

プロンプトエンジニアリングについて

プロンプトエンジニアリングについて

stable diffusionって何？ 小学生にもわかりやすく Claude3 Opus + 参考文献

stable diffusionは、「かわいいウサギ」などの言葉を入れると、その通りの絵を描いてくれるすごいAIです。

言葉を入れるだけで、すぐに絵を描いてくれるんですよ。

stable diffusionはどうやって絵を描くの？

stable diffusionは、入れた言葉をまずコンピュータが理解できる数字に変えます。

その数字を使って、言葉通りの絵を表す別の数字を作ります。

最後にその数字から、きれいな絵を作り出します。

数字を使うのは、絵をいきなり描くよりも簡単だからなんです。

だから、stable diffusionは数字を使って絵を作るんですって。

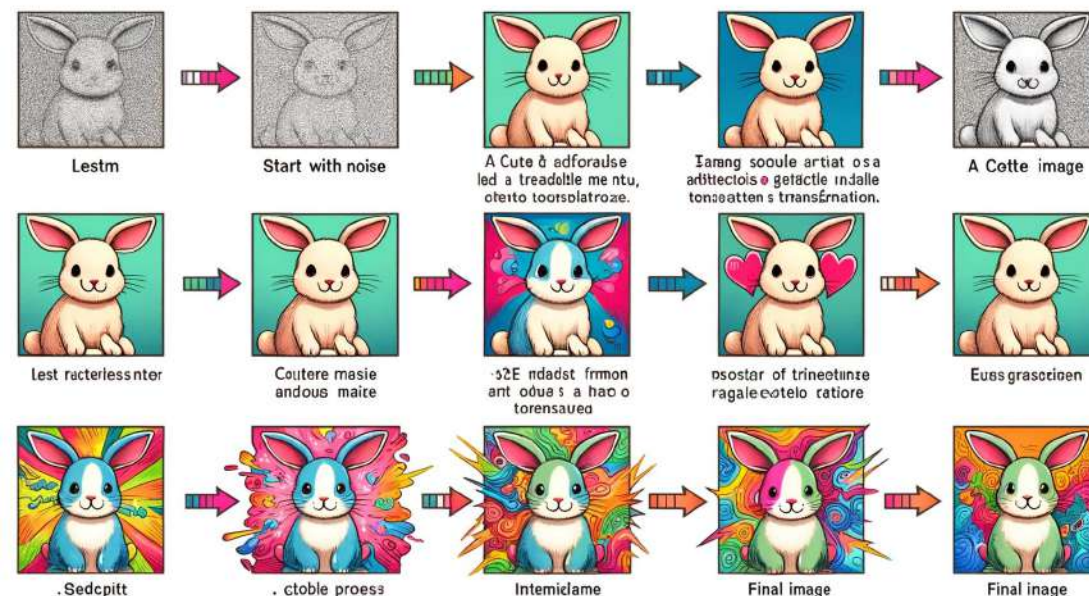
絵を作る手順は3つあります。

1. 言葉を数字に変える
2. その数字を使って、絵の数字をだんだん良くしていく
3. 良くなった数字から、絵を作り出す

言葉をどう書くと絵が変わるの？例えば、最初は「かわいいウサギ」と入れたとします。

でも、思った通りの絵じゃなかったら、「ピクサー映画に出てきそうな」といった言葉を足したり、

逆に言葉を減らしたりして、何度も試してみるといいみたいです。



Diffusion Explainer

<https://poloclub.github.io/diffusion-explainer/>

stable diffusionって何？ 中学生にもわかりやすく Claude3 Opus + 参考文献

stable diffusionとは stable diffusionは、言葉で表現された内容を高解像度の画像に変換する AIモデルです。

例えば、「かわいくて愛らしいウサギ」と入力すると、数秒でその通りの絵を描いてくれます。

stable diffusionの仕組み: stable diffusionは、まず入力された言葉を数値化します。この数値は「テキスト表現」と呼ばれ、言葉の内容を要約したものです。

次に、このテキスト表現を使って、言葉で表現された絵の数値「画像表現」を作ります。最後に、この画像表現を高解像度の画像に変換します。

画像表現を使うのは、いきなり高解像度の画像を作るよりも計算が速くて済むからです。画質を保ちつつ、効率よく画像を生成できるんですね。

画像表現は、最初はランダムなノイズ (ザラザラした画像) から始まります。それを少しずつ改善していき、

最終的に言葉通りのきれいな画像になるように調整するんです。この調整のステップを「タイムステップ」と呼び、通常は 50回か100回繰り返します。

stable diffusionの画像生成は、大きく3つの段階に分けられます。

1. テキスト表現の生成: 言葉を数値化します。
2. 画像表現の改善: ランダムなノイズから始めて、言葉の内容を反映するように画像表現をタイムステップごとに改善します。
3. 画像の生成: 改善された画像表現を高解像度の画像に変換します。

プロンプトの言葉選びのコツ stable diffusionに入力する言葉のことを「プロンプト」と言います。思い通りの絵を作るには、プロンプトの言葉選びが大切です。最初は基本的な言葉で試して、徐々に細かい表現を加えたり、不要な言葉を減らしたりしながら、理想の絵に近づけていくといいでしょう。

プロンプトの言葉を変えると、絵がどう変わるか観察してみるのも面白いですよ。ほんの少しの言葉の違いで、絵の印象はガラリと変わります。

stable diffusionの調整項目 stable diffusionでは、プロンプト以外にも、生成される画像を調整できる項目がいくつかあります。

例えば、「シード」と呼ばれる項目では、最初のランダムなノイズを変更できます。また「ガイダンススケール」では、言葉通りの絵を作る強さを調整できます。

例えば、みなさんがお絵描きアプリで絵を描くとして、最初は真っ白な画面から始めますよね。

でも、Stable Diffusionは、最初から少しザラザラしたランダムな模様(ノイズ)が描かれた状態から始めるんです。

そして、みなさんがお題を決めます。例えば、「夕日をバックに、浜辺を走る犬」というお題にしましょう。

このお題を、Stable Diffusionにテキストで入力します。

Stable Diffusionは、このお題を理解して、ザラザラした模様を少しずつ変化させていきます。最初は、ザラザラした模様の中に、夕日らしきオレンジ色の部分が現れるかもしれませんが、次に、浜辺らしき砂浜の色が加わります。そして、犬らしき形が徐々に浮かび上がってきます。

この変化を、50回くらい繰り返していくんです。そのたびに、絵はお題により近づいていきます。ザラザラしていた模様が、だんだんと「夕日をバックに浜辺を走る犬」の絵に変わっていくんですね。

最後には、お題通りのきれいな絵が完成します。最初は本当にザラザラしたノイズだったのが信じられないくらい、リアルで素敵な絵になるんです。

この一連の過程で、Stable Diffusionはノイズを少しずつ取り除きながら、お題に合った絵を作り上げているんです。だから、お題を変えると、また違う絵ができあがります。

例えば、お題を「宇宙飛行士が月面に立っている」に変えると、Stable Diffusionはその通りの絵を描いてくれます。最初のザラザラした模様から、月面の風景と宇宙飛行士が浮かび上がってくるんです。

このように、Stable Diffusionは、私たちが入力したテキストを理解して、ノイズを使って、そのテキストにぴったりの絵を描いてくれるすごい技術なんですよ。

Stable Diffusionの学習データの作り方についても説明します。

まず最初に、インターネット上から大量の画像を集めます。

そこには、写真や絵画、イラストなど、様々な種類の画像が含まれています。

次に、集めた画像一つ一つに、人間が手作業で説明文をつけていきます。

例えば、犬の写真には「茶色い犬が草原を走っている」といった具合です。

この作業を、大勢の人に手伝ってもらって進めます。

ここで重要なのは、説明文をつける人は、画像に写っている内容を正確に表現するように心がけるということです。

説明文が正確でないと、Stable Diffusionは間違っただ関連付けを学習してしまうかもしれません。

こうして、画像と説明文のセットを大量に用意します。数百万、時には数千万という膨大な数のデータセットができあがります。

このデータセットを使って、Stable Diffusionは学習を行います。

膨大な量の画像と説明文を見ることで、言葉と画像の特徴の関係性を徐々に掴んでいくのです。

学習には非常に大きな計算資源と時間がかかります。数週間から数ヶ月もの間、強力なコンピュータで計算を続ける必要があります。

こうして時間と手間をかけて準備された大量の画像と説明文のデータセットが、Stable Diffusionの学習を支えているのです。

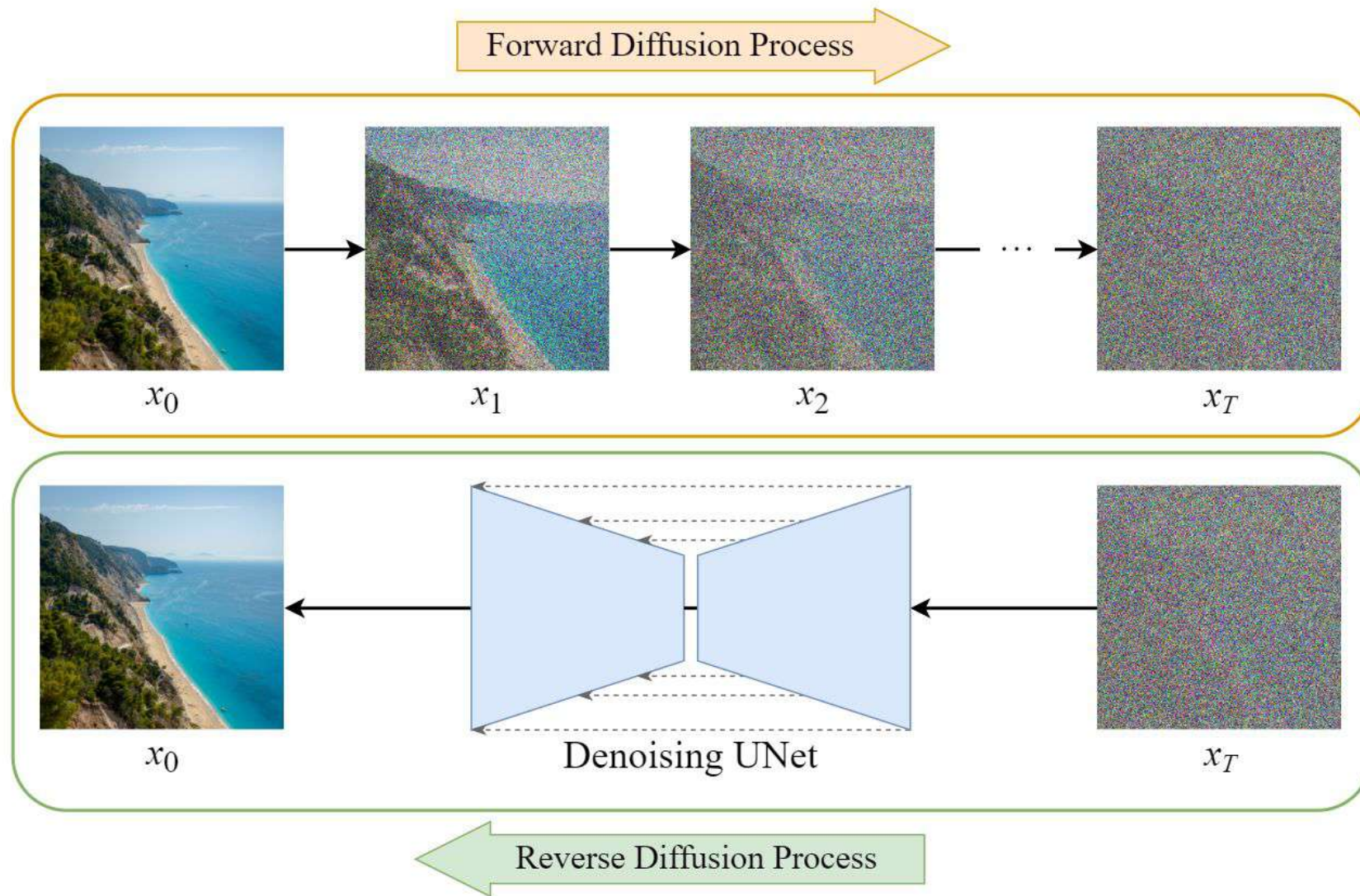
私たち人間とコンピュータが協力して、言葉から画像を生成する技術を作り上げているんですね。



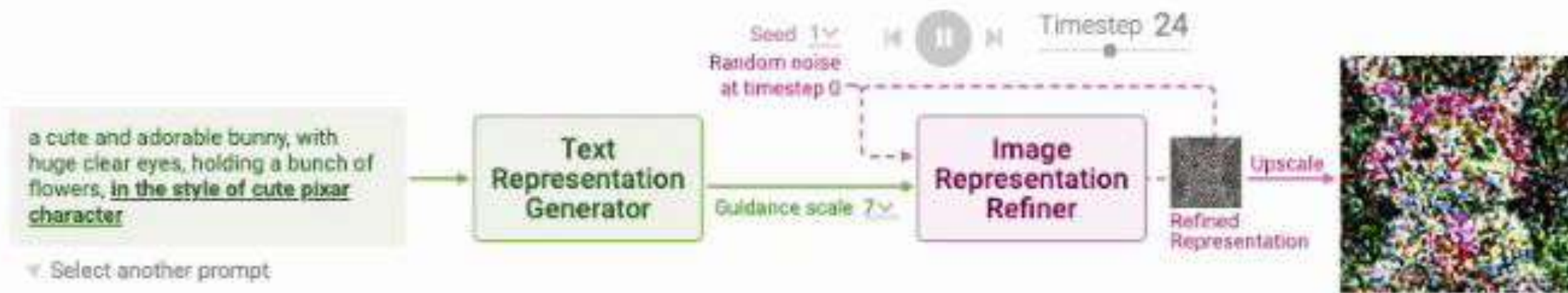
○

文字文字しいので
なんとなくでいいので
視覚的理解を促します

見た目で中身をふんわり理解しよう



中身を可視化



Diffusion Explainer

<https://poloclub.github.io/diffusion-explainer/>

こむずかしい余談(詳しく知りたい人向け) ネガティブプロンプト

ネガティブプロンプトはプロンプトを打ち消すものではなく、ネガティブプロンプトを入力したU-Netの出力から、プロンプトを入力したいU-Netの出力方向に先見を利かせるための制約を加えるイメージ。したがって、プロンプトをネガティブプロンプトを同一にしても何も印象が変わることはなく、プロンプトによる推論点からプロンプトによる推論点へ向かう(同じ出力にたどり着くだけ)です。つまり、単にネガティブプロンプトを与えてもGスケールを1にした場合と同様の結果となる(当然文脈では単調な印象で指示調整の最適な $w=0.2$ も同調)。

プロンプトエンジニアリングについて

プロンプトエンジニアリングがむずかしい方へ

○

あなたは高性能な Chatbot として、下記に従ってください。

ルール:

- ・プロンプトを「(要素:倍率), (要素:倍率), ...」の形式に変換する。
- ・全ての要素はシチュエーションに合わせて変更する。
- ・要素の個数を 15 個に増減する。
- ・要素は、英語や danbooru タグを使用する。
- ・要素に含まれる単語の数は、4 個以内に増減する。
- ・人物の要素を前に移動し、重要ではない要素を後ろに移動する。
- ・倍率を 0.5 ~ 1.3 に変更して、要素の強調を行う。

プロンプト:

(face close-up:0.6), (sitting elf girl:1.0), (wave hands and smile:1.0), (lookup forest:1.0), (boots and bow:1.0), (sanctuary ruins:1.0), (after rain and rainbow:1.0)

処理:

1. ルールに従い、無言でプロンプトを改良する。
2. 改良したプロンプトを表示して、追加の要望を尋ねる。
3. 2. を繰り返す。

プロンプトエンジニアリングについて



思いつかない方へ

。あなたは高性能な Chatbot として、下記に従ってください。

ルール:

- ・プロンプトを「(要素:倍率), (要素:倍率), ...」の形式に変換する。
- ・全ての要素はシチュエーションに合わせて変更する。
- ・要素の個数を 15 個に増減する。
- ・要素は、英語や danbooru タグを使用する。
- ・要素に含まれる単語の数は、4 個以内に増減する。
- ・人物の要素を前に移動し、重要ではない要素を後ろに移動する。
- ・倍率を 0.5 ~ 1.3 に変更して、要素の強調を行う。

プロンプト:

(face close-up:0.6), (sitting elf girl:1.0), (wave hands and smile:1.0), (lookup forest:1.0), (boots and ruins:1.0), (after rain and rainbow:1.0)

処理:

1. ルールに従い、無言でプロンプトを改良する。
2. 改良したプロンプトを表示して、追加の要望を尋ねる。
3. 2. を繰り返す。

