# Cold diffusion

## Inverting Arbitrary Image Transforms Without Noise

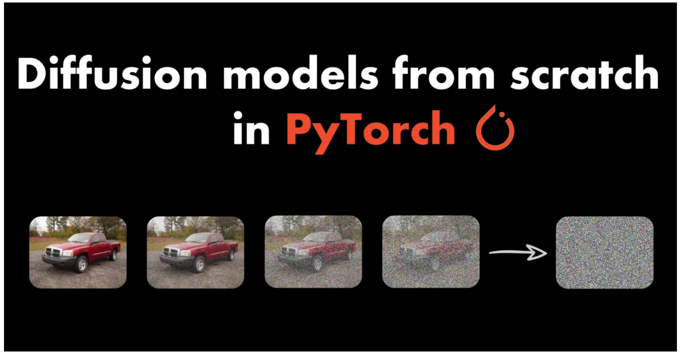
ArXiv preprint: 2208.09392. Currently under open-review for ICLR23 slHNW9yRie0.
Article authors: A. Bansal, E. Borgnia, HM. Chu, JS. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, T. Goldstein
Presented by Pavol Harar at Deep Learning Seminar @dsUniVie, 16. November 2022

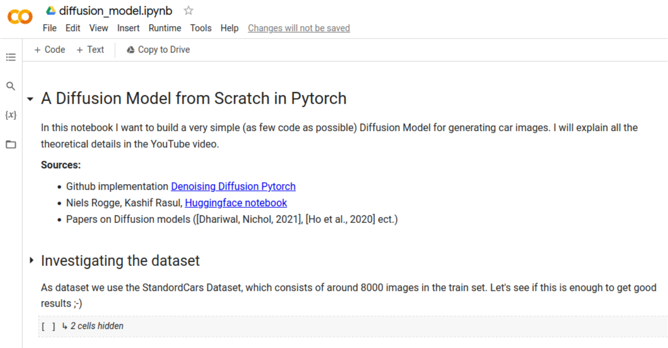
# Diffusion models - recap

- Standard diffusion models are built around two components. First, there is an image degradation /forward/ operator ($D$) that contaminates images with Gaussian noise.
- Second, a restoration /backward/ operator ($R$) is trained to perform denoising.

Given an image $x_0 \in \mathbb{R}^N$, consider the *degradation* of $x_0$ by operator $D$ with severity $t$, denoted $x_t = D(x_0, t)$. The output distribution $D(x_0, t)$ of the degradation should vary continuously in $t$, and the operator should satisfy $D(x_0, 0) = x_0$.

# Simplified implementation




- For a simple and quick implementation example see: https://youtu.be/a4Yfz2FxXiY
- And its accompanying colab notebook: 1sjy9odlSSy0RBVgMTgP7s99NXsqglsUL

# Implementation outline - training

- `for e in range(n_epochs):`
  - `t = random_timestep_up_to_T()`
  - `t_emb = sinusoidal_position_embedding(t)`
  - `noise, noisy_image = D(noiseless_image, t_emb)`
  - `noise_pred = R(noisy_image, t_emb)`
  - `loss = L1(noise, noise_pred)`
  - `loss.backward()`

  `return trained R  # in this case a U-Net`

# Implementation outline - sampling

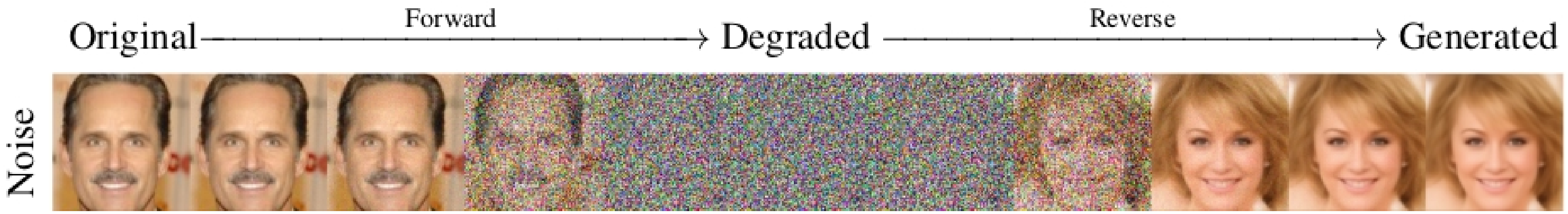
**Algorithm 1** Naive Sampling

**Input:** A degraded sample $x_t$
**for** $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = D(\hat{x}_0, s-1)$
**end for**
**Return:** $x_0$

- Standard sampling works well for noise-based diffusion, however it yields poor results in the case of cold diffusions with differentiable degradations (such as deblurring)

# Required Gaussian noise

- Diffusion has been understood as a random walk around the image density function using Langevin dynamics, which requires Gaussian noise in each step.
- The walk begins in a high temperature (heavy noise) state, and slowly anneals into a "cold" state with little if any noise.



# Cold diffusion – article highlights

- Sheds light on the role of noise in diffusion models.
- Shows that noise is not a necessity in diffusion models.
- Proposes a sampling algorithm called Transformation Agnostic Cold Sampling (TACoS) for generalized diffusions.
- Provides theoretical and empirical results in applications to various inverse problems (conditional generation) and generation of images (unconditional generation).

# Cold diffusion – cited recent works

- Recently, diffusion models have been applied to inverse problems [Song et al., 2021b] such as deblurring, denoising, super-resolution, and compressive sensing [Whang et al., 2021, Kawar et al., 2021b, Saharia et al., 2021, Kadkhodaie and Simoncelli, 2021].
- Although not their focus, previous works experimented with deterministic image generation [Song et al., 2021a, Dhariwal and Nichol, 2021] and in selected inverse problems [Kawar et al., 2022].
- Reviewers criticize that some similar papers are not cited.

# Cold diffusion – proposed sampling

**Algorithm 2** Transformation Agnostic Cold Sampling

**Input:** A degraded sample $x_t$
**for** $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$
**end for**

- Propose TACoS for sampling, and claim it is superior for inverting smooth, cold degradations.
- “Specifically, for a class of linear degradation operations, it produces exact reconstruction (i.e. $x_s = D(x_0, s)$) even when the restoration operator *R* fails to perfectly invert *D*.”

# Cold diffusion – diff in sampling

- *Algorithm 1:*

  $x_{s-1} = D(R(x_s, s), s-1)$

- *TACoS:*

  $x_{s-1} = x_s - D(R(x_s, s), s) + D(R(x_s, s), s-1)$

# Cold diffusion – comparing sampling

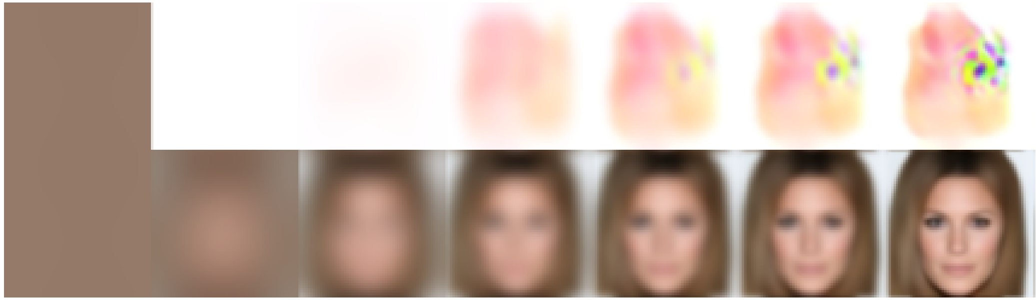


Figure 2: Comparison of sampling methods for cold diffusion on the CelebA dataset. **Top:** Algorithm 1 produces compounding artifacts and fails to generate a new image. **Bottom:** TACoS succeeds in sampling a high quality image without noise.

# Cold diffusion – experiments
**(conditional generation)**

- deblurring
- inpainting
- super-resolution
- snowification

# Cold diffusion – deblurring
## (conditional generation)

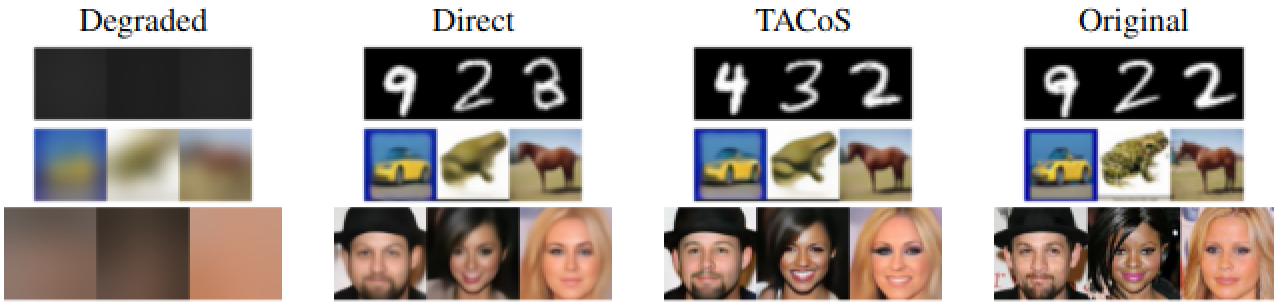



Figure 3: Deblurring models trained on the MNIST, CIFAR-10, and CelebA datasets. **Left to right:** degraded inputs $D(x_0, T)$ , direct reconstruction $R(D(x_0, T))$, sampled reconstruction with TACoS described in Algorithm 2, and original image.

Table 1: Quantitative metrics for quality of image reconstruction using deblurring models.

| | **Degraded** | | | **Sampled** | | | **Direct** | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | FID | SSIM | RMSE | FID | SSIM | RMSE | FID | SSIM | RMSE |
| MNIST | 438.59 | 0.287 | 0.287 | **4.69** | 0.718 | 0.154 | 5.10 | **0.757** | 0.142 |
| CIFAR-10 | 298.60 | 0.315 | 0.136 | **80.08** | 0.773 | 0.075 | 83.69 | **0.775** | 0.071 |
| CelebA | 382.81 | 0.254 | 0.193 | **26.14** | 0.568 | 0.093 | 36.37 | **0.607** | 0.083 |

# Cold diffusion – inpainting
## (conditional generation)

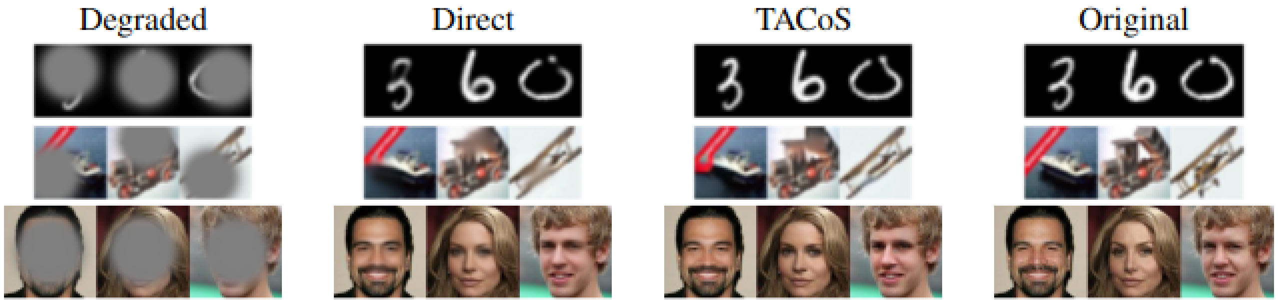



Figure 4: Inpainting models trained on the MNIST, CIFAR-10, and CelebA datasets. **Left to right:** Degraded inputs $D(x_0, T)$ , direct reconstruction $R(D(x_0, T))$, sampled reconstruction with TACoS described in Algorithm 2, and original image.

Table 2: Quantitative metrics for quality of image reconstruction using inpainting models.

| Dataset | Degraded FID | Degraded SSIM | Degraded RMSE | Sampled FID | Sampled SSIM | Sampled RMSE | Direct FID | Direct SSIM | Direct RMSE |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | 108.48 | 0.490 | 0.262 | **1.61** | 0.941 | 0.068 | 2.24 | **0.948** | 0.060 |
| CIFAR-10 | 40.83 | 0.615 | 0.143 | **8.92** | 0.859 | 0.068 | 9.97 | **0.869** | 0.063 |
| CelebA | 127.85 | 0.663 | 0.155 | **5.73** | 0.917 | 0.043 | 7.74 | **0.922** | 0.039 |

# Cold diffusion – inpainting

**(conditional generation)**

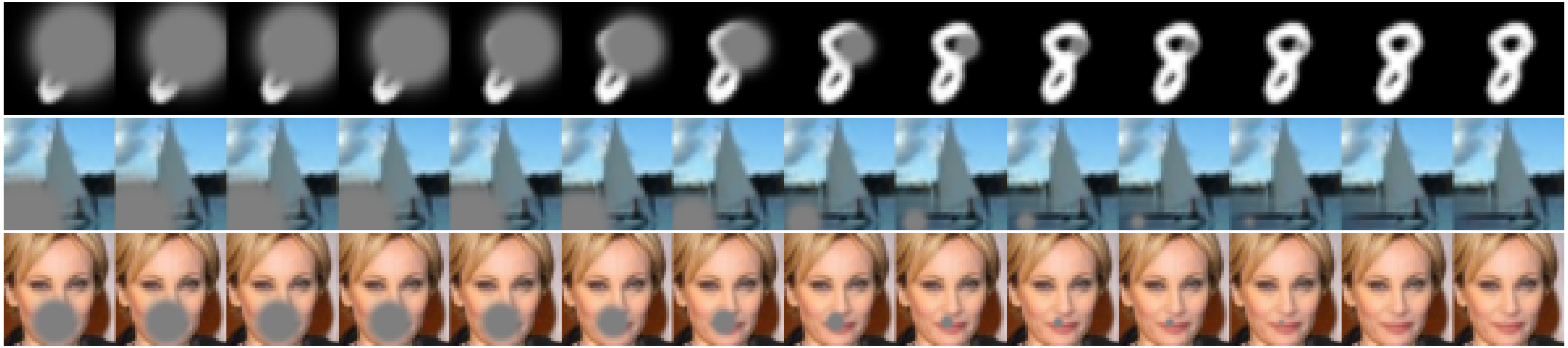


Figure 10: Progressive inpainting of selected masked MNIST, CIFAR-10, and CelebA images.

# Cold diffusion – super-resolution

**(conditional generation)**

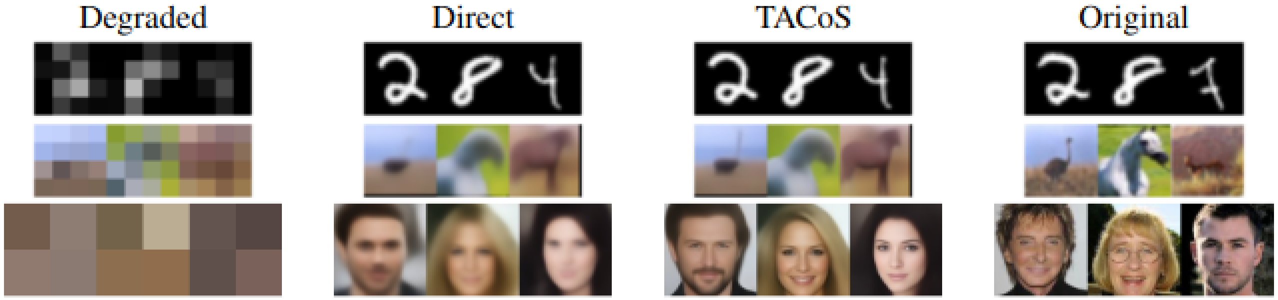



Figure 5: Superresolution models trained on the MNIST, CIFAR-10, and CelebA datasets. **Left to right:** degraded inputs $D(x_0, T)$ , direct reconstruction $R(D(x_0, T))$, sampled reconstruction with TACoS described in Algorithm 2, and original image.

Table 3: Quantitative metrics for quality of image reconstruction using super-resolution models.

| Dataset | Degraded FID | Degraded SSIM | Degraded RMSE | Sampled FID | Sampled SSIM | Sampled RMSE | Direct FID | Direct SSIM | Direct RMSE |
|---|---|---|---|---|---|---|---|---|---|
| MNIST | 368.56 | 0.178 | 0.231 | 4.33 | 0.820 | 0.115 | **4.05** | **0.823** | 0.114 |
| CIFAR-10 | 358.99 | 0.279 | 0.146 | **152.76** | 0.411 | 0.155 | 169.94 | **0.420** | 0.152 |
| CelebA | 349.85 | 0.335 | 0.225 | **96.92** | 0.381 | 0.201 | 112.84 | **0.400** | 0.196 |

# Cold diffusion – super-resolution

## (conditional generation)

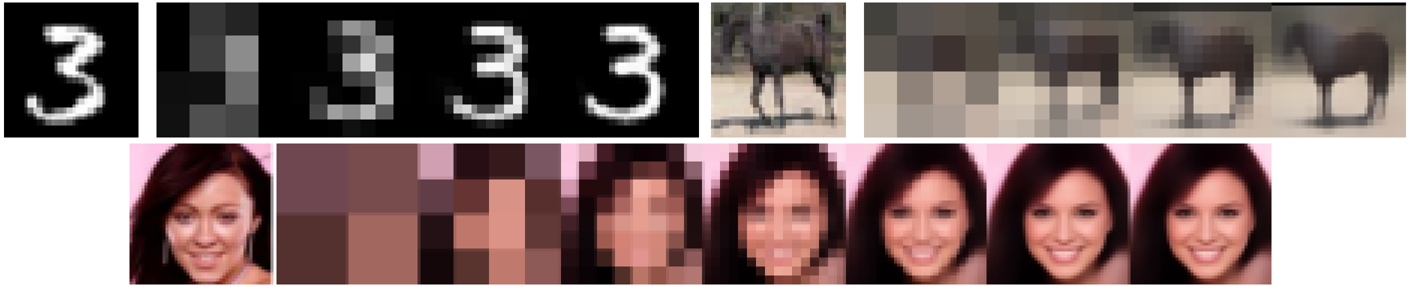


Figure 12: Progressive upsampling of selected downsampled MNIST, CIFAR-10, and CelebA images. The original image is at the left for each of these progressive upsamplings.

# Cold diffusion – snowification
## (conditional generation)


Figure 6: *Desnowification* models trained on the CIFAR-10, and CelebA datasets. **Left to right:** degraded inputs $D(x_0, T)$ , direct reconstruction $R(D(x_0, T))$, sampled reconstruction with TACoS described in Algorithm 2, and original image.

Table 4: Quantitative metrics for quality of image reconstruction using *desnowification* models.

| | **Degraded Image** | | | **Reconstruction** | | |
|---|---|---|---|---|---|---|
| Dataset | FID | SSIM | RMSE | FID | SSIM | RMSE |
| CIFAR-10 | 125.63 | 0.419 | 0.327 | 31.10 | 0.074 | 0.838 |
| CelebA | 398.31 | 0.338 | 0.283 | 27.09 | 0.033 | 0.907 |

# Cold diffusion – experiments

**(unconditional or cold generation)**

- Using deterministic noise degradation
- Using blur
- Using other transformations

# Cold diffusion – deterministic noise

**(unconditional or cold generation)**

Table 5: FID scores for CelebA and AFHQ datasets using hot (using noise) and cold diffusion (using blur transformation). This table shows that This table also shows that breaking the symmetry withing pixels of the same channel further improves the FID scores.

| | **Hot Diffusion** | | **Cold Diffusion** | |
|---|---|---|---|---|
| Dataset | Fixed Noise | Estimated Noise | Perfect symmetry | Broken symmetry |
| CelebA | 59.91 | 23.11 | 97.00 | 49.45 |
| AFHQ | 25.62 | 20.59 | 93.05 | 54.68 |

# Cold diffusion – blur

## (unconditional or cold generation)

Figure 7: Examples of generated samples from 128 × 128 CelebA and AFHQ datasets using cold diffusion with blur transformation

# Cold diffusion – other transforms
## (unconditional or cold generation)

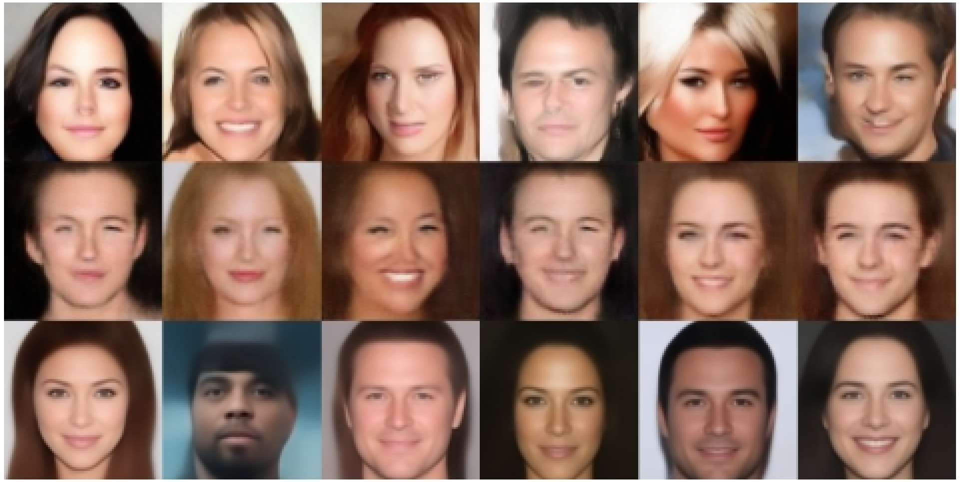


Figure 8: Preliminary demonstration of the generative abilities of other cold diffusins on the $128 \times 128$ CelebA dataset. The top row is with *animorphosis* models, the middle row is with inpainting models, and the bottom row exhibits super-resolution models.

# Cold diffusion – conclusion

- Random noise can be removed entirely from the diffusion model framework
- Random noise can be replaced with arbitrary transforms
- Proposed generalization allowes to restore images afflicted by deterministic degradations
- This framework paves the way for a more diverse landscape ofdiffusion models