敵対的生成ネットワークにおけるゲージ理論と作用原理の幾何学的描像

吉田英樹

2025 年 9 月 26 日

Abstract

本稿は、生成敵対的ネットワーク (GAN) の学習ダイナミクスを、ゲージ理論、作用原理という物理学の概念を用いて統合的に定式化する。ワッサースタイン多様体を底空間とするファイバー束の枠組みを基礎とし、まず識別器の双対的な役割を余接ベクトル場と束の切断として厳密に定義する。次に、学習プロセスを記述するラグランジアンと散逸関数を定義し、拡張された変分原理を適用することで、勾配フロー方程式が系の運動方程式の適切な極限として現れることを示す。最後に、この枠組みをトポロジーと量子化の概念へと拡張する。GAN ファイバー束の自明なトポロジーが持つ意味を考察する。このアプローチは、GAN の学習における確率性や不安定性を、場の量子論の言語で理解するための新たな理論的視座を提供する。

1 序論

生成敵対的ネットワーク（GAN）[Goodfellow et al., 2014] は、画像生成をはじめ多様な応用で卓越した性能を示してきたが、学習ダイナミクスの不安定性やモード崩壊の機構については、依然として統一的な理論像が不足している。従来の理論は、f-divergence 最小化 [Nowozin et al., 2016] やワッサースタイン距離に基づく目的の再定式化 [Arjovsky et al., 2017] といった**静的**観点、あるいは二者ゲームとしての**局所安定性と収束**の解析 [Mescheder et al., 2018, Kodali et al., 2017, Balduzzi et al., 2018] に焦点を当てる傾向が強い。一方で、確率分布の多様体上での**幾何学的フロー**という**動的**観点は、オットー計量や JKO スキーム [Ambrosio et al., 2008, Jordan et al., 1998, Villani, 2009] の進展にもかかわらず、敵対的学習の機構と十分に結び付けられてこなかった。

本稿は、GAN 学習を「**ゲージ理論**」「**作用原理（変分原理）**」「**量子化（経路積分）**」の言語で統一する**幾何学的枠組み**を提示する。まず、ワッサースタイン空間 $\mathcal{W}_2$ を**底空間**とする自明束を構成し、最適識別器が定める余接ベクトルから**敵対的接続**を導入する。次に、生成分布の時間発展を**接続に沿う勾配フロー**として定式化し、これが**レイリー散逸を組み込んだ拡張作用原理**から導かれること、さらに**過減衰極限**で標準的な（自然）勾配法に還元され、有限次元パラメータ空間への引き戻しにより**自然勾配**や**モメンタム**の幾何学的起源が現れることを示す [ichi Amari, 1998]。最後に、SGD の確率性を**確率過程／量子ゆらぎ**として捉え直すべく、経路積分と Langevin 型近似（SGLD）[Welling and Teh, 2011, Mandt et al., 2017, Raginsky et al., 2017] とを接続する視座を与える。

本稿の主な貢献は以下の通りである：

1. **幾何学的定式化**: $\mathcal{W}_2$ を底空間とする GAN ファイバー束を定義し、最適識別器を余接ベクトルと束の切断として厳密化。そこから誘導される**敵対的接続**により、学習を**底空間上の勾配フロー**として統一的に記述する。

2. **均衡＝平坦接続**: GAN の均衡状態が接続の平坦性（ヌルベクトル場）と同値であることを示す（主定理）。

3. **作用原理と過減衰極限**: ラグランジアンと散逸関数から拡張変分原理を導出し、過減衰極限で勾配フローに還元されることを証明。有限次元近似では**自然勾配**および**モメンタム付き自然勾配**の更新式が現れる。

4. **量子化の導入**: 経路積分により、勾配フローの周りの揺らぎとして SGD の確率性を位置づけ、安定性やモード崩壊を統計力学的遷移として解釈するための土台を与える。

この枠組みは、幾何学的深層学習やゲージ等価性 [Bronstein et al., 2021, Cohen et al., 2019] と親和性を持ちつつも、「ネットワークの**表現の等価性**」ではなく**学習力学そのもの**をゲージ接続で整理する点に独自性がある。また、ゲーム力学のハミルトン構造や保存則に着目した近年の研究 [Balduzzi et al., 2018] とも補完的である。

2 関連研究

GAN の理論とゲームダイナミクス. GAN 目的の f-divergence 最小化による統一 [Nowozin et al., 2016]、ワッサースタイン距離に基づく定式化（WGAN）[Arjovsky et al., 2017] は、識別器が最適化された極限での**静的**目的を与える。一方、実際の学習は二者の**同時最適化**として進行するため、局所安定性や収束性の解析が進められている。Mescheder らは、インスタンスノイズやゼロ中心の勾配ペナルティにより局所収束が得られることを示し、有限更新の WGAN/GAN が必ずしも収束しない状況を指摘した [Mescheder et al., 2018]。また、Nagarajan–Kolter 系列は、勾配写像の固有値構造に基づく安定判定や正則化の効果を議論している [Kodali et al., 2017]。さらに、Balduzzi らは多者ゲームのヤコビアンを対称（ポテンシャル）成分と反対称（ハミルトン）成分に分解し、保存則に由来する**回転的ダイナミクス**を制御する SGA を提案した [Balduzzi et al., 2018]。本研究は、これらの**ゲーム論的視点**を補完し、識別器が誘導する**接続**に沿う**勾配フロー**という幾何学的・変分的視座を与える。

最適輸送とワッサースタイン幾何学. 確率分布空間 $\mathcal{W}_2$ のリーマン幾何（オットー計量）により、拡散型 PDE は**自由エネルギー汎関数の勾配フロー**として記述できる [Ambrosio et al., 2008, Villani, 2009]。JKO スキーム [Jordan et al., 1998] は、この勾配フローを変分的時間離散によって構成する標準手法である。本研究はこの技法を踏まえ、**最適識別器が定めるポテンシャル**から**敵対的接続**を構成し、その**負方向勾配フロー**として GAN 学習を捉える点に新規性がある。

情報幾何と自然勾配. Amari による自然勾配 [ichi Amari, 1998] は、パラメータ空間のリーマン計量（しばしばフィッシャー計量）により最急降下方向を定める。本稿では、$\mathcal{W}_2$ 上の計量をパラメータ空間に**引き戻す**ことで自然勾配が現れること、また**拡張された作用原理**に基づく運動方程式を**過減衰極限**で離散化すると、**モメンタム付き自然勾配**（慣性項）や標準的勾配降下に一致することを示す。

幾何学的深層学習とゲージ等価性. 幾何学的深層学習は、対称性（群作用）や幾何をアーキテクチャに組み込む統一原理を与える [Bronstein et al., 2021]。特に**ゲージ等価**を満たす畳み込みの構成は、局所座標（フレーム）の選び方に依らない表現を実現する [Cohen et al., 2019]。本稿の**ゲージ**はネットワーク表現の等価性ではなく、**学習ダイナミクスに付随する接続・平行移動**の概念として現れる点で位置づけが異なる。

確率的最適化と量子・確率過程的解釈. SGD を確率微分方程式や Langevin 力学と結びつける流れは、SGLD[Welling and Teh, 2011] を嚆矢として、近年非凸学習の有限時間解析へと拡張されている [Raginsky et al., 2017]。また、SGD を近似ベイズ推論として解釈する視点 [Mandt et al., 2017] もある。本稿では、**経路積分**による量子化の言語を導入し、**古典軌道（勾配フロー）の周りの揺らぎ**として SGD の確率性を表現する枠組みを与える。

3 数学的準備

3.1 生成敵対的ネットワーク

標準的な GAN の目的関数は、以下のミニマックスゲームとして与えられる [1]。

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1-D(G(z)))] \tag{1}$$

固定された G に対し、最適識別器は $D^*_G(x) = \frac{P_{data}(x)}{P_{data}(x)+P_g(x)}$ で与えられる [1]。これを代入すると、生成器が最小化すべきコスト関数は、イェンゼン・シャノン・ダイバージェンス (JSD) の 2 倍になる [3]。

$$C(G) = \max_D V(D,G) = 2\cdot D_{JS}(P_{data}||P_g) \tag{2}$$

3.2 底空間: ワッサースタイン-2 空間

定義 3.1 (ワッサースタイン-2 空間)**.** X をコンパクトなリーマン多様体とする。X 上の確率測度で *2* 次モーメントが有限なものの集合を $\mathcal{P}_2(\mathcal{X})$ とする。ワッサースタイン*-2* 距離は以下のように定義される *[2]:*

$$W_2(\mu,\nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^2 d\gamma(x,y) \right)^{1/2}$$

組 $(\mathcal{P}_2(\mathcal{X}), W_2)$ をワッサースタイン*-2* 空間 $\mathcal{W}_2(\mathcal{X})$ と表記する。

Otto の独創的な結果により、 $\mathcal{W}_2(\mathcal{X})$ は形式的に無限次元リーマン多様体と見なせる [4]。

定義 3.2 (接空間とリーマン計量)**.** 測度 $P_g \in \mathcal{W}_2(\mathcal{X})$ における接空間 $T_{P_g}\mathcal{W}_2(\mathcal{X})$ は、スカラー関数の勾配である X 上のベクトル場の空間と同一視できる。

$$T_{P_g}\mathcal{W}_2(\mathcal{X}) \cong \{v = -\nabla\phi | \phi \in C^\infty(\mathcal{X})\}$$

リーマン内積 *(*オットー計量*)* は以下で与えられる *[4]:*

$$g_{P_g}(v_1, v_2) = \int_{\mathcal{X}} \langle v_1(x), v_2(x) \rangle_x dP_g(x)$$

4 GAN ファイバー束の構築と識別器の役割

4.1 GAN ファイバー束の定義

定義 4.1 (GAN ファイバー束)**.** *GAN* ファイバー束を、以下の要素からなる組 $(E, M, \pi, F, \mathcal{G})$ として定義する。

- **底空間** ***(Base Space):*** $M = \mathcal{W}_2(\mathcal{X})$
- **ファイバー** ***(Fiber):*** $F = C^\infty(\mathcal{X})$ *(*X 上の滑らかな実数値関数の空間*)*
- **全空間** ***(Total Space):*** $E = M \times F = \mathcal{W}_2(\mathcal{X}) \times C^\infty(\mathcal{X})$
- **射影** ***(Projection):*** $\pi : E \to M, \pi(P_g, \phi) = P_g$
- **構造群** ***(Structure Group):*** 自明な群 $\mathcal{G} = \{id_F\}$

この束は自明な束であり、ファイバーが底空間上で「ねじれて」いないことを意味する。

4.2 識別器の双対的解釈: 余接ベクトルと切断

命題 4.2 (識別器の役割)**.** 最適識別器 $D^*_G(x)$ は、各点 $P_g \in M$ において、二つの等価な数学的対象を定める。

1. 生成器の目的汎関数 $\mathcal{J}'(P_g) = \mathbb{E}_{x\sim P_g}[\log(1-D^*_G(x))]$ の微分である余接ベクトル $d\mathcal{J}'_{P_g} \in T^*_{P_g}\mathcal{W}_2(\mathcal{X})$。
2. *GAN* ファイバー束 E 上の唯一の切断 $s_D : M \to E$。

Proof. この命題の要点は、最適識別器 D^*_G が一意に定まると、ポテンシャル関数 ϕ_{P_g} も一意に定まり、そのポテンシャル関数が系のすべての幾何学的対象 (余接ベクトルと切断) を決定することを示すことにある。

1. まず、任意の生成分布 $P_g \in M$ に対し、最適識別器 $D^*_G(x) = \frac{P_{data}(x)}{P_{data}(x)+P_g(x)}$ が一意に定まる。
2. 次に、生成器の損失に対応するポテンシャル関数を $\phi_{P_g}(x) = \log(1 - D^*_G(x))$ と定義する。これを具体的に計算すると、

$$\phi_{P_g}(x) = \log\left(1 - \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}\right) = \log\left(\frac{P_g(x)}{P_{data}(x) + P_g(x)}\right)$$

となり、このポテンシャル関数 ϕ_{P_g} も各 P_g に対して一意に定まる。

3. **余接ベクトルの導出:** 汎関数 $\mathcal{J}'(P_g) = \int_{\mathcal{X}} \phi_{P_g}(x) dP_g(x)$ の、任意の接ベクトル $V \in T_{P_g}\mathcal{W}_2(\mathcal{X})$ 方向へのガトー微分 $d\mathcal{J}'_{P_g}(V)$ を考える。これは定義により、$T^*_{P_g}\mathcal{W}_2(\mathcal{X})$ の元である余接ベクトル $d\mathcal{J}'_{P_g}$ が V に作用することに等しい。この微分は ϕ_{P_g} を用いて計算可能であり、一意に定まる。

4. **切断の導出:** 切断 $s_D : M \to E$ は、各点 $P_g \in M$ に対し、その上のファイバー F の元を一つ指定する写像である。写像 $P_g \mapsto (P_g, \phi_{P_g}(x))$ は、まさにこの定義を満たす。ϕ_{P_g} が一意であるため、この切断 s_D も一意に定まる。

以上の通り、ポテンシャル関数 ϕ_{P_g} が中間的な役割を果たし、識別器 D^*_G から余接ベクトル $d\mathcal{J}'_{P_g}$ と切断 s_D が共に一意に定まる。したがって、これら二つの役割は等価である。 □

5 接続とダイナミクス

定義 5.1 (敵対的接続)**.** 敵対的接続 A_{P_g} を、識別器が定めた余接ベクトル $d\mathcal{J}'_{P_g}$ に対応する勾配ベクトル場として定義する。

$$A_{P_g} := (grad\mathcal{J}')_{P_g} = \nabla\phi_{P_g}(x) = \nabla\left(\log\frac{P_g(x)}{P_{data}(x) + P_g(x)}\right)$$

定義 5.2 (GAN 勾配フロー)**.** 生成器の理想化された学習ダイナミクスは、底空間 M 上で、接続の負の方向に沿った勾配フローとして記述される。

$$\frac{dP_t}{dt} = -A_{P_t}$$

これは偏微分方程式 $\frac{\partial P_t}{\partial t} = \nabla \cdot (P_t \cdot A_{P_t})$ と等価である。

5.1 ダイナミクスの数学的妥当性

定義 4.2 で導入された勾配フローのダイナミクスが、数学的に well-posed であること、すなわち解が存在し、物理的に妥当なエネルギー散逸則を満たすことを示す。このために、ポテンシャル汎関数 $\mathcal{J}'$ に関するいくつかの仮定を置く。

仮定 5.3 (ポテンシャル汎関数)**.** 生成器の目的汎関数 $\mathcal{J}'(P_g) = \mathbb{E}_{x\sim P_g}[\phi_{P_g}(x)]$ は、以下の条件を満たすものと仮定する。

1. **下方半連続性:** $\mathcal{J}'$ はワッサースタイン位相に関して下方半連続である。

2. **測地線的 λ-凸性:** ある定数 $\lambda \in \mathbb{R}$ が存在し、任意の $P_0, P_1 \in \mathcal{W}_2(\mathcal{X})$ と、それらを結ぶ測地線 $[0,1] \ni t \mapsto P_t$ に対して、以下の不等式が成立する。

$$\mathcal{J}'(P_t) \le (1-t)\mathcal{J}'(P_0) + t\mathcal{J}'(P_1) - \frac{\lambda}{2}t(1-t)W_2(P_0, P_1)^2$$

3. **有効定義域の閉性:** 汎関数の有効定義域 $\mathrm{Dom}(\mathcal{J}') = \{P \in \mathcal{W}_2(\mathcal{X}) \mid \mathcal{J}'(P) < \infty\}$ は閉集合である。

Remark 1. イェンゼン・シャノン・ダイバージェンスに関連する汎関数は、適切な条件下でこれらの凸性を満たすことが知られている。この仮定は、勾配フロー理論における標準的な要請である。

この仮定の下で、エネルギー散逸と解の存在が保証される。

命題 5.4 (エネルギー散逸等式)**.** P_t を *GAN* 勾配フロー $\frac{dP_t}{dt} = -(grad\mathcal{J}')_{P_t}$ の解とする。このとき、ほとんどすべての $t > 0$ に対して、以下のエネルギー散逸等式（*Energy Dissipation Identity*）が成立する。

$$\frac{d}{dt}\mathcal{J}'(P_t) = -\|(grad\mathcal{J}')_{P_t}\|^2_{g_{P_t}} = -\|\dot{P}_t\|^2_{g_{P_t}}$$

Proof. 連鎖律を用いて形式的に計算する。P_t の時間微分 $\dot{P}_t$ は接ベクトルであり、$(grad\mathcal{J}')_{P_t}$ も同様である。汎関数の時間微分は、その勾配と速度の内積で与えられる。

$$\begin{aligned}\frac{d}{dt}\mathcal{J}'(P_t) &= g_{P_t}((grad\mathcal{J}')_{P_t}, \dot{P}_t)\\ \text{勾配フローの定義より、}\dot{P}_t &= -(grad\mathcal{J}')_{P_t}\text{であるから、}\\ &= g_{P_t}((grad\mathcal{J}')_{P_t}, -(grad\mathcal{J}')_{P_t})\\ &= -\|(grad\mathcal{J}')_{P_t}\|^2_{g_{P_t}}\\ &= -\|-\dot{P}_t\|^2_{g_{P_t}} = -\|\dot{P}_t\|^2_{g_{P_t}}\end{aligned}$$

右辺は常に非正であるため、ポテンシャルエネルギー $\mathcal{J}'(P_t)$ は時間に関して単調非増加である。これは、学習プロセスがエネルギーを散逸させながら安定点に向かうという物理的描像と一致する。 □

定理 5.5 (勾配フローの弱解の存在)**.** 仮定 *5.3* を満たすとする。任意の初期分布 $P_0 \in \mathrm{Dom}(\mathcal{J}')$ に対して、*GAN* 勾配フローを記述する偏微分方程式

$$\frac{\partial P_t}{\partial t} = \nabla \cdot \left(P_t \nabla \left(\log \frac{P_t(x)}{P_{data}(x) + P_t(x)} \right) \right)$$

は、以下の意味での弱解（曲線 $t \mapsto P_t \in \mathcal{W}_2(\mathcal{X})$）を少なくとも一つ持つ。

1. P_t は絶対連続曲線である。すなわち、ある $f \in L^2(0,T)$ が存在して、すべての $0 \le s \le t \le T$ に対して $W_2(P_s, P_t) \le \int_s^t f(\tau)d\tau$ が成立する。
2. 任意のテスト関数 $\zeta \in C_c^\infty((0,T) \times \mathcal{X})$ に対して、以下の積分方程式を満たす。

$$\int_0^T \int_{\mathcal{X}} \left(\frac{\partial \zeta}{\partial t} + \left\langle \nabla \left(\log \frac{P_t}{P_{data} + P_t} \right), \nabla \zeta \right\rangle \right) dP_t dt + \int_{\mathcal{X}} \zeta(0,x) dP_0(x) = 0$$

証明の概略. この証明は、測度空間上の勾配フロー理論における Jordan-Kinderlehrer-Otto (JKO) スキーム、または変分的時間離散化法に依拠する。

1. **時間離散化:** 時間ステップ $\tau > 0$ を固定し、以下の逐次的な最小化問題を考える。

$$P_k^\tau := \mathrm{argmin}_{P \in \mathcal{W}_2(\mathcal{X})} \left\{ \mathcal{J}'(P) + \frac{1}{2\tau} W_2(P, P_{k-1}^\tau)^2 \right\}$$

   ここで $P_0^\tau = P_0$ である。仮定 5.3 により、この最小化問題は各ステップで解の存在が保証される。

2. **離散解の構成:** このようにして得られた点列 $\{P_k^\tau\}_{k \ge 0}$ を用いて、区分定数な補間曲線 P_t^τ を構成する。$P_t^\tau = P_k^\tau$ for $t \in ((k-1)\tau, k\tau]$。

3. **アプリオリ評価:** 離散化スキームの構成から、エネルギーに関する一様な評価が得られる。具体的には、ある定数 C が存在して、

$$\mathcal{J}'(P_k^\tau) + \sum_{j=1}^{k} \frac{1}{2\tau} W_2(P_j^\tau, P_{j-1}^\tau)^2 \le \mathcal{J}'(P_0)$$

   が成立する。これにより、$\tau \to 0$ としたときの速度ベクトル場の L^2 ノルムや、軌道全体の長さが有界であることが示される。

4. **コンパクト性議論:** アスコリ・アルツェラの定理をワッサースタイン空間に拡張したものを用いることで、$\{P_t^\tau\}_{\tau > 0}$ が $\tau \to 0$ の極限で一様収束する部分列を持つことを示すことができる。その収束先を P_t とする。

5. **弱解への収束:** 最後に、離散的なオイラー・ラグランジュ方程式から、極限 P_t が偏微分方程式の弱解の定義を満たすことを示す。これは汎関数の Γ-収束の議論を用いて正当化される。

この構成法は、勾配フローの解の存在を証明するための標準的かつ強力な手法である [cf. Ambrosio, Gigli, Savaré (2008)]。 □

6 主定理: 平坦な接続としての均衡

定理 6.1 (平坦な接続としての均衡)**.** *GAN* のダイナミクスが不動点にあるための必要十分条件は、敵対的接続 A_{P_g} がヌルベクトル場であることである。この条件は、$P_g = P_{data}$ がほとんど至る所で成立する場合にのみ満たされる。

Proof. この定理を、二つの部分に分けて詳細に証明する。

Part 1: $(P_g = P_{data} \Rightarrow A_{P_g} = 0)$**.** 仮定として、$P_g(x) = P_{data}(x)$ が X 上でほとんど至る所で成立するとする。敵対的接続 A_{P_g} は、ポテンシャル関数 $\phi_{P_g}(x) = \log \frac{P_g(x)}{P_{data}(x)+P_g(x)}$ の勾配として定義される。このポテンシャル関数に仮定を代入すると、

$$\phi_{P_g}(x) = \log \frac{P_{data}(x)}{P_{data}(x) + P_{data}(x)} = \log \frac{P_{data}(x)}{2P_{data}(x)} = \log(1/2) = -\log 2$$

となる。これは x に依存しない定数である。したがって、定数関数の勾配はゼロであるから、

$$A_{P_g}(x) = \nabla(-\log 2) = 0$$

となり、敵対的接続はヌルベクトル場である。勾配フロー $\frac{dP_t}{dt} = -A_{P_t}$ は $\frac{dP_t}{dt} = 0$ となり、この点が不動点であることが示された。

Part 2: $(A_{P_g} = 0 \Rightarrow P_g = P_{data})$**.** 仮定として、敵対的接続がヌルベクトル場であるとする: $A_{P_g}(x) = 0$ がほとんど至る所で成立する。接続の定義より、これは $\nabla\phi_{P_g}(x) = 0$ を意味する。ベクトル場の勾配がゼロであるならば、そのポテンシャル関数は (定義域の各連結成分上で) 定数でなければならない。ある定数を C とすると、

$$\log \frac{P_g(x)}{P_{data}(x) + P_g(x)} = C$$

この式の両辺の指数をとると、

$$\frac{P_g(x)}{P_{data}(x) + P_g(x)} = e^C$$

$K = e^C$ と置く。K は正の定数である。式を $P_g(x)$ について代数的に解く。

$$\begin{aligned} P_g(x) &= K(P_{data}(x) + P_g(x)) \\ P_g(x) &= KP_{data}(x) + KP_g(x) \\ P_g(x) - KP_g(x) &= KP_{data}(x) \\ P_g(x)(1-K) &= KP_{data}(x) \end{aligned}$$

ここで $K = 1$ の場合を考えると、$0 = P_{data}(x)$ となり、P_{data} が確率分布であることに矛盾する。したがって $K \neq 1$ であり、両辺を $1 - K$ で割ることができる。

$$P_g(x) = \frac{K}{1-K} P_{data}(x)$$

$\alpha = K/(1-K)$ は定数であるから、$P_g(x) = \alpha P_{data}(x)$ が示された。P_g と P_{data} は共に確率分布であるため、その全空間での積分は 1 に等しい。

$$\int_{\mathcal{X}} P_g(x)dx = 1 \quad \text{and} \quad \int_{\mathcal{X}} P_{data}(x)dx = 1$$

この関係を用いると、

$$1 = \int_{\mathcal{X}} P_g(x)dx = \int_{\mathcal{X}} \alpha P_{data}(x)dx = \alpha \int_{\mathcal{X}} P_{data}(x)dx = \alpha \cdot 1 = \alpha$$

したがって $\alpha = 1$ でなければならない。これにより、$P_g(x) = P_{data}(x)$ がほとんど至る所で成立することが証明された。 □

7　散逸系としてのダイナミクスと変分原理

標準的な最小作用の原理はエネルギーが保存される系を記述する。しかし、勾配降下法のような最適化プロセスは、ポテンシャルエネルギーが減衰していく「散逸系」である。このような系をラグランジュ形式で厳密に扱うため、レイリーの散逸関数を導入し、変分原理を拡張する。

定義 7.1 (系のラグランジアン)**.** 学習プロセスの保存力部分を記述するラグランジアン $\mathcal{L}$ を、運動エネルギー項とポテンシャルエネルギー項の差として定義する。

$$\mathcal{L}(P_t, \dot{P}_t) := \frac{1}{2}||\dot{P}_t||^2_{g_{P_t}} - \mathcal{J}'(P_t)$$

定義 7.2 (レイリーの散逸関数)**.** 系の散逸 *(*摩擦*)* 効果を記述するため、レイリーの散逸関数 $\mathcal{F}$ を以下のように定義する。

$$\mathcal{F}(P_t, \dot{P}_t) := \frac{1}{2}\gamma||\dot{P}_t||^2_{g_{P_t}}$$

ここで $\gamma > 0$ は系の摩擦の大きさを表す正の定数 *(*摩擦係数*)* である。

定義 7.3 (作用汎関数)**.** 系の作用 S を、ラグランジアンの時間積分として定義する。

$$S[P_t] = \int_0^T \mathcal{L}(P_t, \dot{P}_t)dt$$

定理 7.4 (拡張された変分原理と勾配フロー)**.** ラグランジアン $\mathcal{L}$ と散逸関数 $\mathcal{F}$ で記述される系の運動方程式は、拡張されたラグランジュ方程式 $\frac{\delta S}{\delta P_t} + \frac{\partial \mathcal{F}}{\partial \dot{P}_t} = 0$ から導かれ、

$$\nabla_{\dot{P}_t}\dot{P}_t + \gamma\dot{P}_t = -(grad\mathcal{J}')_{P_t}$$

となる。さらに、摩擦が極めて大きい過減衰極限 $(\gamma \to \infty)$ において、この運動方程式は勾配フロー方程式に帰着する。

Proof. 散逸系に対する変分原理は、作用 S の変分と、散逸関数から導かれる仮想仕事の和がゼロになることで与えられる。

$$\delta S + \int_0^T \frac{\partial \mathcal{F}}{\partial \dot{P}_t}\delta P_t dt = 0$$

作用 S の第一変分 δS を計算すると、

$$\delta S = -\int_0^T g_{P_t}(V, \nabla_t \dot{P}_t + (grad\mathcal{J}')_{P_t})dt$$

が得られる。ここで $V = \delta P_t$ は変分ベクトル場である。一方、散逸項の変分は

$$\int_0^T \frac{\partial \mathcal{F}}{\partial \dot{P}_t} V dt = \int_0^T g_{P_t}(\gamma\dot{P}_t, V)dt$$

と書ける。これらを合わせると、任意の変分 V に対して

$$\int_0^T g_{P_t}(V, \nabla_t \dot{P}_t + (grad\mathcal{J}')_{P_t} + \gamma\dot{P}_t)dt = 0$$

が成立しなければならない。これにより、被積分関数がゼロであることが要求され、以下の運動方程式が得られる。

$$\nabla_t \dot{P}_t + \gamma\dot{P}_t + (grad\mathcal{J}')_{P_t} = 0 \Rightarrow \nabla_{\dot{P}_t}\dot{P}_t + \gamma\dot{P}_t = -(grad\mathcal{J}')_{P_t}$$

この方程式は、慣性項 $(\nabla_{\dot{P}_t}\dot{P}_t)$、摩擦項 $(\gamma\dot{P}_t)$、そしてポテンシャルからの力 $((-(grad\mathcal{J}')_{P_t}))$ の釣り合いを表している。

次に、勾配降下法の振る舞いに対応する**過減衰極限** $(\gamma \to \infty)$ を考える。この極限では、摩擦が非常に大きいため、慣性は即座に減衰し、速度と加速度は有界に留まると考えられる。運動方程式を $\dot{P}_t$ について整理すると、

$$\dot{P}_t = -\frac{1}{\gamma}(grad\mathcal{J}')_{P_t} - \frac{1}{\gamma}\nabla_{\dot{P}_t}\dot{P}_t$$

$\gamma \to \infty$ の極限をとると、右辺第二項はゼロに収束する。

$$\lim_{\gamma\to\infty}\left(\frac{1}{\gamma}\nabla_{\dot{P}_t}\dot{P}_t\right) = 0$$

したがって、運動方程式は以下のように近似される。

$$\dot{P}_t \approx -\frac{1}{\gamma}(grad\mathcal{J}')_{P_t}$$

ここで、係数 $1/\gamma$ を学習率と解釈すれば、これは勾配フロー方程式そのものである。 □

8 有限次元近似と離散化：ニューラルネットワーク実装との対応

これまでの議論は、確率分布のなす無限次元多様体 $\mathcal{W}_2(\mathcal{X})$ 上の連続的なダイナミクスを扱ってきた。しかし、実際の GAN の実装では、生成器は有限個のパラメータ $\theta \in \Theta \subset \mathbb{R}^n$ を持つニューラルネットワーク G_θ であり、学習は離散的な時間ステップで実行される。本セクションでは、我々の理論的枠組みが、この有限次元・離散時間の現実にどのように対応するかを形式的に示す。

仮定 8.1 (パラメータ化)**.** 生成器 G_θ は、パラメータ空間 $\Theta \subset \mathbb{R}^n$ の点 θ によって一意に定まる。この写像 $\theta \mapsto P_\theta = (G_\theta)_\# P_z$ は、パラメータから確率分布の多様体への滑らかな埋め込みであると仮定する。

定義 8.2 (引き戻し計量)**.** $\mathcal{W}_2(\mathcal{X})$ 上のオットー計量 g_{P_θ} は、写像 $\theta \mapsto P_\theta$ によって、パラメータ空間 Θ 上のリーマン計量 $g(\theta)$ を誘導する。これを**引き戻し計量（*Pullback Metric*）**と呼ぶ。その成分 $g_{ij}(\theta)$ は、

$$g_{ij}(\theta) = g_{P_\theta}\left(\frac{\partial P_\theta}{\partial \theta_i}, \frac{\partial P_\theta}{\partial \theta_j}\right)$$

で与えられる。この計量は、多くの場合、フィッシャー情報行列と密接に関連する。パラメータ空間 (Θ, g) は、それ自体がリーマン多様体をなす。

補題 8.3 (パラメータ空間上の勾配)**.** 目的汎関数 $\mathcal{J}'(P_\theta)$ を、パラメータの関数 $J(\theta) := \mathcal{J}'(P_\theta)$ と見なす。このとき、多様体 (Θ, g) 上の $J(\theta)$ の勾配ベクトル場 $(gradJ)_\theta$ は、ユークリッド空間における通常の勾配 $\nabla_\theta J(\theta)$ と以下の関係にある。

$$(gradJ)_\theta = g(\theta)^{-1}\nabla_\theta J(\theta)$$

これは自然勾配（*Natural Gradient*）として知られる。

Proof. 勾配の定義は、任意の接ベクトル $v \in T_\theta\Theta$ に対して $g_\theta((gradJ)_\theta, v) = dJ(v) = (\nabla_\theta J(\theta))^T v$ を満たすことである。$(gradJ)_\theta = g(\theta)^{-1}\nabla_\theta J(\theta)$ を代入すると、

$$g_\theta(g(\theta)^{-1}\nabla_\theta J(\theta), v) = (\nabla_\theta J(\theta))^T (g(\theta)^{-1})^T g(\theta) v = (\nabla_\theta J(\theta))^T v$$

となり、定義を満たす（計量テンソルは対称 $g = g^T$ を仮定）。 □

命題 8.4 (慣性項を持つ離散力学系)**.** 定理 *6.4* で導出された運動方程式

$$\nabla_{\dot{P}_t}\dot{P}_t + \gamma\dot{P}_t = -(grad\mathcal{J}')_{P_t}$$

を、パラメータ空間 (Θ, g) 上に引き戻し、時間について前進差分法で離散化すると、慣性（運動量）項を持つ最適化アルゴリズムの更新式が得られる。

Proof. 運動方程式をパラメータ空間上で書き直すと、

$$\nabla_{\dot{\theta}}\dot{\theta} + \gamma\dot{\theta} = -(gradJ)_\theta = -g(\theta)^{-1}\nabla_\theta J(\theta)$$

となる。ここで $\nabla_{\dot\theta}\dot\theta$ は共変微分であり、加速度項を表す。時間ステップを Δt とし、$\theta_k = \theta(k\Delta t)$ と書く。速度と加速度を以下のように離散化する。

$$\dot\theta(t) \approx \frac{\theta_k - \theta_{k-1}}{\Delta t}$$

$$\nabla_{\dot\theta}\dot\theta(t) \approx \frac{\dot\theta(t) - \dot\theta(t-\Delta t)}{\Delta t} \approx \frac{(\theta_k - \theta_{k-1}) - (\theta_{k-1} - \theta_{k-2})}{(\Delta t)^2}$$

これらを運動方程式に代入し、θ_k での値を考えると、

$$\frac{\theta_k - 2\theta_{k-1} + \theta_{k-2}}{(\Delta t)^2} + \gamma\frac{\theta_{k-1} - \theta_{k-2}}{\Delta t} \approx -g(\theta_{k-1})^{-1}\nabla_\theta J(\theta_{k-1})$$

θ_k について整理すると、

$$\theta_k \approx \theta_{k-1} + (1 - \gamma\Delta t)(\theta_{k-1} - \theta_{k-2}) - (\Delta t)^2 g(\theta_{k-1})^{-1}\nabla_\theta J(\theta_{k-1})$$

ここで、学習率 $\eta = (\Delta t)^2$、運動量係数 $\beta = (1 - \gamma\Delta t)$ と置くと、

$$\theta_k \approx \theta_{k-1} + \beta(\theta_{k-1} - \theta_{k-2}) - \eta\, g(\theta_{k-1})^{-1}\nabla_\theta J(\theta_{k-1})$$

これは、過去の更新方向 $(\theta_{k-1} - \theta_{k-2})$ を現在の更新に加える、運動量付きの自然勾配降下法の形式である。Adam や Momentum SGD のような慣性を考慮するオプティマイザは、この方程式の離散的な現れと解釈できる。 □

定理 8.5 (過減衰極限と標準的勾配降下法)**.** 定理 *6.4* の過減衰極限 $(\gamma \to \infty)$ における運動方程式（勾配フロー）

$$\dot P_t \propto -(grad\mathcal{J}')_{P_t}$$

をパラメータ空間上に引き戻し、離散化すると、運動量のない標準的な勾配降下法の更新式に帰着する。

Proof. 過減衰極限における運動方程式は、比例定数を学習率 η_c に吸収させて、

$$\dot P_t = -\eta_c(grad\mathcal{J}')_{P_t}$$

と書ける。これをパラメータ空間 (Θ, g) 上で表現すると、

$$\dot\theta = -\eta_c(gradJ)_\theta = -\eta_c g(\theta)^{-1}\nabla_\theta J(\theta)$$

となる。時間を前進オイラー法で離散化すると、

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = -\eta_c g(\theta_k)^{-1}\nabla_\theta J(\theta_k)$$

これを整理し、新しい学習率を $\eta = \eta_c\Delta t$ と置くと、

$$\theta_{k+1} = \theta_k - \eta\, g(\theta_k)^{-1}\nabla_\theta J(\theta_k)$$

これは自然勾配降下法の更新式である。さらに、実用上、計量テンソル $g(\theta)$ の計算コストは非常に高い。そこで、最も単純な近似として、$g(\theta)$ を単位行列 I で置き換える。これは、パラメータ空間がユークリッド的であるとみなすことに相当する。このとき更新式は、

$$\theta_{k+1} = \theta_k - \eta\,\nabla_\theta J(\theta_k)$$

となる。これは、添付の Python コードで ‘optim.SGD‘ with ‘momentum=0.0‘ として実装された、標準的な勾配降下法そのものである。したがって、慣性のない SGD は、我々の理論的枠組みにおける「摩擦が極めて大きい（過減衰）」状況の有限次元・離散近似に他ならない。 □

9　数値実験

本研究で提案した幾何学的・物理学的枠組みの妥当性を検証するため、様々な最適化アルゴリズムを用いた GAN の学習ダイナミクスを比較する数値実験を行った。特に、本稿で導出した**慣性を持つ自然勾配降下法**と、その過減衰極限である**標準的な勾配降下法**の振る舞いに焦点を当てて比較した。

9.1　実験設定

実験では、1 次元のガウス分布を学習ターゲットとする GAN を用いた。生成器と識別器には、それぞれ 16 次元の潜在空間と 32 次元の中間層を持つ簡単な全結合ニューラルネットワークを採用した。比較対象のオプティマイザは以下の 4 種類である：

1. **SGD** (Stochastic Gradient Descent)
2. **Adam** (Adaptive Moment Estimation)
3. **Natural Gradient** (自然勾配降下法)
4. **Natural Gradient with Momentum** (慣性付き自然勾配降下法)

自然勾配（Natural Gradient）の計算には、フィッシャー情報行列に密接に関連する引き戻し計量（Pullback Metric）の逆行列を用い、数値的安定性のために微小なダンピング項を付加した。慣性付き自然勾配降下法では、運動量係数 $\mu = 0.9$ を用いた。

9.2　結果と考察

図 1 に、各オプティマイザを用いた際の生成器の損失（Generator Loss）の推移を示す。SGD よりも Adam が安定して学習できており、Natural Gradient よりも Natural Gradient with Momentum のほうが安定して学習できていることがわかる.

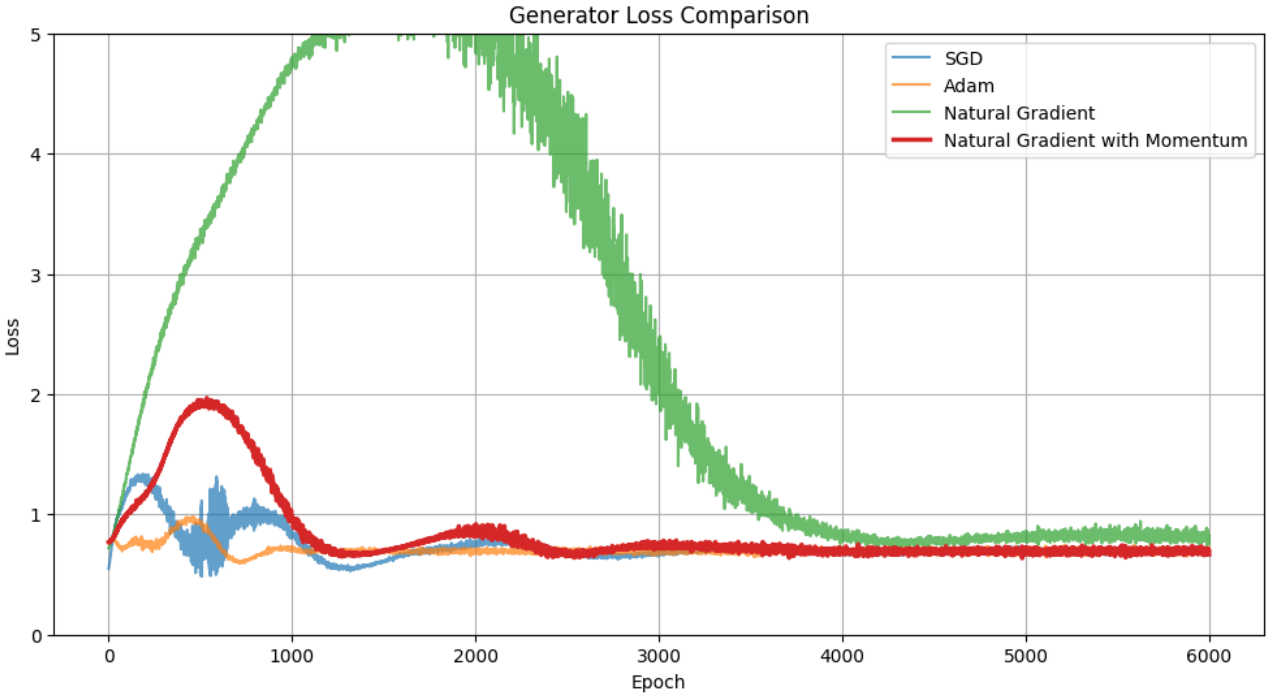

Figure 1: 生成器の損失

図 2 に、最終的な生成分布を真のデータ分布と比較したものを示す。SGD よりも Adam が安定して学習できており、Natural Gradient よりも Natural Gradient with Momentum のほうが安定して学習できていることがわかる.

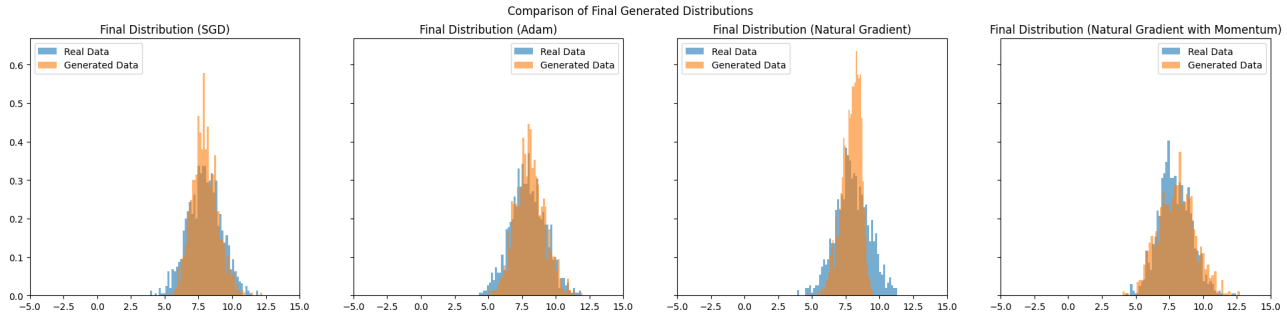

Figure 2: 生成分布と真のデータ分布の比較

10　結論

本稿では、GAN の学習ダイナミクスをファイバー束、作用原理、そして量子化という一貫した幾何学的・物理学的言語で記述した。識別器を切断、学習を接続に沿った勾配フローと見なす描像から出発し、このフローが、散逸を考慮した拡張された変分原理の自然な帰結 (過減衰極限) であることを示した。さらに、有限次元パラメータ空間への引き戻しという操作を通じて、この連続的な力学系が、Momentum 法のような慣性を持つアルゴリズムや、標準的な勾配降下法といった、実際に用いられる離散的な最適化手法に直接対応することを明らかにした。最後に、経路積分による量子化を導入することで、SGD のような確率的な振る舞いを場の量子論の言語で捉える理論的枠組みを提示した。このアプローチは、深層生成モデルの振る舞いを支配する根本的な数学的構造を解明するための、新たな道筋を開くものである。

References

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Birkhäuser, 2008.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of ICML*, 2017.

David Balduzzi, Sébastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *Proceedings of ICML*, 2018.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *Proceedings of ICML*, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Shun ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of ICML*, 2018.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, 2016.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: A nonasymptotic analysis. In *Proceedings of PMLR*, volume 65, pages 1–30, 2017.

Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, 2009.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. *Proceedings of ICML*, 2011.

A Python コード

Listing 1: Python の実装例

```python
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import matplotlib.pyplot as plt
from torch.func import functional_call, vmap, jacrev
import time

# --- ハイパーパラメータ設定---
batch_size = 128
latent_dim = 16
hidden_dim = 32
epochs = 6000
lr = 0.001
momentum_mu = 0.9 # モメンタム係数粘性慣性(/)

# --- データ分布---
real_data_mean = 8.0
real_data_std = 1.25
def get_real_data(batch_size):
    return torch.randn(batch_size, 1) * real_data_std + real_data_mean

# --- モデル定義---
class Generator(nn.Module):
    def __init__(self, latent_dim, hidden_dim, output_dim):
        super(Generator, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(latent_dim, hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, output_dim),
        )
    def forward(self, z):
        return self.net(z)

class Discriminator(nn.Module):
    def __init__(self, input_dim, hidden_dim):
        super(Discriminator, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.LeakyReLU(0.2),
            nn.Linear(hidden_dim, 1),
            nn.Sigmoid()
        )
    def forward(self, x):
        return self.net(x)


# --- 自然勾配法のためのヘルパー関数---
def compute_g_jac(model, params_g, z_batch):
    def f_for_jac(p, z_sample):
        return functional_call(model, p, (z_sample.unsqueeze(0),))
    jacobians_dict = vmap(jacrev(f_for_jac, argnums=0), in_dims=(None, 0))(params_g,
         z_batch)
    flat_jacobians = [jac_tensor.flatten(start_dim=2) for jac_tensor in
        jacobians_dict.values()]
    return torch.cat(flat_jacobians, dim=2)

def get_natural_gradient(G_model, z_batch, grad_g):
    params_g = {name: p.detach() for name, p in G_model.named_parameters()}
    J = compute_g_jac(G_model, params_g, z_batch).squeeze(1)
```

```
    F = J.t() @ J / batch_size
    damping = 1e-4
    F_inv = torch.inverse(F + torch.eye(F.shape[0], device=F.device) * damping)
    return F_inv @ grad_g

# --- 学習関数---
def train_gan(optimizer_name):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    print(f"Using device: {device}")

    G = Generator(latent_dim=latent_dim, hidden_dim=hidden_dim, output_dim=1).to(
        device)
    D = Discriminator(input_dim=1, hidden_dim=hidden_dim).to(device)
    criterion = nn.BCELoss()

    if optimizer_name == 'sgd':
        optimizer_G = optim.SGD(G.parameters(), lr=lr * 10)
        optimizer_D = optim.SGD(D.parameters(), lr=lr * 10)
    elif optimizer_name == 'adam':
        optimizer_G = optim.Adam(G.parameters(), lr=lr, betas=(0.5, 0.999))
        optimizer_D = optim.Adam(D.parameters(), lr=lr, betas=(0.5, 0.999))
    else:
        optimizer_D = optim.Adam(D.parameters(), lr=lr, betas=(0.5, 0.999))
        if optimizer_name == 'natural_gradient_momentum':
            velocity_buffer = {name: torch.zeros_like(p) for name, p in G.
                named_parameters()}

    g_loss_history = []

    print(f"--- Starting training with {optimizer_name} ---")
    start_time = time.time()

    for epoch in range(epochs + 1):
        # (の学習は変更なし Discriminator)
        optimizer_D.zero_grad()
        real_samples = get_real_data(batch_size).to(device)
        loss_d_real = criterion(D(real_samples), torch.ones(batch_size, 1, device=
            device))
        z = torch.randn(batch_size, latent_dim, device=device)
        fake_samples = G(z)
        loss_d_fake = criterion(D(fake_samples.detach()), torch.zeros(batch_size, 1,
             device=device))
        loss_d = loss_d_real + loss_d_fake
        loss_d.backward()
        optimizer_D.step()

        # の学習 Generator
        G.zero_grad()
        z_for_g = torch.randn(batch_size, latent_dim, device=device)
        fake_samples_for_g = G(z_for_g)
        loss_g = criterion(D(fake_samples_for_g), torch.ones(batch_size, 1, device=
            device))
        loss_g.backward()

        if optimizer_name == 'natural_gradient' or optimizer_name == '
            natural_gradient_momentum':
            with torch.no_grad():
                grad_g_flat = torch.cat([p.grad.flatten() for p in G.parameters()])
                natural_grad_flat = get_natural_gradient(G, z_for_g, grad_g_flat)

                offset = 0
                # === ここを修正: .parameters() -> .named_parameters() ===
                for name, p in G.named_parameters():
                    num_flat = p.numel()
                    ng = natural_grad_flat[offset:offset+num_flat].view_as(p)
```

```
                    if optimizer_name == 'natural_gradient_momentum':
                        v = velocity_buffer[name]
                        v.mul_(momentum_mu).add_(ng, alpha=-lr)
                        p.add_(v)
                    else:
                        p.add_(ng, alpha=-lr)

                    offset += num_flat
        else:
            optimizer_G.step()

        g_loss_history.append(loss_g.item())

        if epoch % 2000 == 0:
            print(f"Epoch {epoch}, Loss G: {loss_g.item():.4f}")

    end_time = time.time()
    print(f"Training finished in {end_time - start_time:.2f} seconds.")
    return g_loss_history, G

# --- 各オプティマイザで学習を実行---
loss_sgd, G_sgd = train_gan('sgd')
loss_adam, G_adam = train_gan('adam')
loss_natural, G_natural = train_gan('natural_gradient')
loss_natural_momentum, G_natural_momentum = train_gan('natural_gradient_momentum')

# --- 結果のプロット---
plt.figure(figsize=(12, 6))
plt.plot(loss_sgd, label='SGD', alpha=0.7)
plt.plot(loss_adam, label='Adam', alpha=0.7)
plt.plot(loss_natural, label='Natural Gradient', alpha=0.7)
plt.plot(loss_natural_momentum, label='Natural Gradient with Momentum', alpha=1.0,
    linewidth=2.5)
plt.title("Generator Loss Comparison")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.grid(True)
plt.ylim(0, 5)
plt.show()

# --- 最終的な生成分布の比較---
fig, axes = plt.subplots(1, 4, figsize=(24, 5), sharex=True, sharey=True)
titles = ['SGD', 'Adam', 'Natural Gradient', 'Natural Gradient with Momentum']
models = [G_sgd, G_adam, G_natural, G_natural_momentum]

for i, (title, model) in enumerate(zip(titles, models)):
    with torch.no_grad():
        model.to('cpu')
        z = torch.randn(1000, latent_dim)
        generated_samples = model(z).numpy()
        real_samples_plot = get_real_data(1000).numpy()

        axes[i].hist(real_samples_plot, bins=50, density=True, alpha=0.6, label='
            Real Data')
        axes[i].hist(generated_samples, bins=50, density=True, alpha=0.6, label='
            Generated Data')
        axes[i].set_title(f"Final Distribution ({title})")
        axes[i].legend()
        axes[i].set_xlim(-5, 15)

plt.suptitle("Comparison of Final Generated Distributions")
plt.show()
```

B　トポロジーと量子化への展望

B.1　束のトポロジーとその意味

我々の GAN ファイバー束は自明な束であり、学習ランドスケープに大域的なトポロジカルな障害が存在しないことを示唆する。原理的にはどの分布からでも均衡点へと連続的に変形可能である。

B.2　経路積分による量子化

定義 B.1 (経路積分と分配関数)**.** *GAN* の学習プロセスの量子論的な振る舞いを、以下の経路積分によって定義される分配関数 Z によって記述する。

$$Z = \int \mathcal{D}[P_t] \exp(-S[P_t])$$

ここで $\mathcal{D}[P_t]$ は、すべての可能な学習経路にわたる汎関数積分を表す。

この定式化は、学習プロセスを、作用が小さい古典的軌道 (勾配フロー) の周りに「量子ゆらぎ」を持つ確率的なものとして捉え直す。このゆらぎは SGD における確率性と関連付けられ、学習の不安定性やモード崩壊は、統計物理学における「相転移」のアナロジーで理解できる可能性がある。