (first d dimensions) are ignored and attention is just paid to the token directly after a token in the second d dimensions. 19/22 possible tokens have highest weight on the previous token in a randomly generated input sequence.

Recall that the objective of flow matching is to solve the following optimization problem where v is taken over the set of vector fields $\mathbb{R}^{N\times d} \times \mathbb{R}_+ \to \mathbb{R}^{N\times d}$

$$\min_v \mathbb{E}_{(X_0,X_1)\sim\mu\otimes\nu} \left[|X_1 - X_0 - v((1-t)X_0 + tX_1)|^2\right].$$

Following the original flow matching paper Liu et al. [2022], we parameterized v with a two-layer neural net and approximated the expectation with empirical samples from μ and ν. For our experiment, each sample of μ consists of four two-dimensional tokens. The tokens are sampled independently from each other from a Gaussian with a different mean. Note that we did not normalize the tokens to be on the unite circle. As Figure 7 suggests, flow matching did a good job of approx-

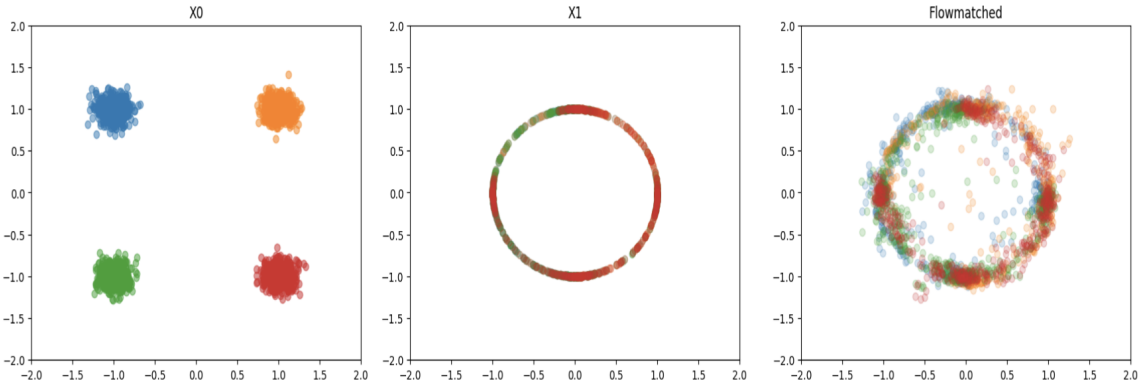


Figure 7: 500 samples from μ superimposed (left). The samples after flowed through the transformer flow (middle). The samples flowed through the flow that is learned by flow matching (right).

imating the transformer flow. We also compared the trajectories that the tokens took under the transformer flow versus the learned flow. In figure 8, we see that, under the learned flow, the tokens

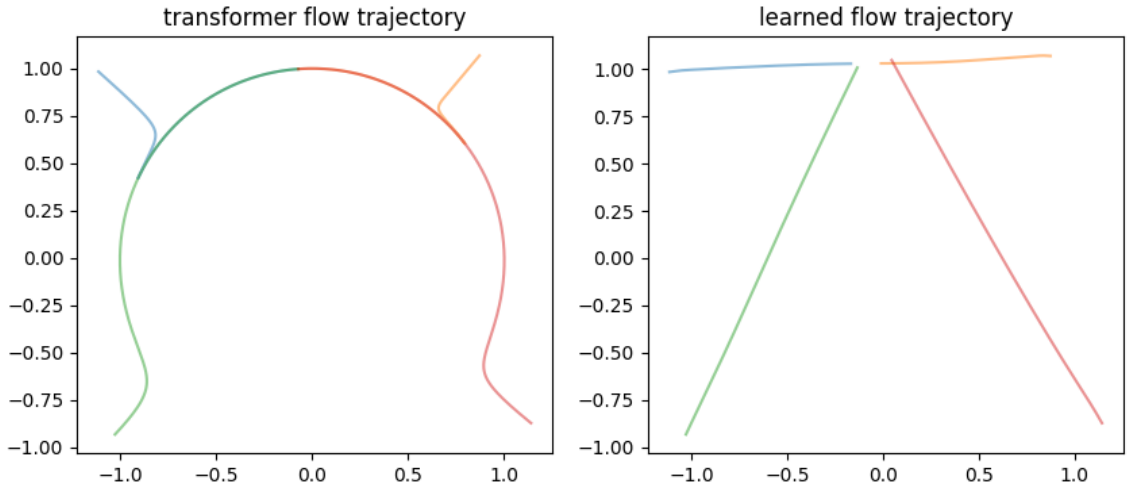


Figure 8: Trajectory of a sample under the transformer flow(left). Trajectory of a sample under the learned flow(right)

take a straight line path towards the final, clustered position whereas, under the transformer flow, the tokens moved along the unit circle. This behavior aligns with our intuition for flow matching, which approximate flows with straight line flows.

5.4 Real Models

In this section, we verify the clustering and interaction energy monotonicity results on real models. In particular, we utilize the Pythia suite of models [Biderman et al., 2023].

In order to normalize to work with outputs that are not necessarily on $\mathbb{S}^{d-1}$, instead of computing the interaction energy as in 3.2, we instead compute

$$\iint_{\mathbb{R}^d \times \mathbb{R}^d} e^{\frac{\langle x, x' \rangle}{\|x\| \|x'\|}} d\mu(x) d\mu(x'). \tag{52}$$

This ensures that the exponent power remains between 0 and 1. Notice that if $x, x' \in \mathbb{S}^{d-1}$, this is exactly proportional to the interaction energy defined in Definition 3.2.

Phases of Training. The Pythia model suite offers checkpoints at various points across training - using these checkpoints alongside the interaction energy allows us to visualize a short story about the training dynamics of a relatively small (70m) autoregressive language model.

We select interspersed checkpoints with more at the beginning of training (roughly log scale). At each selected training checkpoint, we estimate the interaction energy of the outputs of the model over a sample of tokens. We also compute the KL divergence of the distribution over next tokens induced by the model's outputs to an empirical unigram distribution (fitted on a sample of OpenWebText, a large web text corpus).

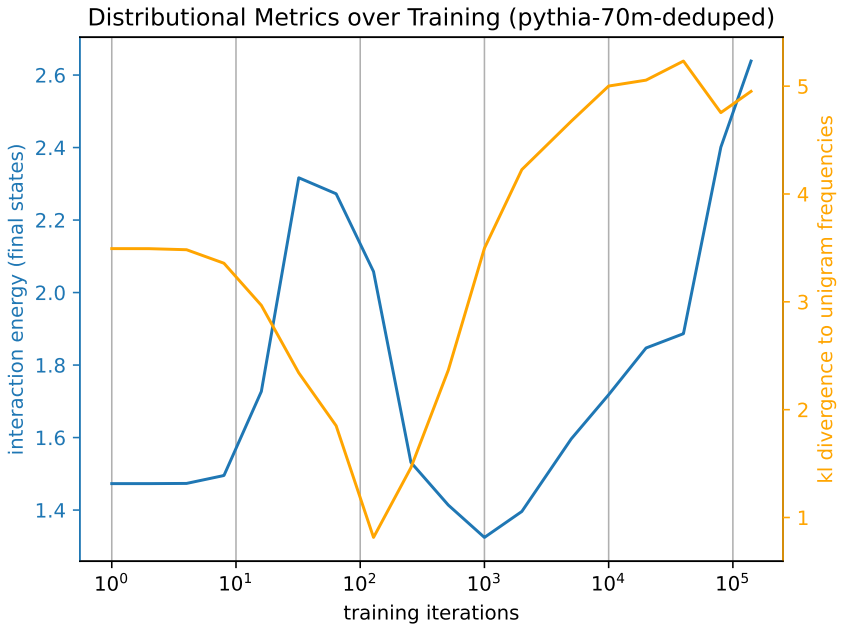


Figure 9: Distributional metrics over log-training iterations in Pythia-70m-Deduped model. We observe simple distinct training phases depending on what the model is causally predicting throughout training.

In Figure 9 we plot these metrics and notice a distinct pattern that tells a story about the training dynamics. Interaction energy spikes once early in training and increases again later on while the KL divergence to the unigram distribution dips exactly when the interaction energy spikes the first time but does not decrease again the second time interaction energy rises. This result suggests that early on, the model learns the unigram distribution and begins outputting it / a similar distribution at every token. The interaction energy correlates this - the model's output at every token is very clustered (e.g. around the vector that induces the unigram distribution). However, soon after, the metric dynamics indicate a transition *away* from just predicting the unigram distribution. The model's predictions are still eventually high interaction energy, but the distance of the distributions induced by the outputs from the unigram distribution indicate the model has figured out a smarter method to organize concepts in embedding space.

6 Conclusions and Future Work

Findings. We distill the existing work on continuous time dynamics of attention that model the forward pass of a modern transformer. These existing works largely focus on the existence of long time limits to either metastable states or single point clusters. From these theoretical results, we construct rigorous numerical simulations that both verify and expand on the existing results, such as the correspondence between the transformer flow and the flow matching flow and reverse engineering specific results.

However, we quickly find the common assumptions in theoretical results such as time independence and identity matrices to be restrictive and impractical. Although mathematically interesting, we are only able to extract vaguely useful applications out of the theoretical applications derived from this particular continuous time perspective. For example, we find the interaction energy somewhat useful for identifying phases in training of a suite of causal language models. However, modern transformers do not necessarily exhibit single point clustering nor metastable clustering, so there must exist behaviors unexplained by the simple continuous time model presented here.

Future Work. As aforementioned, it remains to be seen what novel insights about transformers can be made from this particular continuous time perspective. It appears that explicit analysis is extremely difficult when Q, K, and V are not trivial matrices and varying in time. Numerical simulations are not particularly illuminating because they are equivalent to just directly simulating the forward pass of a transformer, albeit infinite depth. However, if anything, the clustering results demonstrate that transformers almost always don't have $V = I$, as otherwise all transformers would exhibit single point clustering (which is uninteresting). Understanding when transformers converge to matrices of this sort could potentially strengthen the applicability of these results.

The current model also omits many important characteristics of transformers. Both multi-head attention and multi-layer perceptrons play an important role in modern transformers. Understanding how the low-rank partitioning of the embedding space into individual head subspaces is likely important for understanding how their dynamics interleave over the time variable. Furthermore, multi-layer perceptrons are often regarded as the information storage units of transformers, so understanding how they interplay with the attention layers is crucial to understanding how transformers can retrieve (1) memorized and (2) in-context information.

7 Acknowledgements

We would like to thank Yair Shenfeld for his amazing mentorship throughout the project and the wonderful APMA2822B course. We would also like to thank Philippe Rigollet for the interesting talk he gave at Brown on this topic in Spring 2025.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pages 244–253. PMLR, 2018.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36: 57026–57037, 2023a.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers. *arXiv preprint arXiv:2312.10794*, 2023b.

Borjan Geshkovski, Hugo Koubbi, Yury Polyanskiy, and Philippe Rigollet. Dynamic metastability in the self-attention model. *arXiv preprint arXiv:2410.06833*, 2024a.

Borjan Geshkovski, Philippe Rigollet, and Domènec Ruiz-Balet. Measure-to-measure interpolation using transformers. *arXiv preprint arXiv:2411.04551*, 2024b.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

N. Karagodin, Y. Polyanksiy, and P. Rigollet. Clustering in causal attention masking. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL `https://arxiv.org/abs/2209.03003`.

Govind Menon. The geometry of the deep linear network. *arXiv preprint arXiv:2411.09004*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.