One-step Diffusion with Distribution Matching Distillation

- アブスト
  - Distribution Matching Distillation (DMD)と呼ばれるワンステップの画像生成器に拡散モデルを転移させる手法を提案。画像の品質をあまり落とすことなく実現できた
  - 分布レベルでワンステップの画像生成器と拡散モデルがマッチするようにKLダイバージェンスを最小化することで学習させた
    - このKLダイバージェンスの勾配は、ターゲット分布のスコア関数と画像生成器によって生成された合成分布のスコア関数の差として表現される
    - これらのスコア関数は、それぞれの分布に対して個別にトレーニングされた2つの拡散モデルとしてパラメータ化されている
- イントロ
  - サンプリング速度を向上させるために、従来手法では元のマルチステップの拡散サンプリングによって発見されたノイズから画像へのマッピングをシングルパスの生徒モデルに蒸留していた。しかしこのような高次元で複雑なマッピングを適応するのは困難であった
    - 生徒モデルの損失を一度計算するために、逆過程を全て実行しなければならなく、コストが高かった
    - 最近の手法では、元の拡散モデルの逆過程を全て実行せずに、生徒のサンプリング距離を段階的に増加させることでこれを緩和している。しかし、蒸留されたモデルの性能は、依然として元のマルチステップの拡散モデルに劣っている
      - 拡散モデルでは通常、ある初期状態から徐々にノイズを加えていく順過程によってデータを生成する。この順過程は、多くのステップに分

かれており、各ステップでノイズを除去し、元のデータに近づけるようにサンプリングが行われる。「sampling distance」は、この過程におけるステップ間で進行する距離や変化の度合いを意味します。距離が小さいほど、ステップ間での変化が少なく、より細かいサンプリングが行われる。逆に、距離が大きいと、サンプリングがより大きなステップで行われ、全体のサンプリングプロセスが早くなる可能性がありますが、その分、生成されるサンプルの精度や品質が低下するリスクもあります。

    - 対照的に、ノイズと拡散過程によって生成された画像の間の対応関係を強制するのではなく、単に生徒モデルによる生成画像が元の拡散モデルと区別がつかないようにすることを目指した。GMMNやGANsなど、他の分布マッチング型の生成モデルと同じ挙動を考えている。これらのモデルがリアルな画像を作成する上で成功しているのにもかかわらず、text-to-imageのデータにモデルを拡張することは依然として困難な課題となっている
    - 本研究では、大規模なtext-to-imageのデータで既に訓練された拡散モデルを使用することで、この問題を解決した。具体的には、事前に訓練された拡散モデルをファインチューニングし、データの分布だけでなく、蒸留された生成器が生成する偽の分布も学習させる。拡散モデルは拡散分布におけるスコア関数を近似することが知られているため、拡散モデルの出力を、画像をよりリアルにするための勾配方向として解釈することができる。また、拡散モデルが偽の画像で学習されている場合、より偽っぽくする方向としても解釈できる。生成器の勾配更新規則はこれらの差分として構成され、生成画像をよりリアルにし、偽っぽさを減少させるように調整できた
    - さらに、マルチステップの拡散サンプリングの結果を適度な数だけ事前に計算し、それに対するシンプルな回帰損失を用いることで、分布マッチング損失の存在下でも効果的な正則化を実現できることがわかった。また、回帰損失により、ワンステップ生成器が教師モデルと整合することが保証され、リアルタイム生成の可能性が示された。この手法は、VSD、GANs、およびpix2pixを参考にしており、拡散モデルを用いてリアルおよびフェイクの分布をモデル化し、マルチステップの拡散過程の出力に一致させるためにシンプルな回帰損失を使用することで、高忠実度のワンステップ生成モデルを訓練できることを保証した
- 関連研究

- Diffusion Acceleration
  - LuhmanらおよびDSNOは、逆過程の軌跡を事前に計算し、ピクセル空間での回帰損失を用いて生徒モデルを訓練するというシンプルなアプローチを提案した。しかし、各損失関数の計算ごとに全てのノイズ除去過程を計算するため、非常に高コストであった。この問題に対処するために、Progressive Distillation (PD)は、前のモデルのサンプリングステップ数を半分にした一連の生徒モデルを訓練した

- Distribution Matching Distillation
  - 2つの損失の合計を最小化することで高速な生成器を訓練させた。一つは分布をマッチさせる目的で、その勾配更新は2つのスコア関数の差として表現できる。もう一つは回帰損失で、事前に作成されたノイズと画像のペアのデータを使用して、生成器がベースモデルの出力に一致するように学習させた。実際の分布と偽の分布のスコア関数をそれぞれモデル化するために、2つの拡散モデルを使用し、それらは様々な強度のガウスノイズで摂動させている

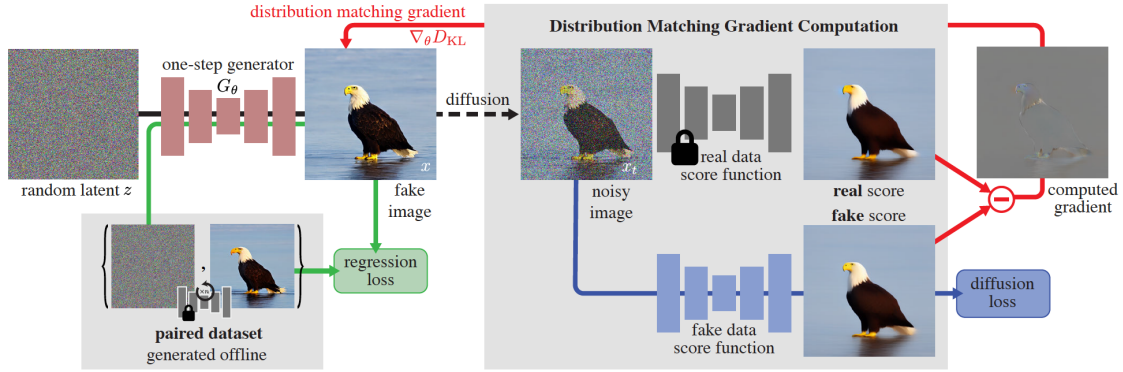

Figure 2. **Method overview.** We train one-step generator G_θ to map random noise z into a realistic image. To match the multi-step sampling outputs of the diffusion model, we pre-compute a collection of noise–image pairs, and occasionally load the noise from the collection and enforce LPIPS [85] regression loss between our one-step generator and the diffusion output. Furthermore, we provide **distribution matching gradient** $\nabla_\theta D_{KL}$ to the fake image to enhance realism. We inject a random amount of noise to the fake image and pass it to two diffusion models, one pretrained on the real data and the other continually trained on the fake images with a diffusion loss, to obtain its denoised versions. The denoising scores (visualized as mean prediction in the plot) indicate directions to make the images more realistic or fake. The difference between the two represents the direction toward more realism and less fakeness and is backpropagated to the one-step generator.

- Pretrained base model and One-step generator
  - 学習済みのEDMとStable Diffusionを使用した
  - One-step generator
    - ワンステップ生成器G_θは、時間条件付けを持たないベース拡散モデルのアーキテクチャを採用している。また、訓練前に、そのパラメータθ

をベースモデルで初期化している。つまり、すべてのz に対して、$G_\theta(z) = \mu_{base}(z, T-1)$と初期化している（コメント：これってT-1のノイズの状態からスタートしていて、次の１ステップを学習しているだけじゃね？蒸留するモデルも別の構造を取っているわけでなく、教師モデルと同じっぽい）

- Distribution Matching Loss
  - 真と偽の画像分布のKLダイバージェンスを最小化する$D_{KL}(p_{fake}||p_{real})$

$$
\begin{aligned}
D_{KL}\left(p_{\text{fake}} \parallel p_{\text{real}}\right) &= \mathop{\mathbb{E}}_{x \sim p_{\text{fake}}} \left(\log \left(\frac{p_{\text{fake}}(x)}{p_{\text{real}}(x)} \right) \right) \\
&= \mathop{\mathbb{E}}_{\substack{z \sim \mathcal{N}(0;\mathbf{I}) \\ x = G_\theta(z)}} -\big(\log\ p_{\text{real}}(x) - \log\ p_{\text{fake}}(x) \big).
\end{aligned}
$$

  - 勾配を取ると以下のようになる

$$
\nabla_\theta D_{KL} = \mathop{\mathbb{E}}_{\substack{z \sim \mathcal{N}(0;\mathbf{I}) \\ x = G_\theta(z)}} \Big[-\big(s_{\text{real}}(x) - s_{\text{fake}}(x)\big) \nabla_\theta G_\theta(z) \Big], \tag{2}
$$

$$
\text{where } s_{\text{real}}(x) = \nabla_x \log p_{\text{real}}(x),\ s_{\text{fake}}(x) = \nabla_x \log p_{\text{fake}}(x)
$$

    s_{real}, s_{fake}はそれぞれの分布のスコアである。s_{real}はxをp_{real}の方向に移動させ、s_{fake}はxをp_{fake}の方向に移動させる

  - データ分布に様々な標準偏差を持つランダムなガウスノイズを加えることで、空間全体をカバーするぼやけた分布を作成し、それらが重なるようにする。これにより、式（2）の勾配が適切に定義される。Score-SDE は、訓練された拡散モデルが拡散された分布のスコア関数を予測できることを示している
  - Real Score

    $s_{real}(x_t, t) = -\frac{x_t - \alpha_t \mu_{base}(x_t, t)}{\sigma_t^2}$
  - Dynamically-learned fake score

    $s_{fake}(x_t, t) = -\frac{x_t - \alpha_t \mu_{fake}^{\phi}(x_t, t)}{\sigma_t^2}$

偽の拡散モデルを、事前に訓練された拡散モデルから初期化し、以下の標準的なノイズ除去の損失関数を最小化することで、訓練中にパラメータを更新する

$$\mathcal{L}_{denoise}^{\phi} = ||\mu_{fake}^{\phi}(x_t, t) - x_0||_2^2$$

- Regression loss and final objective
  - 先ほどの分布をマッチさせるのは、t>>0の場合、すなわち生成されたサンプルに大量のノイズが加えられている時に定義される。しかし、ノイズが少ない場合、$s_{real}(x_t, t)$はしばしば信頼できなくなり、特に$p_{real}(x_t, t)$がゼロに近づくとその傾向が顕著になる。さらに、スコア$\nabla_x \log p$は確率密度関数 p のスケーリングに対して不変であるため、最適化がモード崩壊やモードドロップに陥りやすくなります。つまり、偽の分布が特定のモードのサブセットに対して全体的な密度を割り当てやすくなるという現象が起こり得る。これを避けるために、追加の回帰損失を用いてすべてのモードが保持されるようにする。この損失は、同じ入力ノイズを与えられた場合の生成器とベース拡散モデルの出力間の点ごとの距離を測定する。具体的には、ランダムなガウスノイズ画像 z と、事前に訓練された拡散モデルμ_{base}を用いて決定論的なODEソルバーで得られる出力 y のペアデータセット$D = \{z, y\}$を構築する。この回帰損失にはLPIPSを使用した

- Limitations
  - 提案したワンステップのモデルと、100回や1000回サンプリングするような拡散モデルでは品質に僅かな差がある。また、モデルの性能は、教師モデルの能力によって本質的に制限されている。Stable Diffusion v1.5で観察された制限と同様に、1ステップ生成器は、判読可能なテキストや小さな顔や人物の詳細な描写に苦労している。モデルはトレーニング中に固定されたガイダンススケールを使用しているが、Guided-Distillationで使用されている条件付きガイダンスのように、可変なガイダンススケールを導入することで、推論時により柔軟性の高い画像の生成が見込める