Iterated Vector Fields and Conservatism, with Applications to Federated Learning

Zachary Charles[*]
Google Research
zachcharles@google.com

Keith Rush[*]
Google Research
krush@google.com

Abstract

We study when iterated vector fields (vector fields composed with themselves) are conservative. We give explicit examples of vector fields for which this self-composition preserves conservatism. Notably, this includes gradient vector fields of loss functions associated to some generalized linear models (including non-convex functions). As we show, characterizing the set of smooth vector fields satisfying this condition yields non-trivial geometric questions. In the context of federated learning, we show that when clients have loss functions whose gradient satisfies this condition, federated averaging is equivalent to gradient descent on a surrogate loss function. We leverage this to derive novel convergence results for federated learning. By contrast, we demonstrate that when the client losses violate this property, federated averaging can yield behavior which is fundamentally distinct from centralized optimization. Finally, we discuss theoretical and practical questions our analytical framework raises for federated learning.

1 Introduction

In this work, we consider vector fields of the form $V : \mathbb{R}^n \to \mathbb{R}^n$. Recall that V is conservative if there is some function $f : \mathbb{R}^n \to \mathbb{R}$ such that $V = \nabla f$. We are interested in whether *iterated* vector fields (vector fields of the form $V \circ V \circ \cdots \circ V$) are conservative. This question has rich connections to a variety of areas, including differential geometry, dynamical systems, and optimization. As we will show, conservative iterated vector fields are particularly important for understanding optimization algorithms for federated learning.

Notation. Let $\mathcal{V}(\mathbb{R}^n, \mathbb{R}^m)$ denote the collection of functions from $\mathbb{R}^n$ to $\mathbb{R}^m$. We let $\mathcal{D}(\mathbb{R}^n, \mathbb{R}^m)$ denote the subset of differentiable functions, and $\mathcal{C}^k(\mathbb{R}^n, \mathbb{R}^m)$ denote the subset of $\mathcal{C}^k$ functions. If $m = n$, we abbreviate these by $\mathcal{V}(\mathbb{R}^n)$, $\mathcal{D}(\mathbb{R}^n)$ and $\mathcal{C}^k(\mathbb{R}^n)$. Throughout, $\|\cdot\|$ denotes the ℓ_2 norm on $\mathbb{R}^n$ with corresponding inner product $\langle \cdot, \cdot \rangle$. We let $I \in \mathcal{V}(\mathbb{R}^n)$ denote the identity map.

Given $V \in \mathcal{V}(\mathbb{R}^n)$, we use exponents to denote repeated iterations of V. That is, for $k \geq 1$ we define:

$$V^k(x) := \underbrace{V \circ V \circ \cdots \circ V}_{\text{k times}}(x)$$

By convention, for any $V \in \mathcal{V}(\mathbb{R}^n)$ we define $V^0 := I$.

Summary. Let $V \in \mathcal{V}(\mathbb{R}^n)$, and let k be a positive integer. We study the following question.

Question 1. *If V is conservative, is V^k also conservative?*

This leads to the following definition.

[*] Authors contributed equally to this work

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

Definition 1. *V is k-conservative if V^k is conservative. V is ∞-conservative if V^k is conservative for all $k \geq 1$.*

For convenience, we use "conservative" and "1-conservative" interchangeably. In a slight abuse of notation, we say that $\mathcal{A} \subseteq \mathcal{V}(\mathbb{R}^n)$ is k-conservative if for all $V \in \mathcal{A}$, V is k-conservative. In order to show that $\mathcal{A}$ is ∞-conservative, it suffices to show that $\mathcal{A}$ is conservative and closed under self-composition, as reflected in the following definition.

Definition 2. *$\mathcal{A} \subseteq \mathcal{V}(\mathbb{R}^n)$ is closed under self-composition if for all $V \in \mathcal{A}$ and $k \geq 1$, $V^k \in \mathcal{A}$.*

This leads us to the following specialization of Question 1.

Question 2. *Let $\mathcal{A} \subseteq \mathcal{V}(\mathbb{R}^n)$ be conservative. Is $\mathcal{A}$ closed under self-composition?*

Vector Fields and Optimization. Motivated by optimization, we will often consider vector fields of the form $V(x) = \nabla f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable. Given $\mathcal{F} \subseteq \mathcal{D}(\mathbb{R}^n, \mathbb{R})$, we define $\nabla \mathcal{F} = \{V \in \mathcal{V}(\mathbb{R}^n) : V = \nabla f, f \in \mathcal{F}\}$. For $\gamma \in \mathbb{R}$, we define $I - \gamma\nabla\mathcal{F} := \{I - \gamma\nabla f : f \in \mathcal{F}\}$. A recurring theme in this work is whether a set $I - \gamma\nabla\mathcal{F}$ is k-conservative. Such vector fields arise naturally in optimization, as gradient descent on a function f with learning rate γ corresponds to the discrete-time dynamical system given by $x_{t+1} = (I - \gamma\nabla f)(x_t)$.

Given an initial point x_0, the iterates of gradient descent then satisfy $x_k = V^k(x_0)$ where $V = I - \gamma\nabla f$. Therefore, if $I - \gamma\nabla f$ is ∞-conservative, then the k-th iterate of gradient descent is actually $\nabla h_k(x_0)$ for some function $h_k : \mathbb{R}^n \to \mathbb{R}$. In general, this viewpoint will allow us to understand the behavior of optimization algorithms by analyzing properties of the functions h_k.

2 Connections to Federated Learning

Questions about whether a vector field is k-conservative have important implications for federated learning, one noteworthy approach to which is *federated averaging* (FEDAVG) [9]. A slightly simplified version of FEDAVG operates as follows. Suppose we have clients $c = 1, 2, \dots, C$, each with loss function $f_c : \mathbb{R}^n \to \mathbb{R}$. At each round, the server broadcasts its model the clients. The clients perform k steps of gradient descent (with learning rate γ) on their loss functions, and send the resulting models to the server. The server updates its model as the average of these client models. Since communication from clients to the server is often a bottleneck [1, 9], this algorithm is often practical only when $k > 1$. When $k = 1$, this is equivalent to gradient descent with learning rate γ on the average of the client loss functions.

More formally, let $V_c := I - \gamma\nabla f_c$. At each round t, each client computes $V_c^k(x_t)$, and the server updates its model via $x_{t+1} = C^{-1}\sum_{c=1}^C V_c^k(x_t)$. This "operator-theoretic" view of FEDAVG has been previously used to leverage techniques from operator theory to analyze and design federated learning algorithms [7, 8, 12]. In order to allow the server to determine the magnitude of its update at each step, [13] introduces a "model delta" version of FEDAVG. This corresponds to the server update

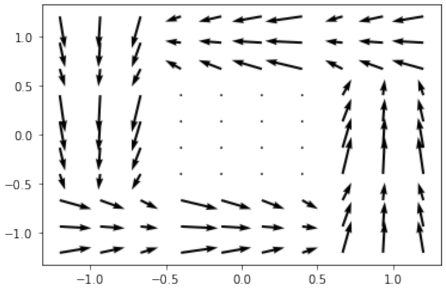
$$x_{t+1} = x_t - \frac{\eta}{C}\sum_{c=1}^{C}\left(x_t - V_c^k(x_t)\right) \tag{1}$$

where $\eta > 0$ is the server learning rate, which we may set to 1 to recover the average of the client models. In the sequel we let FEDAVG denote the update rule in (1). If we let V_s be the "server" vector field given by

$$V_s = \frac{1}{C}\sum_{c=1}^{C}(I - V_c^k) \tag{2}$$

then (1) is equivalent to $x_{t+1} = x_t - \eta V_s(x_t)$. This leads us to our guiding observation: If each V_c is k-conservative, then V_s is an average of conservative vector fields and is conservative as well. Therefore, there is some function f_s such that $\nabla f_s = V_s$, and the dynamics of FEDAVG are then equivalent to the dynamics of gradient descent on the function f_s (see Theorem 3 for a formal statement of this). This allows us to reduce the behavior of FEDAVG to the behavior of gradient descent on this "surrogate loss" f_s. Such an approach was used in [3] to understand the dynamics of FEDAVG and related methods on quadratic functions. In this work, we consider more general functions, including convex and non-convex functions.

Figure 1: Non-conservative server vector field V_s induced by f_1, f_2 in (3) for k sufficiently large.



2.1 Non-Conservative Dynamics in Federated Learning

As we sketched in the section above (and formalize in Section 5.1), when the vector fields $I - \gamma \nabla f_c$ are k-conservative, FEDAVG with k local steps behaves identically to gradient descent on some surrogate loss function. On the other hand, in this section we show that without k-conservatism, FEDAVG can demonstrate fundamentally non-conservative behavior, making its dynamics distinct from those of gradient descent. Notably, this can occur even when $C = 2$ and there is no stochasticity whatsoever. For example, for $c \in \{1, 2\}$, consider the client loss functions

$$f_c(x, y) := f_c^{(1)}(x, y) + f_c^{(2)}(x, y) \tag{3}$$

where

$$f_c^{(1)}(x, y) := \min\left(\frac{\alpha_c}{2}(y - y_c)^2 + \frac{\beta_c}{2}(x - x_c)^2, 1\right),$$
$$f_c^{(2)}(x, y) := \min\left(\frac{\alpha_c}{2}(y + y_c)^2 + \frac{\beta_c}{2}(x + x_c)^2, 1\right).$$

Notably, $I - \gamma \nabla f_c$ may not be k-conservative for $k > 1$. As we show in Appendix C, for some choice of $\alpha_c, \beta_c \in \mathbb{R}$, $x_c, y_c \in \mathbb{R}^2$ (for $c = 1, 2$), $\gamma > 0$ and k sufficiently large, the resulting server vector field V_s in (2) is non-conservative.

To illustrate this, in Fig. 1 we plot this server vector field V_s. Running FEDAVG with appropriate initialization over the 2-client setup in Appendix C follows the vector field V_s, and cycles endlessly (see Fig. 1 in Appendix C). In short, FEDAVG may behave badly in the absence of k-conservatism.

3 Examples of k-Conservative Vector Fields

We now give concrete examples of k-conservative vector fields. These include vector fields associated to linear and logistic regression. Let $\mathcal{P}_d(\mathbb{R}^n, \mathbb{R}^m)$ denote the subset of $\mathcal{V}(\mathbb{R}^n, \mathbb{R}^m)$ whose coordinate functions are homogeneous polynomials of degree d. We abbreviate this as $\mathcal{P}_d(\mathbb{R}^n)$ when $n = m$.

Constant Vector Fields. The space $\mathcal{P}_0(\mathbb{R}^n)$ of constant vector fields is clearly closed under self-composition. Constant vector fields are conservative, so $\mathcal{P}_0(\mathbb{R}^n)$ is ∞-conservative.

Affine Vector Fields. Let $\mathcal{A}(\mathbb{R}^n)$ be the set of affine vector fields in $\mathcal{V}(\mathbb{R}^n)$. This consists of all V of the form $V(x) = Ax + b$ for $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Let $\mathcal{S}(\mathbb{R}^n)$ denote the set of such V where A is symmetric. If $V \in \mathcal{A}(\mathbb{R}^n)$ is conservative, it is the gradient of some quadratic function. Simple algebraic manipulation then implies that V is conservative iff A is symmetric. Since $\mathcal{S}(\mathbb{R}^n)$ is closed under self-composition, $\mathcal{S}(\mathbb{R}^n)$ is ∞-conservative while $\mathcal{A}(\mathbb{R}^n)$ is not conservative. In particular, if f is a quadratic function, ∇f and $I - \gamma \nabla f$ are both ∞-conservative.

Continous Univariate Functions. Consider the set $\mathcal{C}^0(\mathbb{R})$ of continuous functions from $\mathbb{R}$ to $\mathbb{R}$. By elementary analysis, $\mathcal{C}^0(\mathbb{R})$ is closed under self-composition, and by the fundamental theorem of calculus, it is conservative. Thus, $\mathcal{C}^0(\mathbb{R})$ is ∞-conservative.

More generally, let $\mathcal{C}^0(\mathbb{R})^n$ denote the subset of $\mathcal{V}(R^n)$ containing vector fields of the form

$$V(x_1, \ldots, x_n) = (f_1(x_1), f_2(x_2), \ldots, f_n(x_n))$$

where $f_1, \dots, f_n \in \mathcal{C}^0(\mathbb{R})$. Then note that $V(x_1, \dots, x_n) = \nabla \left(\sum_{i=1}^n \int_0^{x_i} f_i(t)dt \right)$ so $\mathcal{C}^0(\mathbb{R})^n$ is conservative. Since $\mathcal{C}^0(\mathbb{R})^n$ is closed under self-composition, it is also ∞-conservative.

Non-example: Cubic Polynomials. Let $f(x, y) = x^2y$. By direct computation,

$$(\nabla f)^2(x, y) = \begin{pmatrix} 4x^3y \\ 4x^2y^2 \end{pmatrix} =: \begin{pmatrix} h_1(x, y) \\ h_2(x, y) \end{pmatrix}.$$

We then have $\frac{\partial}{\partial y} h_1(x, y) = 4x^3$, $\frac{\partial}{\partial x} h_2(x, y) = 8xy^2$. By Clairaut's theorem (see [14, Chapter 4]), $(\nabla f)^2$ is not conservative. Thus, $\nabla \mathcal{P}_3(\mathbb{R}^2, \mathbb{R})$ is conservative but not 2-conservative.

3.1 Gradient Vector Fields of Generalized Linear Models

Let $\mathcal{G} \subsetneq \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$ denote the class of functions of the form

$$f(x) = \sum_{i=1}^m \sigma(\langle x, z_i \rangle) \tag{4}$$

where m is a positive integer, $z_i \in \mathbb{R}^n$, and $\sigma \in \mathcal{C}^1(\mathbb{R})$. Such functions arise in statistics and optimization when considering generalized linear models. For example, when $\sigma(t) = \ln(1 + e^{-t})$, (4) is effectively the loss function used in logistic regression.

We further define $\mathcal{G}_\perp \subsetneq \mathcal{G}$ to be the set of functions of the form (4) where $\{z_i\}_{i=1}^m$ are mutually orthogonal. We then have the following result.

Theorem 1. *Let $f \in \mathcal{G}_\perp$ be as in (4). Let $\phi_i(t) = \|z_i\|^2\sigma'(t)$. For all $k \geq 2$,*

$$(\nabla f)^k(x) = \nabla \left(\sum_{i=1}^m \int_0^{\langle x, z_i \rangle} \sigma'(\phi_i^{k-1}(t))dt \right). \tag{5}$$

Thus, $\nabla \mathcal{G}_\perp$ is ∞-conservative and closed under self-composition.

In order to understand the dynamics of gradient descent on generalized linear models, we now extend Theorem 1 to the function class $I - \gamma \nabla \mathcal{G}_\perp$.

Theorem 2. *Let $f \in \mathcal{G}_\perp$ be as in (4). For fixed $\gamma \in \mathbb{R}$, let $\psi_i(t) = t - \gamma\|z_i\|^2\sigma'(t)$. For all $k \geq 2$,*

$$(I - \gamma \nabla f)^k(x) = x - \gamma \nabla \left(\sum_{i=1}^m \int_0^{\langle x, z_i \rangle} \sigma'(\psi_i^{k-1}(t))dt \right). \tag{6}$$

Thus, $I - \gamma \nabla \mathcal{G}_\perp$ is ∞-conservative and closed under self-composition.

On the other hand, $\nabla \mathcal{G}$ is not 2-conservative. Let $f_1(x, y) = e^x$, $f_2(x, y) = e^{x+y}$, $f_3 = f_1 + f_2$. Note that by Theorem 1, $\nabla f_1, \nabla f_2$ are both ∞-conservative. However, by direct computation

$$(\nabla f_3)^2(x, y) = \begin{pmatrix} \exp(e^x + e^{x+y}) + \exp(e^x + 2e^{x+y}) \\ \exp(e^x + 2e^{x+y}) \end{pmatrix} =: \begin{pmatrix} h_1(x, y) \\ h_2(x, y) \end{pmatrix}.$$

One can then verify that $\frac{\partial}{\partial y} h_1(x, y) \neq \frac{\partial}{\partial x} h_2(x, y)$, so by Clairaut's theorem, ∇f_3 is not 2-conservative. Notably, f_1, f_2 and f_3 are all convex functions, demonstrating that whether $\nabla \mathcal{F}$ is ∞-conservative is not determined by whether the class $\mathcal{F}$ is convex.

Finally, we note that it is not clear whether there are ∞-conservative vector fields in $\nabla \mathcal{G} \backslash \nabla \mathcal{G}_\perp$. Exactly characterizing the set of ∞-conservative vector fields in $\nabla \mathcal{G}$ remains open.

4 Smooth k-Conservative Vector Fields

We now explicitly construct the space of smooth, k-conservative vector fields. Given $V \in \mathcal{C}^\infty(\mathbb{R}^n)$, let $J(V) : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ denote its Jacobian, which we can view as an $n \times n$ matrix over $\mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$. If $V \in \mathcal{C}^\infty(\mathbb{R}^n)$, then by the Poincaré lemma (see [16, Section 4.18] for reference), V is k-conservative if and only if $J(V^k)$ is symmetric. For $k \geq 1$, we then define $D_k : \mathcal{C}^\infty(\mathbb{R}^n) \to \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R}^{n \times n})$ by

$$D_k(V) := J(V^k) - J(V^k)^\intercal. \tag{7}$$

Thus, $V \in \mathcal{C}^\infty(\mathbb{R}^n)$ is k-conservative if and only if $D_k(V) = 0$. We may now define the space of smooth, k-conservative vector fields by $\mathcal{W}^k(\mathbb{R}^n) := D_k^{-1}(\{0\})$ and $\mathcal{W}^\infty(\mathbb{R}^n) := \cap_{k=1}^\infty \mathcal{W}^k(\mathbb{R}^n)$. We note a few facts about $\mathcal{W}^\infty(\mathbb{R}^n)$:

1. $W^k(\mathbb{R}^n)$ and $\mathcal{W}^\infty(\mathbb{R}^n)$ are closed in $\mathcal{C}^\infty(\mathbb{R}^n)$ under several natural topologies, like that of uniform convergence of all derivatives on compact sets. To see this, note that D_k is a continuous function in this topology, so $D_k^{-1}(\{0\}) = W^k(\mathbb{R}^n)$ is closed. Thus, $\mathcal{W}^\infty(\mathbb{R}^n)$ is an intersection of closed sets, and is closed itself.
2. $\mathcal{W}^\infty(\mathbb{R}^n)$ is closed under scalar multiplication. While it contains linear subspaces (such as the space of symmetric linear vector fields, see Section 3), it is not closed under addition. For a simple counter-example, see the end of Section 3.1.
3. While $\mathcal{W}^\infty(\mathbb{R}^n)$ is closed under self-composition, it is not closed under arbitrary composition. See Appendix A for an explicit counter-example.

Some basic open questions on the structure of $\mathcal{W}^\infty(\mathbb{R}^n)$:

1. How does $\mathcal{W}^k(\mathbb{R}^n)$ relate to $\mathcal{W}^j(\mathbb{R}^n)$ for $k \neq j$? As we show in Appendix A, $\mathcal{W}^k(\mathbb{R}^n) \not\subseteq \mathcal{W}^j(\mathbb{R}^n)$ for $j < k$. More generally, are there smooth vector fields that are k-conservative but not j-conservative for $j \neq k$?
2. If we restrict to $\mathcal{P}_d(\mathbb{R}^n)$, the zero locus of D_k defines a projective variety over the coefficients of polynomials in $\mathcal{P}_d(\mathbb{R}^n)$. For example, applying Eq. (7) to $\mathcal{P}_d(\mathbb{R}^n)$, we find:
   - $\mathcal{W}^1(\mathbb{R}^n) \cap \mathcal{P}_1(\mathbb{R}^n)$ is a hyperplane.
   - $\mathcal{W}^2(\mathbb{R}^2) \cap \mathcal{P}_1(\mathbb{R}^2)$ is a union of two hyperplanes.
   - $\mathcal{W}^3(\mathbb{R}^2) \cap \mathcal{P}_1(\mathbb{R}^2)$ is a union of a hyperplane and a quadric surface.
   - $\mathcal{W}^1(\mathbb{R}^2) \cap \mathcal{W}^2(\mathbb{R}^2) \cap \mathcal{P}_2(\mathbb{R}^2)$ is a quadric surface.

   See Appendix A for the full details on these computations. Can we say anything more general? For example, what is the degree of $\mathcal{W}^k(\mathbb{R}^n) \cap \mathcal{P}_d(\mathbb{R}^n)$?

5 Implications for Optimization

In this section, we show that k-conservatism has important ramifications for optimization. We first show that if f is a smooth function and ∇f is ∞-conservative, then functions g_k with $\nabla g_k = (\nabla f)^k$ inherit many geometric properties of f. Due to their importance in optimization, we focus on notions of convexity. When f is smooth, we can reduce such properties to questions about eigenvalues of Jacobian matrices. We first bound the Jacobian eigenvalues of iterated vector fields.

Proposition 1. *Suppose $f \in \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$ and ∇f is j-conservative for $1 \leq j \leq k$, with $(\nabla f)^j = \nabla g_j$. Then for all such j, the function g_j is smooth and satisfies:*

1. *Suppose there are $\alpha, \beta \geq 0$ such that for all x, $\alpha I \preceq J(\nabla f)(x) \preceq \beta I$. Then for all x,*
$$\alpha^k I \preceq J(\nabla g_j)(x) \preceq \beta^k I.$$
2. *Suppose there is some $\lambda \geq 0$ such that for all x, $-\lambda I \preceq J(\nabla f)(x) \preceq \lambda I$. Then for all x,*
$$-\lambda^k I \preceq J(\nabla g_j)(x) \preceq \lambda^k I.$$

Items 1 and 2 also hold if we change $\preceq$ to $\prec$ throughout.

We will use Proposition 1 to show that iterating ∞-conservative vector fields preserves geometric properties, including Lipschitz continuity, as in the following definition.

Definition 3. *A vector field $V \in \mathcal{C}^1(\mathbb{R}^n)$ is β-Lipschitz continuous if for all $x \in \mathbb{R}^n$, $\|J(V)(x)\| \leq \beta$. V is Lipschitz continuous if there is some β for which V is β-Lipschitz continuous.*

In the definition above, $\|\cdot\|$ refers to the operator norm induced by the ℓ_2 norm on $\mathbb{R}^n$, viewing $J(V)(x)$ as an $n \times n$ matrix over $\mathbb{R}$. In the following, we let $\mathcal{L}(\mathbb{R}^n) \subsetneq \mathcal{V}(\mathbb{R}^n)$ denote the set of Lipschitz continuous vector fields. Proposition 1 directly implies the following result.

Corollary 1. *Let $\mathcal{F} \subsetneq \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$ be the set of (a) smooth, strongly convex functions, (b) smooth, strictly convex functions, or (c) smooth, convex functions. Then $\nabla\mathcal{F} \cap \mathcal{W}^\infty(\mathbb{R}^n)$ and $\nabla\mathcal{F} \cap \mathcal{W}^\infty(\mathbb{R}^n) \cap \mathcal{L}(\mathbb{R}^n)$ are closed under self-composition.*

Thus, we see that convexity "lifts" under self-composition of the associated gradient vector field: If f is smooth, convex, and ∇f is k-conservative, then $(\nabla f)^k = \nabla g$ for some smooth, convex function g.

Next, we consider vector fields $V = I - (I - \gamma\nabla f)^k$ where $\gamma > 0$ (induced by gradient descent). In the following lemma, we show that if V is ∞-conservative and $V^k = \nabla h_k$, then h_k inherits smoothness and critical points from f.

Lemma 1. *Let $f \in \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$ and $\gamma \in \mathbb{R}_{>0}$. Suppose that $(I - \gamma\nabla f)$ is j-conservative for $1 \leq j \leq k$. Then $V_k := I - (I - \gamma\nabla f)^k$ is conservative, and if $\nabla h_k = V_k$ then (1) h_k is smooth, and (2) if $\nabla f(y) = 0$, then $\nabla h_k(y) = 0$.*

In fact, many geometric properties important to optimization (such as convexity) are also inherited by h_k, provided that γ is not too large, as in the following.

Lemma 2. *Suppose $f \in \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$ and ∇f is β-Lipschitz continuous. Suppose that for some $\gamma \in \mathbb{R}_{>0}$, $(I - \gamma\nabla f)$ is j-conservative for $1 \leq j \leq k$, with $\nabla h_k = I - (I - \gamma\nabla f)^k$. Then:*

1. *If f is α-strongly convex and $\gamma \leq 2(\alpha + \beta)^{-1}$ then h_k is $(1 - \lambda^k)$-strongly convex and ∇h_k is $(1 + \lambda^k)$-Lipschitz continuous for $\lambda = 1 - \gamma\alpha$.*
2. *If f is convex and $\gamma \leq 2\beta^{-1}$ then h_k is convex and ∇h_k is 2-Lipschitz continuous. If $\gamma \leq \beta^{-1}$, then ∇h_k is 1-Lipschitz continuous.*
3. *If f is strictly convex and $\gamma < 2\beta^{-1}$ then h_k is strictly convex.*
4. *If f is δ-weakly convex for $\delta \leq \beta$ and $\gamma \leq 2\beta^{-1}$, then h_k is $(\lambda^k - 1)$-weakly convex and ∇h_k is $(1 + \lambda^k)$-Lipschitz continuous for $\lambda = 1 + \gamma\delta$.*

5.1 Convergence Rates of FEDAVG

We now use our machinery above to understand the convergence of FEDAVG in various settings. Recall that the server update at each round is given by $x_{t+1} = x_t - \eta V_s(x_t)$, where the "server vector field" V_s is given by (2). Throughout, we assume that each client c performs k steps of gradient descent with learning rate $\gamma > 0$ on their loss function f_c. We make the following assumption.

Assumption 1. *For all c, $f_c \in \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$ and $I - \gamma\nabla f_c$ is j-conservative for $1 \leq j \leq k$.*

This leads to the following result on sufficient conditions for V_s to be conservative.

Theorem 3. *Under Assumption 1, V_s is a conservative vector field. Moreover, if $V_s = \nabla f_s$, then f_s is smooth and the FEDAVG server update in (1) is equivalent to the following gradient descent step:*

$$x_{t+1} = x_t - \eta\nabla f_s(x_t). \tag{8}$$

In this setting, if we have some understanding of f_s (for example, whether f_s is convex), we can immediately apply centralized optimization results to derive convergence results for FEDAVG. To better understand the structure of f_s, we will use Lemma 2. Since this requires Lipschitz continuity, we make the following assumption.

Assumption 2. *For all c, ∇f_c is β-Lipschitz continuous.*

Under Assumptions 1 and 2, Lemma 2 lifts geometric properties of the f_c to f_s. Combining this with Theorem 3, we can translate convergence rates for gradient descent to convergence rates for FEDAVG in strongly convex and convex settings. We make no direct assumptions on client heterogeneity. Throughout, we let f_s be a function such that $V_s = \nabla f_s$, as guaranteed by Theorem 3.

Theorem 4. *Suppose Assumptions 1 and 2 hold, and that for all c, f_c is α-strongly convex. Then f_s has a unique minimizer x_s^*, and if $\gamma = 2(\alpha + \beta)^{-1}$, $\eta = 1$, the iterates $\{x_t\}_{t=0}^\infty$ of FEDAVG satisfy*

$$\|x_t - x_s^*\| \leq \left(\frac{\beta - \alpha}{\beta + \alpha}\right)^{kt} \|x_0 - x_s^*\|. \tag{9}$$

Proof. This follows directly from combining Theorem 3 and Lemma 2 with well-known convergence rates for smooth, strongly convex functions (for example, see [2, Theorem 3.10]). □

The convergence rate in (9) was shown first in [8, Theorem 2.11], which extended to non-conservative gradient vector fields. The salient difference is that under under our assumptions, the limit point x_s^* is actually the global minimizer of some strongly convex function. As we discuss below, this allows us to immediately derive analogous results for variants of FEDAVG that apply other server optimizers.

When $k = 1$, this recovers the convergence of gradient descent on $f_{avg} = C^{-1}\sum_{c=1}^{C} f_c$. Hence, FEDAVG with $k > 1$ yields an exponential improvement in convergence (with respect to k), but may not converge to the minimizer x^* of f_{avg}. To understand this discrepancy, one could analyze $\|x_s^* - x^*\|$. A tight upper bound was given for strongly convex quadratic functions in [3, Lemma 5]. A bound in the general strongly convex setting (not assuming k-conservatism) was given in [8, Theorem 2.14], though whether this bound can be improved under Assumption 1 is an open question.

We now give a convergence rate for FEDAVG in the convex setting.

Theorem 5. *Suppose Assumptions 1 and 2 hold, and that for all c, f_c is convex with finite minimizer. Then f_s has a finite minimizer x_s^*, and if $\gamma = \beta^{-1}$, $\eta = 1$, the iterates $\{x_t\}_{t=0}^{\infty}$ of* FEDAVG *satisfy*

$$f_s(x_t) - f_s(x_s^*) \leq \frac{1}{2t}\|x_0 - x_s^*\|^2. \tag{10}$$

Proof. This follows directly from combining Theorem 3 and Lemma 2 with well-known convergence rates for smooth, convex functions (for example, see [2, Theorem 3.3]). □

To the best of our knowledge, Theorem 5 is the first result showing that FEDAVG exhibits convergent behavior on a class of functions, even with fixed learning rates and $k > 1$. Unlike Theorem 4, it is not clear that the convergence in (10) is "faster" (in some sense) than the convergence of gradient descent on f_{avg}. Such analysis is an open and important problem.

These techniques allow us to transfer convergence rates for many optimization algorithms to federated learning under the same assumptions as Theorems 4 and 5; We can directly analyze any federated learning algorithm where the server gradient descent step in (1) is replaced with another optimizer (as proposed in [13]). For example, using SGD with momentum [5] or adaptive methods such as Adagrad [4, 10] can lead to improved empirical convergence [13]. Notably, our framework can be used directly to show that in the strongly convex setting, methods such as Nesterov momentum accelerate convergence to x_s^*.

6 Open Problems

As we have shown, FEDAVG is well-behaved when certain vector fields are k-conservative (Theorems 4 and 5) and can exhibit non-convergent, circular behavior when they are not (Section 2.1). Better characterizations of when FEDAVG exhibits convergent behavior (or fails to do so) is an important open problem. Similarly, we have only scratched the surface on how the dynamics of the client loss functions lift to the server dynamics. While many convexity-adjacent properties lift (Lemma 2), one can show that many natural properties (including being bounded below) do not lift. What about properties such as the Polyak-Łojasiewicz condition [6]? More generally, which properties lift, and can we understand the behavior of FEDAVG on some class of non-convex functions?

Another important area is understanding the empirical effectiveness of methods such as FEDAVG. As discussed in [15], theoretical convergence rates of federated learning methods often do not improve upon centralized rates for algorithms such as gradient descent. While Theorem 4 shows that FEDAVG accelerates convergence to a non-optimal point, it is unclear whether Theorem 5 implies a similar acceleration. More generally, is there some sense in which the limit point x_s^* is a useful point of convergence, either for learning a global model, or as a starting point for personalization?

Finally, the non-conservative dynamics presented in Section 2.1 point to a fundamental failure of methods such as FEDAVG. This mirrors non-conservative dynamics arising from many GAN training methods [11]. Can we use insights from training multi-agent systems (such as GANs) to create better federated learning methods, or even to simply design better client loss functions?

References

[1] K. A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé M Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In *SysML 2019*, 2019.

[2] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[3] Zachary Charles and Jakub Konečný. Convergence and accuracy trade-offs in federated learning and meta-learning. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

[4] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[5] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[6] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 795–811, Cham, 2016. Springer International Publishing.

[7] Saber Malekmohammadi, Kiarash Shaloudegi, Zeou Hu, and Yaoliang Yu. An operator splitting view of federated learning. *arXiv preprint arXiv:2108.05974*, 2021.

[8] Grigory Malinovskiy, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtarik. From local SGD to local fixed-point methods for federated learning. In *International Conference on Machine Learning*, pages 6692–6701. PMLR, 2020.

[9] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[10] H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

[11] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[12] Reese Pathak and Martin J Wainwright. Fedsplit: an algorithmic framework for fast federated optimization. *Advances in Neural Information Processing Systems*, 33:7057–7066, 2020.

[13] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

[14] Michael Spivak. *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*. CRC press, 2018.

[15] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[16] Frank W. (Frank Wilson) Warner. Foundations of differentiable manifolds and Lie groups, 1983.