Методы ускорения диффузионных моделей
Обзор

Быков И.

Sep 2024

Рассмотренные модели

Cachers

- DeepCache – Кэширование высокоуровневых признаков
- T-Gate – Кэширование первых cross-attention
- FreeU – Нормализация параметров в skip-connection и backbone U-net

Solvers

- DPMSolver – Численное приближение (Тейлор) одной из форм PF-ODE
- DEIS – Численное приближение (Сплайн) одной из форм PF-ODE
- UniPC – Корректор любого Schedulera, солвер произвольного порядка
- EDM – Замена переменных PF-ODE, "контролируемый шум"
- PNDM – Интерпретация ϵ_θ как градиента
- TCD* – Дистилляция модели, "контролируемый шум"

Schedulers

- AYS – Подбор timesteps минимизируя D_{KL} непрерывного и дискретного SDE
- EDM (Karras sigmas) – Timesteps основаны на $1/\sigma_i$

Unet

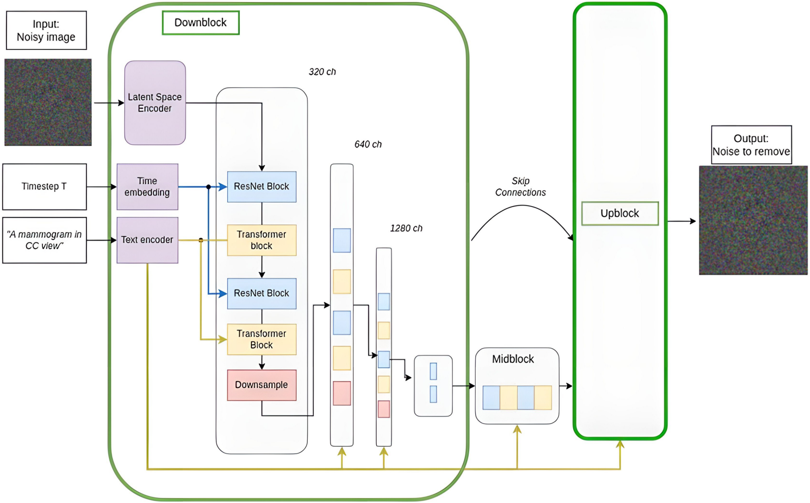


Рис.. Diffusion Unet

DeepCache

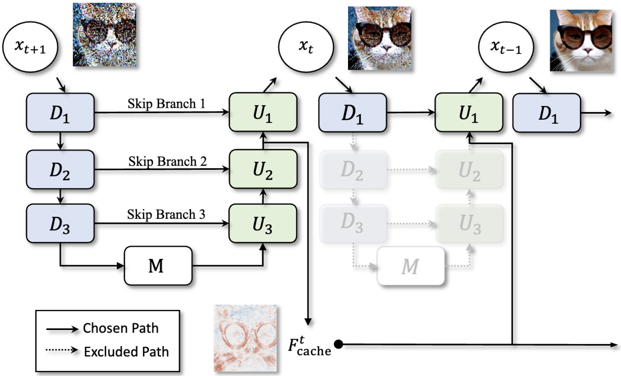


Рис.. Принцип работы DeepCache

- Оптимальный вариант (равномерного) кэширования: делать полный инференс на каждом 3 шаге

T-Gate

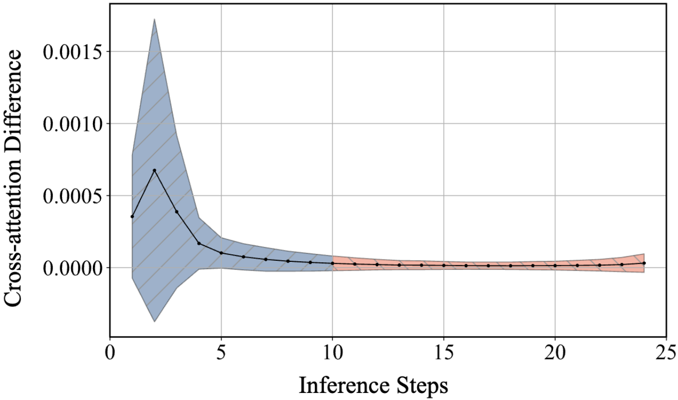


Рис.. Среднее (по изображениям и слоям cross-attention) разницы двух последовательных timestamps

- Кэширование всех cross-attention на $m \approx \text{NFE}//2.5$ timestamp и использование на всех последующих timestamps

FreeU

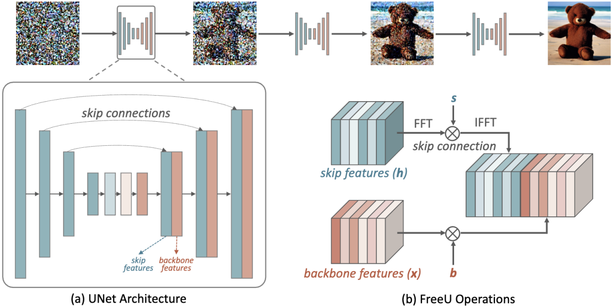
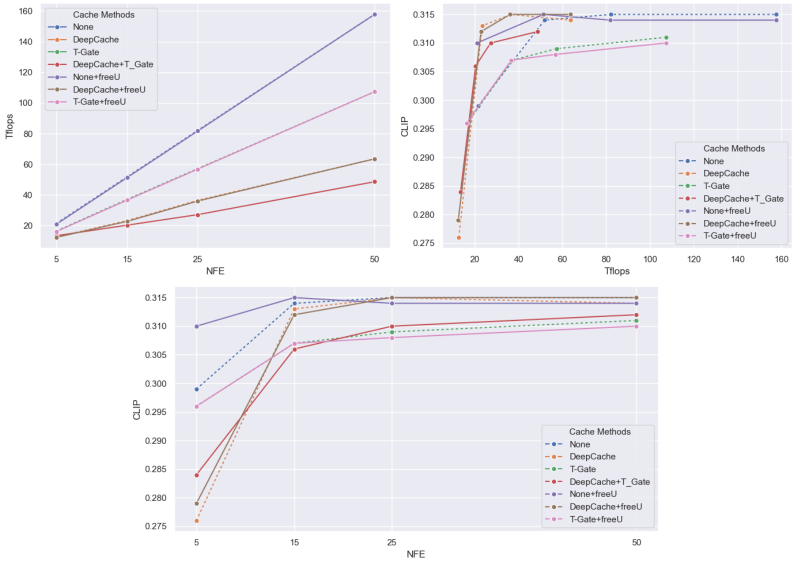


Рис.. Принцип работы FreeU (только для первых двух блоков декодера)

- "Нормализация" параметров $b > 1$: $\quad x \mapsto x \cdot \left[1 + (b-1)\frac{\bar{x} - \min(\bar{x})}{\max(\bar{x}) - \min(\bar{x})}\right]$
- Ослабление высокочастотных признаков в skip-connection с помощью $s < 1$

Cachers


Cache Methods
None
DeepCache
T-Gate
DeepCache+T_Gate
None+freeU
DeepCache+freeU
T-Gate+freeU
Tflops
NFE
CLIP
Tflops
CLIP
NFE

PF ODE

- $x_t \sim N(\alpha_t x_0, \sigma_t^2 I),\ dx_t = x_t f_t dt + g_t dw$

$$\left(\Rightarrow f_t = \dot{\alpha}_t,\ g_t^2 = \frac{\partial \sigma_t^2}{\partial t} - 2\dot{\alpha}_t \sigma_t^2, \text{ см. [1], Yang Song et al}\right)$$

- BWD SDE:

$$dx_t = [x_t f_t + g_t^2 \nabla_x \log p_t(x_t)]dt + g_t d\hat{w}_t, \quad \hat{w}_t - \text{reverse } w_t(t: \ T \to 0) \tag{1}$$

- BWD PF ODE:

$$dx_t = [x_t f_t - \frac{1}{2} g_t^2 \nabla_x \log p_t(x_t)]dt = [x_t f_t + g_t^2 \frac{\epsilon_\theta(x,t)}{2\sigma_t}]dt \tag{2}$$

- решение (2) принимает вид

$$x_t = \frac{\alpha_t}{\alpha_s} x_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \hat{\epsilon}_\theta\left(\hat{x}_\lambda, \lambda\right) d\lambda \tag{3}$$

или

$$x_t = \frac{\sigma_t}{\sigma_s} x_s + \sigma_t \int_{\lambda_s}^{\lambda_t} e^{\lambda} \hat{x}_\theta\left(\hat{x}_\lambda, \lambda\right) d\lambda \tag{4}$$

где

$$\lambda_t = \log(\alpha_t/\sigma_t),\ t_\lambda(\cdot) = \lambda_t^{-1}(\cdot),\ \hat{x}_\lambda = x_{t_\lambda(\lambda)},\ \hat{\epsilon}_\theta\left(\hat{x}_\lambda, \lambda\right) = \epsilon_\theta\left(x_{t_\lambda(\lambda)}, t_\lambda(\lambda)\right)$$

DPMSolver(++)

- в статьях [2] и [3] и были предложены аналитические решения (3), (4)
- метод заключается в разложении ϵ_θ в ряд тейлора и приближением производных (например, методом Адамса)
- в [3] так же предложено аналитическое решение (1) с таким же разложением в ряд, метод – SDE-DPMSolver++

DEIS

- в статье ([5], Qinsheng Zhang et al) решают ODE (2) в общем виде (возможно недиагональная матрица ковариации Σ_t процесса x_t):

$$dx_t = [F_t x_t + G_t G_t^\intercal L_t^{-\intercal} \epsilon_\theta(x_t, t)]dt, \; L_t L_t^\intercal = \Sigma_t$$

- вместо разложения ϵ_θ в ряд Тейлора (как в [2], [3]), авторы приближают ϵ_θ r-полиномом по уже подсчитанным $(t, \; \epsilon_\theta(x_t, t))$:

$$P_r(t) = \sum_{j=0}^{r} \left[\prod_{k \neq j} \frac{t - t_{i+k}}{t_{i+j} - t_{i+k}} \right] \epsilon_\theta \left(x_{t_{i+j}}, t_{i+j} \right)$$

- также можно привести (2) к виду

$$dy_t = \frac{1}{2} \Psi(t, 0) G_t G_t^\intercal L_t^{-\intercal} \epsilon\theta(\Psi(0, t) y_t, t)$$

и дискретизировать с помощью multistep-методов, получая *RK-DEIS*, *AB-DEIS* (для них метрики получаются хуже)

UniPC

- UniC-p: корректор для любого солвера порядка $p \mapsto$ солвер порядка $p+1$
- Авторы [6] видоизменили сэмл шага в (3): для каждого нового шага $\tilde{x}_{t_i}$ дополнительно считается $\tilde{x}^c_{t_i}$ (корректировка):

$$\tilde{x}^c_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\tilde{x}^c_{t_{i-1}} - \sigma_{t_i}\left(e^{h_i}-1\right)\epsilon_\theta\left(\tilde{x}_{t_{i-1}}, t_{i-1}\right) - \sigma_{t_i}B\left(h_i\right)\sum_{m=1}^{p}\frac{a_m}{r_m}D_m,$$

$$h_i = \lambda_{t_i} - \lambda_{t_{i-1}},\ D_m = \epsilon(\tilde{x}_{s_m}, s_m) - \epsilon(\tilde{x}_{t_{i-1}}, t_{i-1}) \tag{5}$$

- a_m подбираются таким образом, чтобы сократить более малые порядки
- UniP-p: если в (5) суммировать до $p-1$, то $\tilde{x}^c_{t_i}$ больше не зависит от $\tilde{x}_{t_i}$ и можно доказать ([6], Corollary 3.2) что получившийся солвер (теперь $\tilde{x}^c_{t_i}$ это просто предсказанное $\tilde{x}_{t_i}$) порядка p.
- комбинация UniP-p и UniC-p дает UniPC-p порядка $p+1$

EDM

- другая параметризация $x_t \sim N(s_t x_0, s_t^2 \sigma_t^2 I)$ вместе с заменой переменных $p(x; \sigma) := p_0(x) * N(x;\ 0,\ \sigma^2 I)$ приводит (2) к выражению вида

$$x_t = \left[\frac{\dot{s}_t}{s_t} x_t - s_t^2 \dot{\sigma}_t \sigma_t \nabla_{x_t} \log p \left(\frac{x_t}{s_t}; \sigma_t \right) \right] dt \tag{6}$$

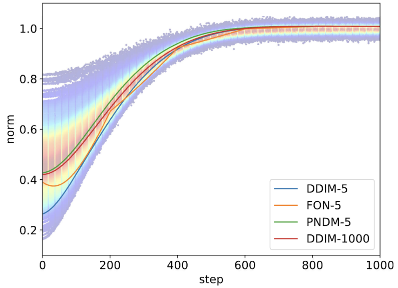
- дискретизируя и используя Heun's 2 order method авторы получают детерминированный алгоритм
- оказывается, аналогичное выражение к (6) можно представить в виде SDE для FWD(+) и BWD(-) уравнений (при $s_t = 1$):

$$\mathrm{d}\boldsymbol{x}_\pm = \underbrace{-\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x}; \sigma(t)\big)\,\mathrm{d}t}_{\text{probability flow ODE (Eq. 1)}} \pm \underbrace{\underbrace{\beta(t)\sigma(t)^2\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x}; \sigma(t)\big)\,\mathrm{d}t}_{\text{deterministic noise decay}} + \underbrace{\sqrt{2\beta(t)}\sigma(t)\,\mathrm{d}\omega_t}_{\text{noise injection}}}_{\text{Langevin diffusion SDE}}, \tag{7}$$

- авторы [4] предлагают брать $\sigma_t = t$, контролировать наличие noise injection параметрами $t_i \in [S_{\min}, S_{\max}]$, S_{churn} (общий уровень случайности) и S_{noise} (множитель к $\sigma_t \mapsto S_{\text{noise}}\sigma_t$)

PNDM

- процесс зашумления порождает нетривиальное многообразие ([7] Zhou Zhao et al):



- авторы показывают, что шаг сэмплирования DDIM "точный" при "точно" предсказанной ошибке ϵ_θ (т.е. лежит на многообразии путей процесса зашумления)
- замечая визуальную схожесть шага $x_{t_i} = Ax_{t_{i-1}} - B\varepsilon_\theta(x_{t_{i-1}}, t_{i-1})$ с шагом градиентного спуска, предлагается интерпретировать ϵ_θ как градиент и использовать multistep method для улучшения сходимости (метод 2 порядка)

TCD

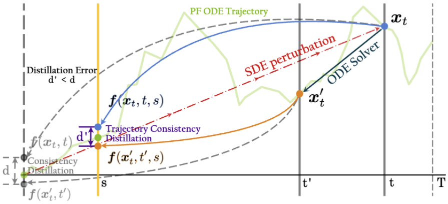


Рис.. TCD train

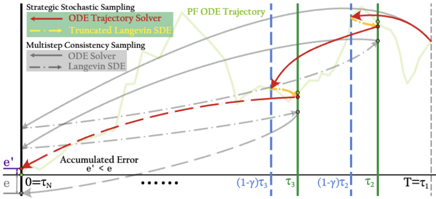


Рис.. TCD sample

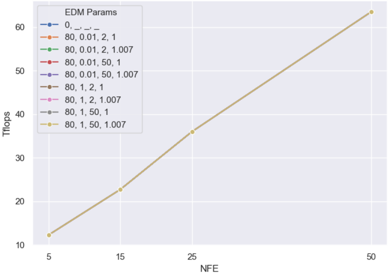
- SSS sampler:

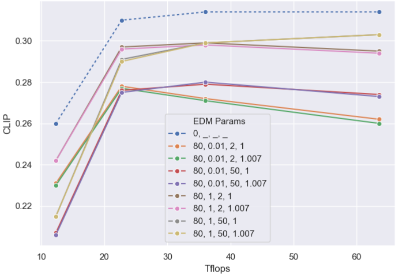
$$x_s = \frac{\alpha_s}{\alpha_{s'}} \underbrace{\left(\alpha_{s'} \frac{x_t - \sigma_t \epsilon_\theta\left(x_t\right)}{\alpha_t} + \sigma_{s'} \epsilon_\theta\left(x_t\right)\right)}_{\text{predicted } x_{s'} = f(x_t, t, s')} + \underbrace{\sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}} z}_{\text{controllable noise}} \quad , \gamma = 1 - s'/s$$

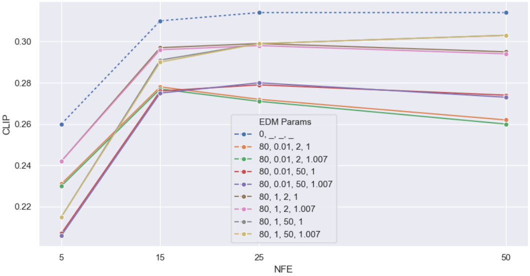
- Train:

$$L(\theta, \theta^-, \phi) = \mathbb{E}\left\| f_\theta(x_{t_{n+k}}, t_{n+k}, t_m) - f_{\theta^-}(\hat{x}_{t_n}^{\phi,k}, t_n, t_m) \right\|, \quad n \sim U(1, N-1),\ m \sim U(1, n)$$

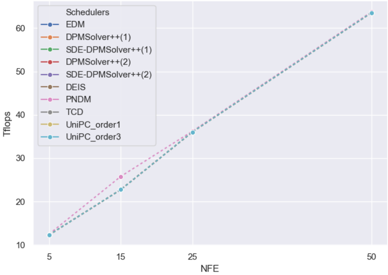
EDM

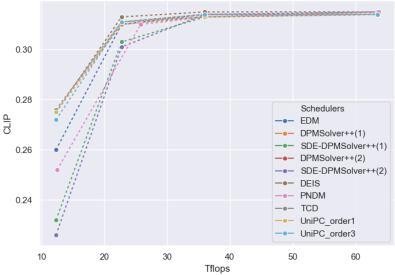

EDM Params
0, _, _, _
80, 0.01, 2, 1
80, 0.01, 2, 1.007
80, 0.01, 50, 1
80, 0.01, 50, 1.007
80, 1, 2, 1
80, 1, 2, 1.007
80, 1, 50, 1
80, 1, 50, 1.007
Tflops
NFE

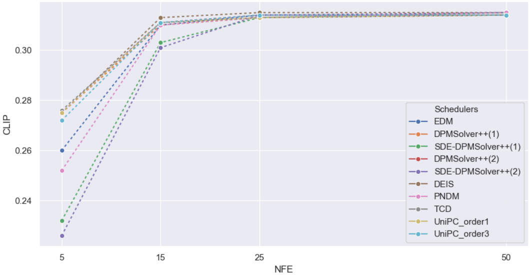

EDM Params
0, _, _, _
80, 0.01, 2, 1
80, 0.01, 2, 1.007
80, 0.01, 50, 1
80, 0.01, 50, 1.007
80, 1, 2, 1
80, 1, 2, 1.007
80, 1, 50, 1
80, 1, 50, 1.007
CLIP
Tflops


EDM Params
0, _, _, _
80, 0.01, 2, 1
80, 0.01, 2, 1.007
80, 0.01, 50, 1
80, 0.01, 50, 1.007
80, 1, 2, 1
80, 1, 2, 1.007
80, 1, 50, 1
80, 1, 50, 1.007
CLIP
NFE

EDM, DPM, DEIS, PNDM, TCD, UniPC







AYS

- Чтобы уменьшить ошибку дискретизации (например, (2)) можно напрямую минимизировать KLUB

$$D_{\mathrm{KL}}(P_1 \| P_2) \leq \mathrm{KLUB}(0, T) \coloneqq$$
$$\frac{1}{2}\mathbb{E}_{P_1^{paths}}\left[\int_0^T \frac{||\mathbf{f}_1(\mathbf{x}_{0\to t}, t) - \mathbf{f}_2(\mathbf{x}_{0\to t}, t)||^2}{g(t)^2} dt\right]$$

- в нашем случае P_1^{paths} соответствует непрерывному процессу (авторы [9] рассматривают (7) при общем виде s_t), а P_2^{paths} дискретному
- оптимальные t_i ищутся при фиксированных $t_{\min}$, $t_{\max}$ итеративно

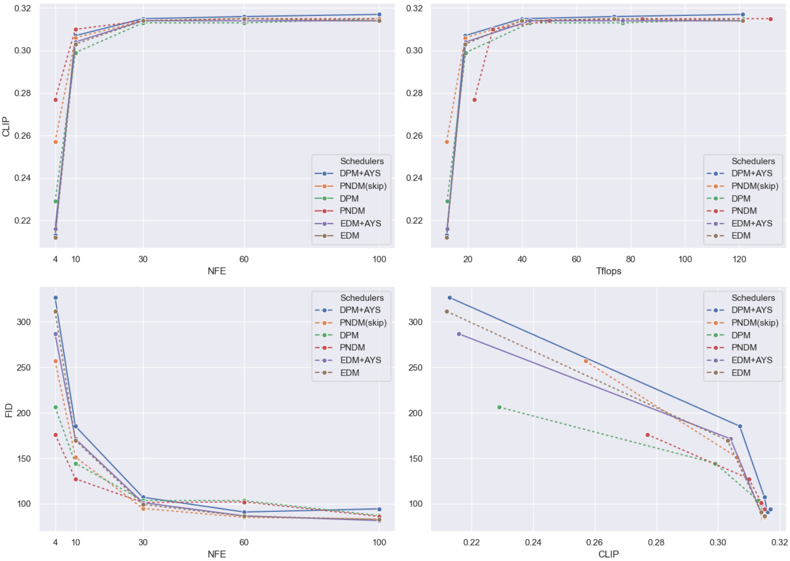
EDM Karras sigmas

- в [4] исследуется зависимость truncation error (накопленная ошибка при дискритезации шагом Эйлера) в зависимости от σ, и оказывается, что при малых значениях ошибка выше
- это приводит авторов к построению σ-based timestamps $t_i = \sigma^{-1}(\sigma_i)$, с следующей сеткой σ_i:

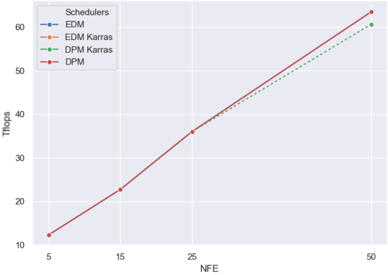
$$\sigma_{i<N} = \left(\sigma_{\max}^{\frac{1}{\rho}} + \frac{i}{N-1}\left(\sigma_{\min}^{\frac{1}{\rho}} - \sigma_{\max}^{\frac{1}{\rho}}\right)\right)^{\rho}, \sigma_N = 0$$

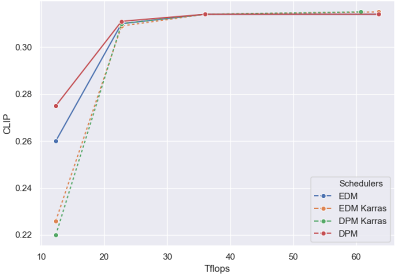
- при этом оптимальное значение $\rho = 7$

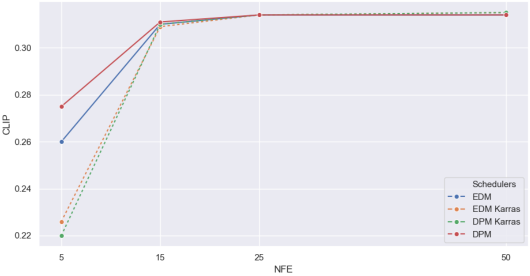
AYS


CLIP
NFE
Tflops
FID
Schedulers
DPM+AYS
PNDM(skip)
DPM
PNDM
EDM+AYS
EDM

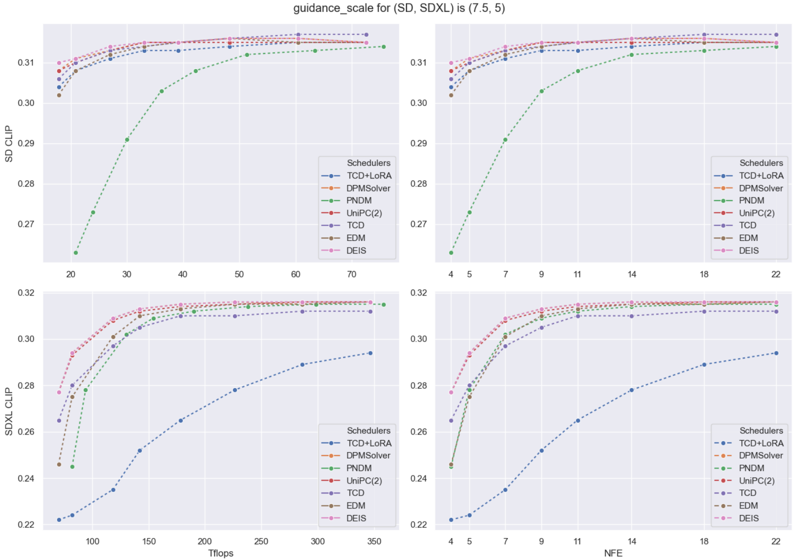
EDM Karras sigmas



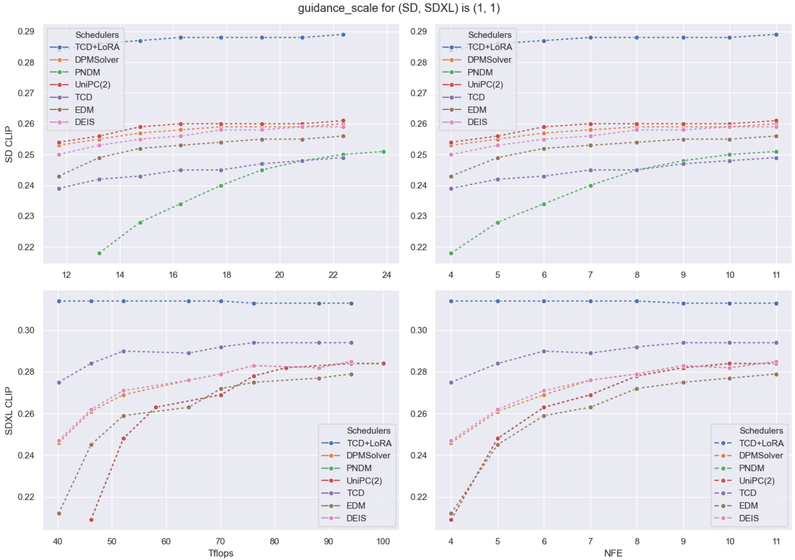




Solvers for SD, SDXL, CFG = default



Solvers for SD, SDXL, $CFG = 1$



References

- Yang Song et al SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS
- Jianfei Chen et al DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps
- Jianfei Chen et al DPM-SOLVER++: FAST SOLVER FOR GUIDED SAM-PLING OF DIFFUSION PROBABILISTIC MODELS
- Tero Karras et al Elucidating the Design Space of Diffusion-Based Generative Models
- Qinsheng Zhang et al FAST SAMPLING OF DIFFUSION MODELS WITH EXPO- NENTIAL INTEGRATOR
- Jiwen Lu et al UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models
- Zhou Zhao et al PSEUDO NUMERICAL METHODS FOR DIFFUSION MODELS ON MANIFOLDS
- Minghui Hu et al TRAJECTORY CONSISTENCY DISTILLATION: Improved Latent Consistency Distillation by Semi-Linear Consistency Function with Trajectory Mapping
- Amirmojtaba Sabour et al Align Your Steps: Optimizing Sampling Schedules in Diffusion Models