ペイント可能なタイムラインによる対話的なハイパーパラメータ最適化

樋口啓太 * 佐野 正太郎 * 五十嵐 健夫 †*

概要. 本論文では，ハイパーパラメータの自動最適化システムにインタラクティブ性を持たせ，ユーザのパラメータ分布に関する事前知識を活用する手法を提案する．本手法では，ユーザは自動最適化の進捗状況を継続的に観察し，パラメータ空間のどこを探索するかを動的に指定する．本手法の実現可能性を示すために，最適化システムのためのインタラクティブダッシュボードのプロトタイプを実装した．インタラクティブなダッシュボードの主な特徴は「ペイント可能なタイムライン」であり，ユーザは標準的なタイムラインのように試行が行われた過去のパラメータ値を観察するだけでなく，簡単なペイント操作でパラメータの選択範囲を指定することができる．機械学習の専門家への聞き取りに基づき設計した，ユーザの介入によりパフォーマンスが向上すると考えられる 3 つのパラメータ最適化問題の例を紹介する．また，専門家を対象とした実験を実施し，対象となる問題のパラメータ分布に関する事前知識があれば，インタラクティブな最適化は完全な自動最適化と比較して良い結果を得ることができることを示した．

1 はじめに

ハイパーパラメータの最適化は，機械学習を用いたアプリケーションの開発において重要だが，時間と手間のかかる作業である．ディープニューラルネットワーク（DNN）のハイパーパラメータを最適化するには，数時間から数日かかることもあるため，この作業を高速化する手法があれば，実用上大きな価値がある．従来の最適化手法は，手動最適化と自動最適化の 2 つのアプローチに分けられる．自動最適化によって，開発者は試行錯誤から解放される．しかし，機械学習技術の専門家への聞き取り調査では，いくつかの理由により，自動最適化が手動最適化に完全に取って代わるには至っていないことがわかった [12]．第一に，パラメータ最適化では，最適化対象のパラメータ選定や探索範囲など探索空間が大きくなる．専門家がパラメータ最適化をより効率的に行うためには，対象となる問題のパラメータ分布に関する独自のドメイン知識を活用する必要がある．また，自動最適化を行うためには，対象となる探索空間が明確に定義されている必要がある． しかし，現実の最適化問題では，最適化結果の質，計算時間，計算コストなど，複数のトレードオフが存在することが多く，適切な探索空間が探索前に明確に定義されているとは限らない．そのため，開発者は探索を実行した後，中間結果を観察して探索空間を修正する必要があることが多い．

このような自動最適化の限界を解決するために，本研究では自動最適化システムにインタラクティブ性を取り入れた対話型ハイパーパラメータ最適化を提案する．このアプローチは，ユーザがハイパーパラメータ最適化プロセスを観察したり介入したりすることで，パラメータ分布に関するユーザのドメイン知識を活用することを目的としている．インタラクティブ最適化では，ユーザは自動最適化の進行状況を観察し，それに基づいて探索空間（どのパラメータ範囲を探索するか）を動的に変更できる．本研究では，対話型最適化の実現可能性と有効性を示すために，対話型ダッシュボードというプロトタイプを開発した．このダッシュボードは，最適化システムのブラウザベースのフロントエンドとして機能する．このシステムの主な特徴は,「ペイント可能なタイムライン」である．このタイムラインは，既存の最適化システムで最適化プロセスの進捗状況を示す典型的な可視化に似ているが，それに加えて探索空間をリアルタイムに設定できる機能を備えている．ユーザは，対象となる問題に関する事前知識があれば，過去の試行結果から最適解が存在する可能性の高い領域を推定し，今後の試行ではその領域に注目するようシステムに指示することができる．その結果，システムは有望な探索空間により多くの時間を費やすことができ，最終的な最適化結果をより向上させることができ得る．

2 関連研究

2.1 自動パラメータ最適化のフレームワーク

ハイパーパラメータ最適化とは，目的関数の値を最大化する最適なハイパーパラメータを求める問題である．自動最適化システムでは，パラメータのサンプリング戦略（ベイジアン，ランダム，進化計算など）を用いてユーザが定義したパラメータ空間を探索する [22, 4, 5]）．また，近年の自動最適化システムでは，最適化コードを記述することなく，目標パラメータと目的関数を定義するだけで最適化を実行で

* Preferred Networks
† 東京大学

きるフレームワークが提供されている [1, 11, 6]．これらのフレームワークは，機械学習アプリケーションの開発だけでなく，シミュレーションや材料解析などにも利用されている [7, 20, 16, 19] しかし，これらのフレームワークでは，探索中のユーザの介入が制限されている．本研究では，探索中にユーザがパラメータ探索範囲を変更できるようにし，専門家のドメイン知識を活用して探索を効率化する手法を実現した．

2.2 AutoML への人間の介入

AutoML とは機械学習モデルの構築や学習を自動化しようとするものである。AutoML は，機械学習の専門家でなくても利用できることを目指して開発されており，先行研究では，AutoML システムへの人間の介入方法が検討されている．Gil らは Human-Guided 機械学習を提案し，それに適したタスクや要件を検討している [10, 9]．Santos らは，機械学習に不慣れな専門家のための対話型 AutoML システム Visus を開発した [21]．また，AutoML システムのヒューマンファクタを調査した結果，可視化の重要性やユーザと AI の協調の重要性が明らかになった．本研究では，ハイパーパラメータ最適化に着目し，そのプロセスに人間が介在する方法を提案している．

2.3 インタラクティブビジュアライゼーション

本研究は，機械学習システムの制御のために設計された対話的な可視化技術にヒントを得ている．ManiMatrix では，ユーザが混同行列を直接操作して分類器の内部パラメータを調整することができる [14]．Dis-function は，ユーザが散布図を直接操作して，距離関数を調整することができる [8]．彼らは，本研究と同様に，システムの監視と制御のために可視化を使用している．また，既存のシステムがスナップショットの制御を主目的としているのに対し，本研究ではタイムライン型の可視化を活用することで，過去の試行のモニタリングと未来の試行の制御をシームレスに統合するアイデアを提案している．

本研究は Klau らによる Human-Guided Search の影響も受けており [15]，HuGS では最適化問題への人間の介入の重要性を強調し，組み合わせ最適化を解くための視覚的なメタファーを提供した．彼らは，そのアプローチの有効性を 4 つの事例（グラフレイアウト，経路探索，タンパク質フォールディング，タスクスケジューリング）で実証した．本研究は同じ最適化への人間の介入に着目しているが，ハイパーパラメータ（ブラックボックス）最適化のための技術を提案している．

ペイントやスケッチは，データとのインタラクションによく使われる．Baudisch は時間間隔を指定するためにペイントを使用した [3]．Hochheiser と Shneiderman はラバーバンドを，Mannino と Azza はフリーハンドのスケッチを時系列データへのクエリとして用いている [13, 17]．Mannino と Azza は人工的なデータの合成にもスケッチを用いている [18]．Asai ら は散布図へのスケッチとペイントをデータ合成と編集に用いている [2]．本研究の提案インタラクションは，過去の最適化結果を観察しながら，未来のパラメータ探索範囲を制御するためにペイントを使用するという点で新規性がある．

3 提案システム

本研究では，専門家へのインタビューを行い，対話型パラメータ最適化の要件と望ましい機能を明らかにした [12]．その結果，パラメータの範囲に関する専門家の予備知識は重要であるが，それを活用して効率的なパラメータチューニングを実現することは容易ではないことがわかった．

このセクションでは，専門家へのインタビューで明らかになった完全自動最適化の限界を解決するために設計したインタラクティブダッシュボードについて説明する．専門家は，パラメータ分布に関する事前の知識を活用することが重要であると指摘したが，既存のフレームワークではパラメータ探索範囲を動的に変更する機能が限られていた．そこで本研究では，ハイパーパラメータ最適化フレームワークに対話的な機能を導入する．この本提案は，ユーザが最適化の進捗状況を観察するだけでなく，最適化プロセスをその場で制御することができる，対話的な最適化をサポートしている．ユーザは，探索の詳細なスケジュール（パラメータ空間のどの領域をいつ探索する必要があるか）を探索前に設定することができ，また，探索中にスケジュールを動的に変更することもできる．

3.1 可視化

インタラクティブダッシュボードは，ハイパーパラメータチューニングシステムのフロントエンドとして機能する．現在の実装は，特定のハイパーパラメータ探索フレームワーク（Optuna）用に設計されているが，基本的なアイデアは他のフレームワークにも適用できるはずである．図 1 に本システムのスクリーンショットを示す．主に，ノブ付きのスケジュールパネル，スコアパネル，パラメータパネルで構成されている．スコアパネルは，一般的なダッシュボードのタイムラインビューに似ている．過去の試行のスコアを可視化し，最適化の進捗状況を示す．スケジュールパネルは，過去と未来の最後までの試行回数を表示する．ノブを使って，ハイパーパラメータ最適化の試行回数を設定・追加することができる．パラメータパネルは，ペイント可能なタイムライン機能を持つ本システムの主要部分である．過去にどのようなパラメータ値が検討され，将来ど

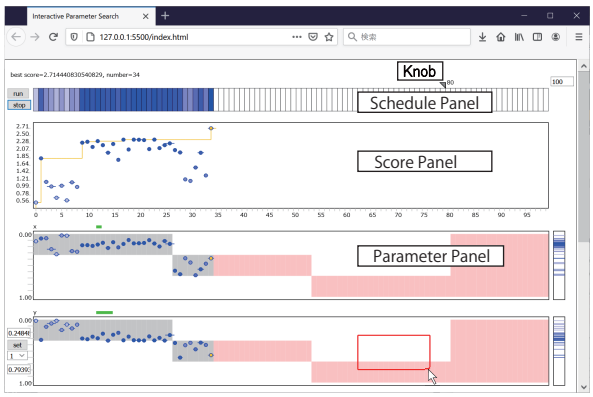

図 1. プロトタイプシステムのスクリーンショット.

のような値が検討されるかを可視化する．また，将来的にどのようなパラメータ値を検討しなければならないかを簡単に設定することができる．また，パラメータパネルには，集約されたビュー（右）とパラメータの推定重要度（左上）が表示される．パラメータパネルでは，高次元のパラメータも扱うことができる．各次元（パラメータ）は，下部の水平パネルで表現され，任意の数の次元を追加することができる．

ペイント可能なタイムラインは，左から順に，過去，現在，未来の試行を表示する．過去のトライアルでは，テストされたパラメータ値とそのスコアが可視化される．また，現在のトライアルでは，テストされているがスコアのないパラメータ値が可視化される．トライアルのスコア，パラメータの範囲，選択された値は，縦方向に配置される．スコアパネルはスコアを，パラメータパネルは選択されたパラメータを，それぞれ円で表現する．例えば，図 1 では，最初のトライアルで，スコアが 0.56，パラメータ x が 0.01，パラメータ y が 0.00 であることを示している．円の色は，各試行の相対的なスコアを示しており，色が濃いほどスコアが良いことを示している．最良のスコアを得た試行の結果は，黄色の円で表される．未来の試行では，検討すべきパラメータ値の範囲をピンクの帯で可視化している．スコアとパラメータのパネルでは，円の後ろにある短い線が，相対的な一試行あたりの計算時間を示している．

3.2 ユーザインタラクション

ペイント可能なタイムラインは，パラメータ探索範囲を可視化するだけでなく，ペイント操作により探索範囲を変更できる．現在の実装では，3 種類のペイント操作をサポートしている．一つ目は，空白領域から始めるラバーバンド操作である（図 2 左）．ユーザがタイムライン上でドラッグ操作を開始すると，システムはラバーバンドを表示する．ユーザは，輪ゴムの境界をドラッグするか，下限値と上限値を数値で入力して，ボックスの位置と大きさを調整する．次に，set ボタンを押すと，探索するパラメータの範囲を設定することができる．また，グリッド数を指定することで，指定した領域を分割してグリッド探索を行うこともできる．2 つ目の操作は，パラメータ範囲の境界をドラッグすることである（Fig 2 中）．境界線にマウスカーソルを合わせると，境界線がハイライトされる．ユーザはマウスを下にクリックして境界をつかみ，任意の場所にドラッグする．その後，ユーザは “set” を押して，パラメータの範囲を更新する．3 つ目は，過去の試行の選択である (図 2 右)．ユーザは，ラバーバンドを使って 1 つまたは複数の試行を選択する．すると，システムは，選択された試行から下界と上界をとったラバーバンドを表示する．これは，過去の試行に基づいて，スコアの高い領域や低い領域を指定することができる．

ペイント可能なタイムラインは，ユーザのドメイン知識や判断を活用して，探索を効率化するように設計されている．図 3 に提案システムを用いたパラメータ最適化の様子を示す．まず，最適化の際に，ユーザはペイント可能なタイムラインを利用して探索空間を徐々に狭めていくことができる． 過去の試行結果やスコアの推定結果を見ながら，どの範囲を探索すれば効率的な探索ができるかを判断することができる．これは，分布が滑らかであり，与えられた範囲内に単一の最適値があることをユーザが知っている場合に有効である．次に，パラメータ値を変えてみる必要がなくなったと判断した場合には，探索中にパラメータ値を固定することができる．このようにして，自動最適化は他のパラメータに集中することができる．3 つ目は，ユーザが，どのパラメータ空間をいつ探索するかを指定することで，今後の探索のスケジュールを立てることである．例えば，ある特定のパラメータ値の組み合わせが有望であることがわかっている場合（例：パラメータ X が高いときはパラメータ Y も高いはず），ペイント可能なタイムラインを使って特定の領域を探索させることもできるし，半自動のグリッド探索に使うこともできる． ユーザは，ペイント可能なタイムラインを使って，グリッドセルを指定する．すると，システムは各グリッドセル内を探索する．これは，一様に探索を行う必要があることがわかっている場合に有効である．第 4 に，スコアパネルの過去の試行をラバーバンドで選択し，選択した試行のパラメータの範囲を設定することができる． これは，数回の試行を経て，有望なパラメータの範囲に探索を集中させるのに有効である．

3.3 実装

本研究では，ブラックボックス最適化フレームワークである Optuna [1] を用いてプロトタイプシステ

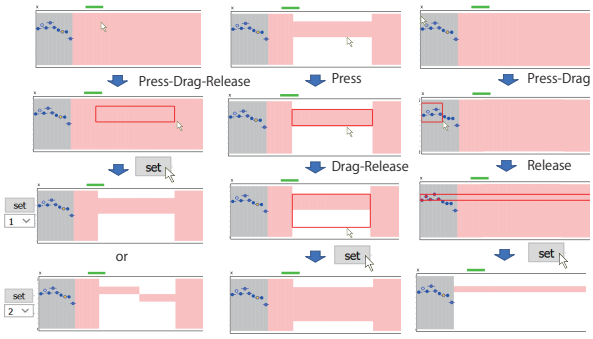

図 2. 実装されているペイント操作．左: 矩形を描く．中: 境界を移動する．右: 過去の試行を選択する．

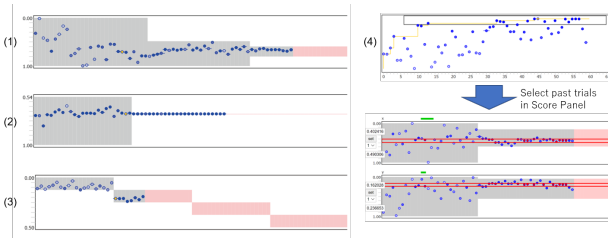

図 3. (1) 探索空間を狭める (2) パラメータの固定 (3) グリッド探索 (4) 過去試行の選択

ムを実装した．このフレームワークでは，ハイパーパラメータ候補を自動的にサンプリングするための複数のアルゴリズムを Sampler としてサポートしており，これを拡張してユーザ定義のサンプリングアルゴリズムを実装することができる．本研究ではインタラクティブなハイパーパラメータ最適化のバックエンドシステムとして，*interactive sampler* を開発した．フロントエンドとバックエンドは WebSocket 介してパラメータ範囲や最適化結果などを通信する。この実装では，ユーザがインタラクティブハイパーパラメータ最適化を利用するために，オリジナルのコードを大幅に変更する必要はないという利点がある．

4 利用シナリオ

このセクションでは，インタラクティブな最適化の利用例を 3 つ紹介する．専門家へのインタビューで語られた代表的な事例をもとに例題を作成した．スコア分布の事前知識を用いた対話的な最適化が完全な自動探索よりも優れていると考えられる 3 つのケースをピックアップし，事前知識を活用して探索を誘導するためのペイント可能なタイムラインの使用方法を説明する．3 つの例とも，操作画面のスクリーンショットは実際のインタラクティブな最適化の結果を示している．

ここでは，ユーザが対象となる最適化問題のスコア分布（最適解がどこに分布しているか）について何らかの事前知識を持っていることを前提としている．この事前知識は，問題の構造に関する既存のドメイン固有の知識から得られるものと，類似の問題に対する過去の最適化の経験から得られるものがある．また，ユーザは，スコア分布は共通しているが，実際の最適スコアやそのパラメータ値が異なる類似の問題に繰り返し取り組むことを想定している．ユーザは，インタラクティブな最適化の方法を徐々に学び，その経験を活用して，より効率的な探索を実現する．

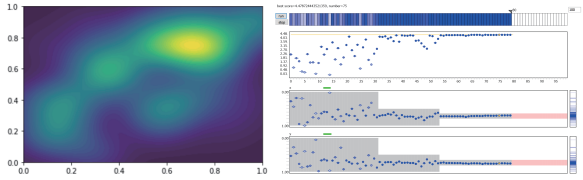

図 4. 例 1: 対角線上の分布

4.1 例 1: 対角線上の分布

最初の例では，パラメータ間に相関があり，最適解が対角線領域の近くに位置する場合を示している．図 4）． この例ではパラメータ x が小さければ，パラメータ y も小さくなるはずであり．パラメータ x が大きければ，パラメータ y も大きいはずであるという事前知識がある．この例は，DNN の学習では、学習率とバッチサイズに相関関係があることが知られていることから考案した（学習率が大きいときは、バッチサイズも大きくする必要がある） [23]．探索空間を対角線付近に限定することで，効率的な探索を行うことができる．図 4 にスコア分布とインタラクティブ最適化の結果を示す．ユーザはまず，自動最適化に任せてパラメータ空間全体を探索する．しばらくして，30 回目くらいの試行で，x と y の両方が高いところでは，おおむねスコアが高いことがわかってくる．続いて，ペイント可能なタイムラインを使って探索空間を絞り込む．スコアの高いサンプルが集中していることを観察しながら，さらに探索空間を絞り込むことができる．その結果，自動最適化よりも対話型最適化の方が，最も有望な領域を効率的に探索することができるため，高いスコアが得られることがある．

4.2 例 2: 支配的パラメータ

2 番目の例は，パラメータ x が支配的で，パラメータ y のスコアへの影響が限定的な場合である（図 5）．この場合，まず最適なパラメータ x を探索し，次にパラメータ x を固定して最適なパラメータ y を探索するのが最適な戦略となる．図 5 にスコア分布と対話型最適化の結果を示す．ユーザはま

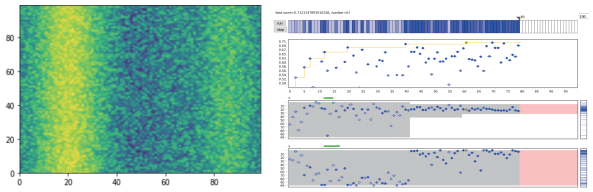

図 5. 例 2: 支配的パラメータ

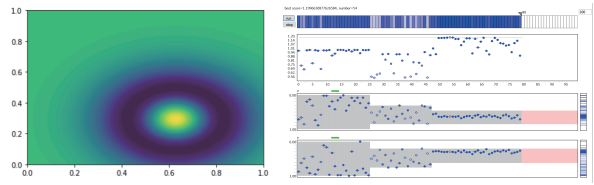

図 6. 例 3: 孤立領域

ず，自動最適化に任せてパラメータ空間全体を探索する．しばらくして，40 回目くらいの試行で，あるパラメータ x の値でスコアが全般的に高くなることが明らかになった．その結果，自動最適化よりも対話型最適化の方が，有望でない領域を無駄に探さずに済むため，高いスコアが得られる可能性がある．

4.3 例 3: 孤立領域

3 つ目の例は，最高スコアが低スコア領域の中央に位置するケースである（図 6）．このような場合，自動最適化では，低スコアの周辺にある高スコアの領域（平地）に焦点を当て，低スコアの領域を無視することが多いため，特に難しい．しかし，このような予備知識があれば，意図的に低スコア領域を探索することで最適解を得ることができる．図 6 にこの例のスコア分布と対話型最適化の結果を示す．ユーザはまず，自動最適化にパラメータ空間全体を探索させる．しばらくして，試行回数 30 回目くらいになると，ある領域でスコアが全体的に低くなっている．そこで，ペイント可能なタイムラインを使って，スコアの低い領域を探索するように指示する．最終的には，孤立した島を見つける．これは，完全な自動探索では難しいことである．

5 ユーザ実験

本研究では提案手法であるペイント可能なタイムラインを用いたインタラクティブな最適化の効果を検証するために，ユーザ実験を実施した．本実験の目的は，スコア分布に関する事前知識を用いたインタラクティブな最適化が，いくつかの特定ケースにおいて自動最適化よりも優れているかどうかを確認することである．

5.1 仮定

ユーザ実験のシナリオとして，2 つの重要な前提を置いている．第一に，ユーザは対象となる問題タイプにすでに精通しており，何度か取り組んだことがあると仮定する．具体的には，スコアの全体的な分布特性や，効率的な探索のための対話型最適化の際の探索範囲の絞り込み方などをすでに知っている．理想的には，研究参加者に何度もその問題タイプに取り組んでもらい，自分で戦略を見つけてもらうのがベストである．しかし，ユーザ実験としてはかなりの時間を要するため，分布と探索戦略を参加者に教えている．

次に，探索には数時間から数日といった相応の時間がかかることを想定している．自動探索で数秒から数分で最適解が見つかるのであれば，ユーザの介入は必要ないかもしれない．インタラクティブな最適化が有効になるのは，全体の探索時間が対話時間よりもはるかに長い場合である．理想的には，最適化に数時間から数日かかるような問題を研究対象とするのがベストである．しかし，ユーザ実験としてはかなりの時間を要するため，1 分以内に終わるような小さな最適化問題を大きな問題の代用としている．参加者には，何時間も何日もかかる問題が前提なので，ユーザとの対話の時間を最短にしようとする必要はないと伝えている．より良いスコアを見つけることは，インタラクションの時間を短くすることよりも重要であると伝えた．

5.2 実験条件

この研究には 7 人の機械学習の専門家が参加した．参加者の数が限られているのは，手動および自動最適化を用いたハイパーパラメータチューニングに十分な経験を持つ専門家が必要だからである．参加者は全員，著書の所属する社内から協力を得たエンジニアもしくはリサーチャーである，7 名のうち 5 名は仕事の一環としてハイパーパラメータチューニングに取り組んでいる．彼らは全員，事前の専門家インタビューに参加している．すべてのユーザ実験のセッションは，オンラインで行われた．各参加者は，3 種類の問題からなる 15 のインタラクティブ最適化タスクに取り組み，それぞれの問題には 5 つのバリエーションがあった．この 3 つのタイプは，4 章で示した例から取られている．各タイプの問題には，ランダムに 5 つのバリエーションを用意し，参加者全員に同じセットを使用した．参加者が与えられたタスクを完了するのに約 1 時間かかった．

まず，参加者に提案システムの使い方を説明し，UI 利用の練習をしてもらった．その後，問題の種類ごとに 1 つずつ取り組んでもらった．問題の種類ごとに，探索の誘導方法をデモで説明し，参加者は練習として同じ問題に取り組んだ．説明は口頭で行い，スコア分布の可視化は見せなかった．

参加者は説明を受けた後，メインタスクとして 15 の課題に取り組んだ．参加者は課題中にいつでも説明を読むことができた．5 つのバリエーションは，一般的な特徴は共通しているが，実際の最適なスコアや場所は参加者にはわからないようになっている．そのため，同じ探索戦略を用いても，試行の結果を見て，探索範囲をどこに設定するかを決める必要がある．

最適化の試行回数（パラメータのサンプリング回数）80 回の内で最高スコアを獲得してもらい，その結果を提案手法の interactive 条件とした．7 人の実験参加者が，1 つの問題タイプにつき 5 つのバリエーションに取り組むため，3 種類のタイプそれぞれに 35 の最適化結果が得られた．また，ベースラインとして，TPE アルゴリズムによりパラメータ探索を行った auto 条件を採用した． 80 回イテレーションの自動最適化を 5 つのバリエーションごとに 7 回行っているので，interactive 条件の場合と同様に，3 種類のタイプごとに 35 個のスコア結果が得られた．両条件とも最初は同じパラメータ範囲を使用しているが，対話型最適化では探索中に参加者が自由にパラメータ範囲を変更した．参加者には，最初の範囲内にベストスコアがあることを伝えた．また，十分に細かいグリッド（1000×1000）でグリッドサーチを行い，その結果を真の最適値とした．手動探索と自動探索の性能は，それぞれのスコアを対応する真の最適値で割ることで測定する．また，interactive 条件では，80 試行中の介入時間を測定した．

5.3 結果

図 7 は，ユーザ実験の最適化結果（ 得られたスコア/最適値 ）を示している．1 つ目のタイプでは，auto 条件と interactive 条件の平均性能は，それぞれ 0.931（SD：0.134），0.964（SD：0.104）となっている．2 つ目のタイプでは，auto 条件と interactive 条件の平均性能は，それぞれ 0.990（SD：0.007），0.993（SD：0.005）であった．3 つ目のタイプでは，auto 条件と interactive 条件でそれぞれ 0.859（SD：0.044），0.940（SD：0.070）となった．Shapiro–Wilk 検定ではすべての結果に正規性が認められなかったため，Mann–Whitney U 検定を用いて両条件での有意性を観察した．また，多重比較のために Benjamini-Hochberg 法を用いて p 値を調整した．検定の結果，すべての機能タイプで条件間に有意な差が見られた．interactive 条件における介入時間の平均は，第 1，第 2，第 3 のタイプで，それぞれ 100.8 秒（SD：58.0），54.7 秒（SD：30.6），114.9 秒（SD：129.6）であった．

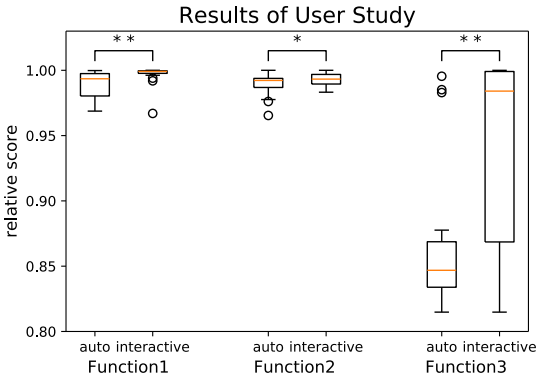

図 7. 実験結果．$*(p < 0.05)$ および $**(p < 0.001)$ は Mann–Whitney U test における統計的有意差を示す．ここに示す相対スコアは，最適スコアをグリッドサーチで探した最適値で割ったものである．

5.4 議論

ユーザ実験の結果，3 つの具体的なケースにおいて，適切な探索戦略を用いたインタラクティブな最適化が自動最適化を上回ることが実証された．すべてのタスクにおいて，対話型の interactive 条件は自動最適化の auto 条件よりも高い平均性能を達成した．利用シナリオで説明したように，3 つの機能タイプでは，事前のドメイン知識が役立つことが予想された．特に，1 番目と 3 番目のタイプは，局所最適領域が大きいため，完全な自動最適化による探索には難しい問題だった．参加者は，説明から得た知識を利用して，パラメータの探索範囲を制御することに成功したことが観察された．このように，ユーザ実験では我々の予想が確認された結果となった．今回の結果は，インタラクティブ最適化が多くの問題において有効だと示したものではない。今後より広範な実験によりインタラクティブ最適化がどのような問題に有用であるかを確認したい．

6 むすび

本研究では，対話型のハイパーパラメータ最適化を提案し，インタラクティブダッシュボードのプロトタイプを開発した．機械学習の専門家へのインタビューをもとに，最適化の進行状況を観察したり，ペイント操作でパラメータ探索範囲に介入できる機能を設計した．また，インタラクティブな最適化が効果的であると思われる 3 つの例を示した．インタラクティブなハイパーパラメータ最適化の実現可能性を示すために，7 人の専門家を対象にユーザ実験を実施した．その結果，3 つの具体的な事例において，適切な探索戦略を用いた対話型最適化が完全な自動最適化よりも優れていることが示された．

参考文献

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, p. 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery.

[2] K. Asai, T. Fukusato, and T. Igarashi. Integrated Development Environment with Interactive Scatter Plot for Examining Statistical Modeling. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–7, New York, NY, USA, 2020. Association for Computing Machinery.

[3] P. Baudisch. Don't Click, Paint! Using Toggle Maps to Manipulate Sets of Toggle Switches. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, p. 65–66, New York, NY, USA, 1998. Association for Computing Machinery.

[4] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, p. 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc.

[5] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13(null):281–305, Feb. 2012.

[6] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, jul 2015.

[7] J. Bergstra, D. Yamins, and D. D. Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. ICML'13, p. I–115–I–123. JMLR.org, 2013.

[8] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 83–92. IEEE, 2012.

[9] Y. Gil, J. Honaker, S. Gupta, Y. Ma, V. D'Orazio, D. Garijo, S. Gadewar, Q. Yang, and N. Jahanshad. Towards Human-Guided Machine Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, p. 614–624, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Y. Gil, K.-T. Yao, V. Ratnakar, D. Garijo, G. V. Steeg, P. Szekely, R. Brekelmans, M. Kejriwal, F. Luo, and I.-H. Huang. P4ML: A Phased Performance-Based Pipeline Planner for Automated Machine Learning. In *Proceedings of Machine Learning Research, ICML 2018 AutoML Workshop*, 2018.

[11] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. Google Vizier: A Service for Black-Box Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, p. 1487–1495, New York, NY, USA, 2017. Association for Computing Machinery.

[12] K. Higuchi, S. Sano, and T. Igarashi. Interactive Hyperparameter Optimization with Paintable Timelines. In *Designing Interactive Systems Conference 2021*, DIS '21, p. 1518–1528, New York, NY, USA, 2021. Association for Computing Machinery.

[13] H. Hochheiser and B. Shneiderman. Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization*, 3(1):1–18, 2004.

[14] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Interactive Optimization for Steering Machine Classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, p. 1343–1352, New York, NY, USA, 2010. Association for Computing Machinery.

[15] G. W. Klau, N. Lesh, J. Marks, M. Mitzenmacher, and G. T. Schafer. The HuGS Platform: A Toolkit for Interactive Optimization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, p. 324–330, New York, NY, USA, 2002. Association for Computing Machinery.

[16] T. Ko. A Tendon-Driven Robot Gripper With Passively Switchable Underactuated Surface and its Physics Simulation Based Parameter Optimization. *IEEE Robotics and Automation Letters*, 5(4):5002–5009, 2020.

[17] M. Mannino and A. Abouzied. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[18] M. Mannino and A. Abouzied. Is This Real? Generating Synthetic Data That Looks Real. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 549–561, New York, NY, USA, 2019. Association for Computing Machinery.

[19] K. Oono and T. Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2019.

[20] Y. Ozaki, Y. Suzuki, T. Hawai, K. Saito, M. Onishi, and K. Ono. Automated crystal structure analysis based on blackbox optimisation. *npj Computational Materials*, 6(1):1–7, 2020.

[21] A. Santos, S. Castelo, C. Felix, J. P. Ono, B. Yu, S. R. Hong, C. T. Silva, E. Bertini, and J. Freire. Visus: An Interactive System for Automatic Machine Learning Model Building and Curation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA'19, New York, NY, USA, 2019. Association for Computing Machinery.

[22] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[23] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le. Don't Decay the Learning Rate, Increase the Batch Size. In *International Conference on Learning Representations*, 2018.