

扩散模型搭建与调参 Playbook (严格基于 6 篇论文)

严格基于 6 篇论文的工程化条目式总结 (出版排版版)

文档类型: 结构化技术综述与实践手册 (期刊可用版)

编者: 井贝贝

版本: 2026-02-14

范围: 仅总结并串联下列 6 篇论文中出现的设计选择与经验; 不额外引入论文之外的调参结论。

写作约束: 每条目严格 4 段: `怎样做 / 为何这样做 / 常见问题 / 解决方式`。

目录

目录	2
摘要	3
关键词	3
贡献与范围说明	3
阅读指南 (读者、用法、引用与符号)	3
六篇论文 (本手册唯一依据)	3
0) 第一性原理(只到够用为止)	3
1) 数据处理	6
2) 模型搭建	13
3) 损失函数选择	20
4) 优化器选择	24
5) 可视化与微调(把“采样链路”纳入调参闭环)	28
附录) 快速查表(只收录 P1-P6 中“可以直接复现”的表与数字)	33
版权与使用声明	39
参考文献 (仅 6 篇)	40

摘要

本文在严格限定的六篇文献范围内，系统梳理扩散模型训练与采样的关键设计变量，以条目化结构给出可复现的实现与调参路径。内容覆盖数据处理、模型结构、损失与优化、采样与评测，并总结少步推理与协方差学习的对照建议。文稿强调可复现性与审稿可核查性，所有条目均标注来源页码以便复核。

关键词

扩散模型；采样器；预条件；方差/协方差学习；评测协议；可复现性

贡献与范围说明

- 在统一符号与引用口径下，汇总 P1-P6 中可复现的关键设计选择与经验，形成可复核的实现清单。
- 给出“现象-定位-改动-复核”的闭环化调参路径，以及与评测链路耦合的最小实验协议。
- 明确不超出六篇论文的结论边界，并对数据、评测与采样器口径给出审稿可核查的记录要求。

阅读指南（读者、用法、引用与符号）

- 读者：做扩散模型训练/采样实现与系统消融的工程与科研读者（默认熟悉基本概率与深度学习训练）。
- 用法：以“现象-定位-改动-复核”为闭环阅读；第 5 章（可视化与微调）用于把采样链路纳入调参面板。
- 引用：每个条目标题末尾的 `引文: [P?, p.?]` 表示依据来源；`p.?` 对应论文 PDF 的页码（以 `Books/text/*.txt` 的 `== PAGE ? ==` 标记对齐）。
- 记号：`T` 总扩散步数，`t` 时间步，`alpha_bar_t` 累乘噪声系数，`sigma` 连续噪声尺度(EDM)，`NFE` 采样时网络评估次数，`FID/NLL` 常用评测指标。

六篇论文（本手册唯一依据）

- [P1] Alex Nichol, Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672, 2021.
- [P2] Jiaming Song, Chenlin Meng, Stefano Ermon. Denoising Diffusion Implicit Models. arXiv:2010.02502, 2021.
- [P3] Tero Karras, Miika Aittala, Timo Aila, Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. arXiv:2206.00364, NeurIPS 2022.
- [P4] Hang Chen, Qian Xiang, Jiaxin Hu, Meilin Ye, Chao Yu, Hao Cheng, Lei Zhang. Comprehensive exploration of diffusion models in image generation: a survey. Artificial Intelligence Review, 58:99, 2025.
- [P5] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, Yang You. Dynamic Diffusion Transformer. ICLR 2025.
- [P6] Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z. Xiao, Yingzhen Li, David Barber. Improving Probabilistic Diffusion Models with Optimal Diagonal Covariance Matching. arXiv:2406.10808, ICLR 2025.

引用格式：在每条目标题中写 `引文: [P? p?]`。

0) 第一性原理(只到够用为止)

0.0 本章上手例子: 用 `L_simple` 跑通“一步训练 + 一步可视化”的最小闭环

引文: P1 p2

怎样做	<p>这是你写训练脚本时的最小闭环, 每个符号都在本节定义。</p> <ol style="list-style-type: none"> 1) 取一批真实图像 x_0 并归一化到 $[-1, 1]$ (例如 $x = (x - 0.5) / 0.5$)。 2) 采样时间步 $t \sim \text{Uniform}(\{0 \dots T-1\})$ 与噪声 $\varepsilon \sim N(0, I)$。 3) 用边际合成 $x_t: x_t = \sqrt{\bar{\alpha}_t} * x_0 + \sqrt{1 - \bar{\alpha}_t} * \varepsilon$。 4) 网络前向: $\varepsilon_\theta = \text{model}(x_t, t)$。 5) 训练目标: $L_{\text{simple}} = E \ \varepsilon - \varepsilon_\theta \ _2^2$, 反传更新一次参数。 6) 每隔固定步数做一次可视化: 固定同一组 x_T 用确定性采样器生成网格图(见 5.0)。
为何这样做	P1 指出该目标等价于对 VLB 的重加权形式且实践上更稳定; 只要先跑通这个闭环, 后续所有“换采样器/学方差/改网络”才有可比较的基线。
常见问题	训练 loss 下降但采样像噪声, 常见原因是 $\bar{\alpha}_t$ /广播维度/采样均值方差项写错, 或可视化时归一化口径不一致。
解决方式	<ol style="list-style-type: none"> 1) 把 $\bar{\alpha}_t$ 预计算并写单元测试: 随 t 单调下降, 且在 $t = 0$ 接近 1, 在 $t = T-1$ 接近 0。 2) 固定 x_T 做一致性诊断: 只改采样步数/轨迹, 高层语义应基本一致(见 5.2)。

0.1 扩散模型究竟在学什么 (反向分布的参数化)

引文: P1 p2, P6 p1-p2

怎样做	<ol style="list-style-type: none"> 1) 写清楚前向链: 定义逐步高斯加噪 $q(x_t x_{t-1})$, 使得 x_T 近似 $N(0, I)$。 2) 预计算并缓存: $\beta_t, \alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{i=0..t} \alpha_i$ (训练与采样都会用)。 3) 选定反向参数化口径: 用网络表示 $p_\theta(x_{t-1} x_t) = N(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$, 并决定你是“只学均值/噪声”还是“均值+方差/协方差都学”(见 2.7/2.8/3.2/3.6/3.7)。 4) 采样时按 $t = T \dots 1$ 逐步从 x_T 去噪到 x_0; 如果要少步采样, 把“采样链路”当作独立模块纳入调参闭环(见第 5 章)。
为何这样做	真实后验 $q(x_{t-1} x_t)$ 依赖全数据分布不可得; 只能用可学习的高斯去近似(均值/协方差)以便从噪声逐步还原到数据。
常见问题	只学均值、协方差用固定启发式时, 当采样步数很少(小 T 或跳步)质量/似然会明显掉; 训练损失下降但采样链路仍可能“看起来像噪声”。
解决方式	<ol style="list-style-type: none"> 1) 少步优先只改采样不改训练: 用 DDIM 子序列/确定性采样画“步数-质量曲线”(见 5.1)。 2) 再改数值积分与时间点: 用 EDM/Heun 等把采样误差压下去(见 5.3)。 3) 若目标是 5/10/20 步质量与似然不塌: 把 Σ_θ 作为显式设计变量, 引入学习方差(P1)或 OCM(P6)(见 3.6-3.7, 5.7-5.8)。

0.2 为什么训练常用“预测噪声”的 MSE

引文: P1 p2

怎样做	1) 每个 batch 随机采样时间步 $t \sim \text{Uniform}(\{0 \dots T-1\})$, 并采样 $\varepsilon \sim N(0, I)$ 。 2) 用边际一次性合成 $x_t = \sqrt{\bar{\alpha}_t} * x_0 + \sqrt{1 - \bar{\alpha}_t} * \varepsilon$ (训练采样不需要逐步加噪)。 3) 把 x_t 和 t 喂给网络, 预测噪声 $\varepsilon_\theta(x_t, t)$ 。 4) 用 $L_{\text{simple}} = E \ \varepsilon - \varepsilon_\theta \ ^2$ 训练。
为何这样做	论文指出该目标可视为 VLB 的重加权形式(且更像 score matching), 实践中样本质量比直接优化 VLB 更好。
常见问题	L_{simple} 对 $\Sigma_\theta(x_t, t)$ 没有学习信号; 如果你还想学方差/协方差, 仅靠该目标不够。
解决方式	1) 若要学习方差: 启用 $L_{\text{hybrid}} = L_{\text{simple}} + \lambda * L_{\text{vlb}}$ 并配套方差参数化(见 2.8/3.2/3.3)。 2) 若要少步质量: 先改采样链路(DDIM/EDM/Heun), 再考虑协方差学习(OCM)(见 5.1-5.4, 3.6-3.7)。

0.3 “训练不变, 采样可变”: DDIM 把反向过程推广到非马尔可夫

引文: P2 p6-p7

怎样做	1) 训练阶段不变: 仍用 $T = 1000$ 与原训练目标训练模型。 2) 子序列选择 τ 按 P2 的两种明确规则二选一(把公式写死, 不留给读者猜): - Linear: $\tau_i = \text{floor}(c * i)$, 选 c 使得 τ_{N-1} 接近 T 。 - Quadratic: $\tau_i = \text{floor}(c * i^2)$, 选 c 使得 τ_{N-1} 接近 T 。 - P2 的经验选择: CIFAR-10 用 quadratic, 其余数据集用 linear (FID 略好)。 3) 设置随机性 η (P2 的定义): $\eta = 0$ 为确定性 DDIM; $\eta = 1$ 对应原始 DDPM 生成过程。 4) 采样更新式按 P2 的闭式公式实现(每一步都可复核): - $x_{\{\tau_{i-1}\}}(\eta) = \sqrt{\alpha_{\{\tau_{i-1}\}}} * ((x_{\{\tau_i\}} - \sqrt{1 - \alpha_{\{\tau_i\}}} * \varepsilon_\theta(x_{\{\tau_i\}}, \tau_i)) / \sqrt{\alpha_{\{\tau_i\}}}) + \sqrt{1 - \alpha_{\{\tau_{i-1}\}} - \sigma_{\{\tau_i\}}(\eta)^2} * \varepsilon_\theta(x_{\{\tau_i\}}, \tau_i) + \sigma_{\{\tau_i\}}(\eta) * \varepsilon$ - $\sigma_{\{\tau_i\}}(\eta) = \eta * \sqrt{(1 - \alpha_{\{\tau_{i-1}\}}) / (1 - \alpha_{\{\tau_i\}})} * \sqrt{1 - \alpha_{\{\tau_i\}} / \alpha_{\{\tau_{i-1}\}}}$ 5) 固定评测协议对比 10/20/50/100 步输出并形成“少步加速曲线”(见 5.1/5.5/5.6)。
为何这样做	P2 说明 DDIM 与 DDPM 共享相同的训练边际分布, 但反向生成过程可以选用一族非马尔可夫过程; 因此“训练一次, 推理可在不同速度-质量点上切换”。
常见问题	把“生成很慢”误认为训练不足; 或误以为要重新训练一个更小 T 的模型才能加速。
解决方式	1) 先确保“同一模型+同一初始 x_T 集合”下只改 τ / η , 否则曲线不可比(见 5.2)。 2) 若少步仍像噪声: 先查 $\bar{\alpha}_t$ 与更新式实现是否一致; 再考虑换 EDM/Heun 或引入方差/协方差学习(见 5.3-5.4, 3.2/3.7)。

0.4 “采样是数值积分”: EDM 把设计空间拆成可操作旋钮

引文: P3 p1,p45

怎样做	<ul style="list-style-type: none"> 1) 把生成显式视为“连续噪声尺度上的去噪/分数学习”。 2) 把实现拆成 5 个可独立开关的旋钮: 预条件(preconditioning)、训练噪声分布、loss weighting、采样器(确定性/随机)、离散时间点与求解器(Heun/Euler)。 3) 先固定“确定性采样器 + 固定 NFE”作为对齐基线(见 5.6), 再逐个旋钮做消融。
为何这样做	P3 的核心贡献是把“为什么一些微小实现差异会导致巨大 FID 差异”系统化, 并给出能复现提升的组合(不仅是单点技巧)。
常见问题	只改一个因素(例如只换求解器或只改权重)时收益不稳定, 甚至变差, 因为其他旋钮仍与之不匹配。
解决方式	<ul style="list-style-type: none"> 1) 每次改动只动一个“旋钮组”并写入配置记录(数据/网络/损失/采样器/NFE/时间点)。 2) 若收益不稳定: 回到确定性采样器与固定评测样本量, 先排除评测波动再判断(见 5.5-5.6)。

1) 数据处理

1.0 本章上手例子: CIFAR-10 32x32 的“可复现数据协议”

引文: P1 p11; P4 p37

怎样做	<p>给一套你能直接复现进代码与论文附录的数据协议。</p> <ul style="list-style-type: none"> 1) 数据集: CIFAR-10 训练集 50K 图像(32x32), 测试集 10K 图像(32x32)。 2) 训练用数据: 默认用完整训练集作为训练分布统计的参考(P1 的 FID 统计口径是“用 full training set”)。 3) 预处理: CIFAR-10 已是 32x32, 不做 resize; 仅做 ToTensor 与归一化到 [-1,1] (例如 $x = (x-0.5) / 0.5$)。 4) 记录协议(用于复现/审稿): 数据来源、许可与风险点; 是否做去重/坏样本过滤; 归一化口径; 训练/测试划分是否只用于评测而不混入训练(P4 强调数据风险与偏置需要披露)。
为何这样做	CIFAR-10 是最常用的扩散入门基准之一, 且 P1 专门讨论了 CIFAR-10 的过拟合现象与 schedule/dropout 的耦合; 用它能快速建立“训练-采样-评测”闭环。
常见问题	归一化口径不一致(例如训练用 [0,1], 采样/可视化按 [-1,1])会导致样本偏色或对比度异常; 训练集/测试集混用会让评测失真。
解决方式	<ul style="list-style-type: none"> 1) 把“归一化到 [-1,1]”写成全链路唯一口径(训练、采样、可视化一致)。 2) 每次更换数据过滤/增强策略后, 先抽样保存一张网格图作为数据协议的“验收样张”(见 5.0)。

1.1 选择数据集与分辨率(以及裁剪/下采样)

引文: P1 p3,p11; P2 p6; P3 p45

怎样做	<ul style="list-style-type: none"> 1) 明确任务与目标分辨率: 要生成的输出是 $32 \times 32 / 64 \times 64 / 256 \times 256$ 等哪一档。 2) 选一个“论文可对照”的基准数据集作为起点: CIFAR-10(32×32)、ImageNet-64、CelebA(64×64)、LSUN(256×256)。 3) 把预处理写成确定性协议并固定: 中心裁剪到正方形 + area downsample 到目标分辨率 + 归一化范围 (并记录随机种子与库版本)。 4) 若最终任务更难(更高分辨率/更复杂分布): 先在低分辨率基准把训练-采样闭环跑通, 再迁移升级。
为何这样做	分辨率和多样性决定训练难度与过拟合风险; P1 直接将 ImageNet-64 作为“多样性与可训练性”的折中来做系统消融。
常见问题	分辨率/多样性更高时更难优化, 需要更长训练与更强算力; 低分辨率容易被误判为“模型 work 了”但迁移失败。
解决方式	<ul style="list-style-type: none"> 1) 固定采样器与评测协议(步数/NFE/样本量/是否用 EMA)后做对照实验(见第 5 章)。 2) 迁移到更高分辨率前, 先复核预处理协议是否一致 (裁剪与下采样的差异会直接改变训练分布)。

1.2 数据集限制与数据稀缺(研究规划层面)

引文: P4 p37

怎样做	<p>不翻论文也能落地判断“规模/质量/偏置”，且完全可复现。</p> <p>1) 先给“规模”分档(用 6 篇论文中反复出现的量级做锚点，方便写论文/报告时不含糊):</p> <ul style="list-style-type: none"> - $5e4$ 级: CIFAR-10 量级(训练集 50K)。 - $1e6$ 级: ImageNet 量级(常见做法是百万级图像; P1/P3 的 ImageNet-64 都属于这一档的典型难度)。 - $1e9$ 级: text-to-image 常见 “billions of (text,image) pairs”(P4 以此解释 T2I 质量跃迁的现实前提)。 <p>2) 再把 P4 的 “high-quality”落地成一张必须交付的《数据质检统计表》(只做统计，不在这里引入额外“阈值结论”):</p> <ul style="list-style-type: none"> - 抽样口径写死: 统一抽样 $K = 50,000$ 个样本做质检(选 50K 是为了与 P3 的 FID 评测样本量同量级，便于对齐统计口径); 若全量不足 50K，则全量质检并写明 $K = D$。 - 可用性(坏样本): 逐个解码抽样样本，统计 $\text{corrupt}_{\text{count}} / K$; 同时记录失败原因分布(文件损坏/解码器不支持/尺寸为 0 等)。 - 分辨率与压缩质量: 统计宽高分布、短边分布、(若有)JPEG quality/文件大小分布；给出分位数表(Q1/median/Q3/p95)。 - 重复/近重复: 统计“完全重复”(字节级 hash)与“近重复”(感知 hash 或 embedding 相似度)的比例与口径；必须写清你用的算法与阈值(否则不可复现)。 - 任务对齐噪声: 仅在你做条件生成/标注任务时填写： - text-image: 抽检/人工复核错配样本比例，并记录错配类型(描述过泛/指代错误/不相关等)。 - class label: 抽检错标比例与常见错标对(例如 cat↔dog)。 <p>3) 最后按 P4 提醒的维度做“偏置风险披露”(不是让你立刻纠偏，而是让读者知道你的数据覆盖缺口在哪里):</p> <ul style="list-style-type: none"> - 至少按 language / ethnicity / gender 三个轴做一次分层统计或抽检(取决于你任务是否包含人相关内容)；输出结论必须包含：覆盖缺口 + 可能影响的生成偏差类型 + 你是否在评测时做了分层评测。
为何这样做	P4 明确指出扩散模型快速发展与“大规模高质量数据”强相关；但很多子任务会遭遇数据稀缺与数据偏置，进而引发质量与公平性问题。
常见问题	小数据导致过拟合/泛化差；数据偏置导致生成结果偏斜并引发下游风险。
解决方式	<p>1) 先把“是否稀缺导致过拟合”用可观测信号确认：固定采样器与 NFE，随训练保存 snapshot 并画 FID/NLL 曲线(见 5.6)。</p> <p>2) 若确认稀缺：优先上 non-leaky augmentation 等正则(见 1.3)并用确定性采样器锁死评测链路。</p> <p>3) 若更像“质量/偏置问题”而非“规模”：回到 1.5 的出版级数据卡与抽检统计补齐，再谈模型调参。</p>

1.3 非泄漏增强(non-leaky augmentation)作为正则

引文: P3 p45

怎样做	<ol style="list-style-type: none"> 1) 选一组增强(几何/颜色等), 把增强的随机参数记为 α。 2) 训练时: 对输入图像应用增强, 同时把 α 作为条件输入给网络(让模型“知道你增强了什么”)。 3) 约定 $\alpha = 0$ 表示“无增强”; 采样/评测时固定 $\alpha = 0$, 保证生成分布不带增强痕迹。 4) 按 P3 的实现策略: 禁用 dataset x-flips, 只通过 non-leaky augmentation 获得增强收益, 避免生成镜像文字/Logo。 5) 把“增强启用概率”写成明确超参并记录: P3 的训练表给出示例 augment probability 为 CIFAR-10: 12%, FFHQ/AFHQ: 15% (用于其 improved 配置对照)。
为何这样做	P3 解释其思路是把增强当作辅助任务/正则, 但通过条件输入避免增强“泄漏到生成图像”; 甚至可用 100% x-flip 获取增强收益而不让生成出现镜像文字/Logo。
常见问题	传统数据增强会让生成分布出现 out-of-domain 几何变换(例如镜像文字); 或者增强越强 FID 越不稳定。
解决方式	<ol style="list-style-type: none"> 1) 若你现在用的是“直接增强不喂参数”: 改成 non-leaky 口径并重新做一次对照评测。 2) 若发现增强让 FID 抖动: 降低增强强度或只保留对任务无害的子集, 并保持 $\alpha = 0$ 的采样约定不变。

1.4 何时不需要增强正则: 以“是否担心过拟合”为准

引文: P3 p45-p46

怎样做	<ol style="list-style-type: none"> 1) 先固定评测链路: 用确定性采样器 + 固定 NFE + 固定样本量(见 5.6)。 2) 用训练过程信号判断过拟合: 若训练 loss 下降但固定采样器下 FID 在后期持续变差, 视为过拟合/泛化退化信号(P3)。 3) 用生成多样性信号复核: 若样本更像训练集近邻/结构重复, 可结合 precision/recall 的覆盖视角做判断(P1)。 4) 若无明显过拟合信号: 不开增强正则也成立(P3 在 ImageNet-64 配置中明确采用该策略)。
为何这样做	增强正则会改变训练分布并引入额外超参, 只有在过拟合成为瓶颈时才值得付出复杂度; P3 把这当作经验性选择而非默认开关。
常见问题	在本就不易过拟合的大数据上盲目加增强, 可能只增加调参负担; 或造成与目标分布不一致的副作用。
解决方式	<ol style="list-style-type: none"> 1) 只有确认过拟合后才启用增强; 且优先用 non-leaky augmentation 以避免分布泄漏(见 1.3)。 2) 增强开启后, 继续用相同评测链路追踪 FID/NLL 是否稳定改善, 不要只看训练 loss。

1.5 数据质量的最小“出版级”清单(用于复现与审稿)

引文: P1 p11; P3 p45; P4 p37

怎样做	<p>出版/审稿可复核口径。</p> <ol style="list-style-type: none"> 1) 写明来源与权限: 数据来源、许可证/使用范围、是否含版权/隐私风险点(P4)。 2) 写明规模与分布: 样本数、分辨率分布、(若适用) 类别/文本长度分布、明显长尾与缺口。 3) 写明质检与过滤: 抽样网格可视化; 损坏/不可读比例; 重复/近重复口径与比例; 水印/低清/错配的处理策略。 4) 写明预处理协议: <code>center crop + area downsample</code> 的参数; 归一化口径; 随机种子; 代码与库版本。 5) 写明训练-采样协议: 噪声调度/噪声分布(如 P3 的 $p_{train}(\sigma)$)、采样器/求解器与固定 NFE(见 5.6)。 6) 写明偏置抽检: 至少对 <code>language/ethnicity/gender</code> 等维度给出一次分层统计或抽检结论与风险说明(P4)。
为何这样做	这 6 篇论文共同的隐含前提是“数据与预处理可被他人复现”，否则模型/采样器/损失的消融结论会被数据差异掩盖；同时 P4 指出数据伦理与隐私是必须面对的外部约束。
常见问题	只写“我们做了清洗/过滤”但不给统计，导致不可复现；预处理细节缺失(裁剪/下采样/归一化口径不一致)导致复现实验偏差；数据偏置未披露导致生成结果系统性偏斜。
解决方式	<ol style="list-style-type: none"> 1) 把数据卡与统计表作为训练前置门槛: 任何大训练前先跑一遍质检/统计并把结果存档。 2) 发现“看似调参无效”: 先回查是否是预处理口径不一致或数据质量/偏置导致的分布漂移，再继续改模型。

1.6 数据集画像: 为主干选型提供“可复核输入”

引文: P4 p37; P1 p11; P3 p45-p46

怎样做	<p>在选主干之前, 先把数据集压缩成一份可复核的“画像表”(写进实验日志/附录), 使后续选型与对照实验有共同输入。</p> <ol style="list-style-type: none"> 1) 基本信息: D (样本数)、模态(无条件/类别条件/text-image)、许可与敏感风险点(P4)。 2) 分辨率与预处理协议: 原始分辨率分布、目标分辨率 H×W、裁剪/下采样方法与参数(见 1.1/1.5)。 3) 质量与噪声: 损坏/不可读比例、压缩伪影/水印比例、重复/近重复比例与口径(见 1.2/1.5; P4)。 4) 规模分档(仅用于叙述与对照, 不引入阈值结论): 参考 P4 的讨论把规模标注为 $5e4 / 1e6 / 1e9$ 量级并写明采样口径(见 1.2)。 5) 过拟合风险信号(用于决定正则与增强, 不是直接决定主干): 先用固定采样器与固定 NFE 跑基线, 观察训练后期 FID 是否持续变差(P3)与覆盖是否退化(P1)(见 1.4/5.6)。 6) 目标推理约束: 计划汇报的步数/NFE 面板(例如 5/10/20/50/100 步或固定 NFE), 以及是否必须少步推理(见第 5 章, P2/P3/P6)。
为何这样做	P4 强调数据规模/质量/偏置会显著影响生成质量与风险; 同时 P1/P3 的大量消融表明“最优结构与正则配置”与分辨率、过拟合程度强耦合。没有画像表, 主干选型与后续改动很难归因与复现。
常见问题	直接从网络结构开始试, 但数据预处理/质量/规模与推理约束没锁定, 导致“换主干/换采样器”的差异被数据口径变化掩盖。
解决方式	<ol style="list-style-type: none"> 1) 先冻结 1.5 的数据协议与 5.6 的评测选点协议, 任何主干对照都复用同一画像表与评测面板。 2) 若数据画像本身不稳定(过滤/裁剪策略频繁改), 先把数据协议收敛到稳定版本再做主干对照。

1.7 从画像到主干候选集: 优先匹配“论文锚点任务”

引文: P1 p11; P3 p45-p46; P5 p17; P4 p37

怎样做	<p>在本手册“仅依赖 P1-P6”的约束下, 主干选型不追求泛化规则, 而是用“与论文实验最接近的锚点任务”给出可复现起步基线, 再用离散档位做最小扫点。</p> <p>1) 先按分辨率/任务把数据集映射到最近锚点:</p> <ul style="list-style-type: none"> - 32×32 无条件(或弱条件) → 以 CIFAR-10 作为锚点, 主干优先采用 P1 的 U-Net 档位与正则网格(见 2.1/4.5)或采用 P3 的 improved 训练/网络配置作为锚点起点(见 4.5 附录表)。 - 64×64 无条件/类别条件 → 以 ImageNet-64 作为锚点, 主干优先采用 P1 的 ImageNet-64 U-Net 配置表作为“可复现默认”(见 2.0); 若按 P3 体系实现, 则以其 ImageNet 相关网络/训练表为锚点起步(见附录表)。 - $256 \times 256 / 512 \times 512$ 且使用 DiT 体系 → 仅在与你任务设定相近(如 ImageNet 256/512)时, 以 P5 的 DiT-S/B/XL 离散档位作为主干起点(见 2.6*/附录 A.1)。 <p>2) 若目标数据集不落在上述锚点分辨率上: 先做一个“锚点分辨率 pilot”(按 1.1 统一裁剪/下采样到 $32 / 64 / 256$ 之一), 用同一评测协议比较主干趋势; 再把最稳的主干迁移到目标分辨率做复核(不在此处引入论文之外的最优性结论)。</p> <p>3) 条件信息决定“条件注入形态”而不是直接决定主干范式: 按 P1 的 scale-and-shift 条件注入写成固定实现(见 2.3), 确保不同主干对照时条件路径一致。</p>
为何这样做	P1/P3/P5 分别在其锚点任务上给出可复现的主干规模与配置表; 在缺少跨数据集的统一选型结论时, “先匹配锚点并复现基线”能最大化可比性与复核性, 也符合 P4 对可复现数据协议与风险披露的要求。
常见问题	在新数据集上“自造结构/自造规模”, 同时改预处理与采样器, 导致无法判断到底是主干、训练、采样还是数据口径导致效果变化。
解决方式	<p>1) 主干对照阶段只允许改“主干与其规模档位”, 其余(数据协议、训练时长、采样器/NFE、评测样本量)全部冻结(见 5.5-5.6)。</p> <p>2) 对照只在离散档位内进行: U-Net 按 P1 的 $C = 64 / 96 / 128 / 192$ 档位, DiT 按 P5 的 S/B/XL 档位; 避免引入不可对照的中间规模。</p>

1.8 主干选定后的最小扫点: “离散档位 + 联动超参”

引文: P1 p8,p11; P3 p45; P5 p17

怎样做	<p>选定一个主干家族后, 用论文给出的离散规模档位做最小扫点, 并按论文口径联动关键超参, 以获得可复核的“规模-质量”曲线。</p> <ol style="list-style-type: none"> 1) 若用 P1 风格 U-Net: 只在 $C = 64 / 96 / 128 / 192$ 四档内扫, 并按 P1 的学习率缩放规则联动 lr(见附录 A.6); 其余结构字段(下采样次数、resblocks、注意力分辨率)先固定为锚点配置(见 2.0-2.2)。 2) 若用 P5 风格 DiT: 只在 DiT-S/B/XL 三档内扫, 并沿用其微调/训练协议的关键字段(warmup、batch、λ 网格等)作为可复现起点(见附录 A.1-A.2)。 3) 若用 P3 的 improved 配置: 主干/训练/采样旋钮强耦合, 先整体复现其配置组作为锚点, 再做“一次只改一个旋钮组”的消融(见 0.4/4.5/5.3-5.4)。 4) 选点与报告: 按 5.6 用固定确定性采样器与固定 NFE 选最优 snapshot, 并同步报告该主干档位下的 FID/NLL/覆盖指标面板(见 5.5-5.6)。
为何这样做	P1/P5 明确展示“规模档位+联动超参”的可复现对照方式; P3 强调配置耦合, 随意打散会破坏复现与结论稳定性。用离散档位扫点能避免将不必要的自由度引入审稿不可复核的实验。
常见问题	扫规模时只改主干不改学习率/训练协议, 导致收敛性质变化被误当作“主干更好/更差”; 或者把 P3 的配置组拆散单改一个超参而得出不稳定结论。
解决方式	<ol style="list-style-type: none"> 1) 把“规模档位→联动超参”的规则写进配置系统并强制日志打印, 避免实验过程中被手动覆盖。 2) 若资源有限, 先做最小档位对照(例如 U-Net: C=64 vs 128; DiT: S vs B), 在固定评测协议下确认趋势再扩展扫点。

2) 模型搭建

2.0 本章上手例子：直接复现 P1 的 ImageNet-64 U-Net 架构配置表

引文: P1 p11

怎样做	<p>把这段当作“配置文件”，读者直接复现即可实现同款骨干。</p> <ol style="list-style-type: none"> 1) 输入输出分辨率: 64×64。 2) 下采样链: 4 次下采样，分辨率为 $64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4$。 3) 每个下采样 stage 的残差块数: 3 (共 4 个 stage, 每个 stage 3 个 resblocks)。 4) 上采样链: 镜像对称(4 次上采样, 每个 stage 3 个 resblocks)。 5) 通道表(把每个分辨率的通道写死, 避免读者猜“4×4 到底是多少通道”): <ul style="list-style-type: none"> - 设 $C = 128$ - 64×64: 128 - 32×32: 256 - 16×16: 384 - 8×8: 512 - 4×4(瓶颈): 512 6) 注意力: 在 16×16 与 8×8 两个分辨率插入 self-attention; 4 heads(总通道数不变)(见 2.2)。 7) 条件注入: 用 scale-and-shift 形式 $\text{GroupNorm}(h) * (w + 1) + b$(见 2.3)。
为何这样做	这是一套 P1 在 ImageNet-64 的系统消融中使用的明确架构，其好处是“每个数字都可被复现”，且规模可用 C 单旋钮缩放。
常见问题	读者只记住“U-Net + 注意力”但漏掉了“下采样次数/每 stage resblocks/通道表”，导致实现的计算量与感受野完全不同，最终效果不可比。
解决方式	<ol style="list-style-type: none"> 1) 把上面的 7 个字段写进你自己的 config，并在每次实验输出里把它们打印出来(防止误用旧配置)。 2) 任何“改深度/改宽度”都必须落到这 7 个字段的改动上，并在固定采样协议下对齐评测(见 5.6)。

2.1 以 U-Net 为主干时, 通道/分辨率/残差块的组织方式

引文: P1 p11

怎样做	<p>1) 先把分辨率链写成确定值(不要留给读者猜): 设最低分辨率为 4×4, 则下采样次数 $n_{\text{down}} = \log_2(H / 4)$。</p> <ul style="list-style-type: none"> - 例: $64 \times 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4$, 所以 $n_{\text{down}} = 4$。 - 例: $32 \times 32 \rightarrow 16 \rightarrow 8 \rightarrow 4$, 所以 $n_{\text{down}} = 3$。 - 例: $256 \times 256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4$, 所以 $n_{\text{down}} = 6$。 <p>2) 按 P1 的 ImageNet-64 配置给出一套可直接复现的“默认 U-Net”(把每个数字写死):</p> <ul style="list-style-type: none"> - 分辨率 stages(高->低): $64, 32, 16, 8, 4$ - 下采样次数: 4 - 每个下采样 stage 的 residual blocks: 3 - 上采样堆栈: 作为下采样堆栈的镜像(同样每个 stage 3 个 residual blocks) - 通道设计(按分辨率写死): 设 $C = 128$, 则 <ul style="list-style-type: none"> - 64×64: 128 - 32×32: 256 - 16×16: 384 - 8×8: 512 - 4×4(瓶颈): 512 - 注意力插入点与 heads: 见 2.2 (P1 为 16×16 与 8×8, 4 heads) - 规模锚点(用于报告/复现): $C = 128$ 时约 120M 参数、单次前向约 39B FLOPs <p>3) 再把“通道 C 取多少”写成离散档位(不要留给读者主观设定): P1 在 ImageNet-64 给出 4 个可复现档位, 你只需要在它们中选一个起跑:</p> <ul style="list-style-type: none"> - first-layer channels($=C$): $64 / 96 / 128 / 192$ - 参数量(论文图注给出): $30M / 68M / 120M / 270M$ - 学习率缩放规则(P1 明确写法): 以 128 通道模型 $\text{lr} = 1e-4$ 为基准, 其他模型用 $\text{lr} = 1e-4 / \sqrt{C / 128}$。 - 例: $C=64 \rightarrow \text{lr} \approx 1.41e-4$; $C=96 \rightarrow \text{lr} \approx 1.15e-4$; $C=192 \rightarrow \text{lr} \approx 8.16e-5$ - 写实验记录时, 只允许写 “$C=64/96/128/192$ 之一”, 不允许写 “$C=160$(随便试的)” 这种不可对照的规模。 <p>4) 按 P1 的 CIFAR-10 配置给出一套可直接复现的“小模型”(同样把数字写死):</p> <ul style="list-style-type: none"> - 分辨率 stages(高->低): $32, 16, 8, 4$ - 下采样次数: 3 - 每个下采样 stage 的 residual blocks: 3 - 通道设计(按分辨率写死): 设 $C = 128$, 则 <ul style="list-style-type: none"> - 32×32: 128 - 16×16: 256 - 8×8: 256 - 4×4(瓶颈): 256 <p>5) 把“你实际用了什么”写成配置表并随实验固定: 分辨率链、通道表、resblocks、注意力分辨率与 heads、是否用 scale-shift 条件注入(见 2.3)。</p>
为何这样做	P1 在 ImageNet-64 上系统消融, 给出一套可扩展的通道分配与 block 堆叠方式, 并能用 C 直接控制参数量与 FLOPs(用于规模化实验)。
常见问题	规模增大后, 训练超参(尤其学习率/批量)不再“原封不动”适用; 结构加深也可能只带来似然提升却伤害 FID。

解决方式	<p>1) 每次扩大 C 或加深 block 之前, 先把 batch / lr / EMA 一起写成联动超参方案(见 4.2)。</p> <p>2) 评估不要只看训练 loss: 固定采样器与样本量后同时看 FID 与 NLL/似然侧指标(见 5.5/5.7)。</p>
------	--

2.2 注意力层与多头设置(在 U-Net 的哪些分辨率插入)

引文: P1 p11

怎样做	<p>1) 先把注意力“只放在少数分辨率”写成配置项 (避免全分辨率都上注意力导致 FLOPs 爆炸)。</p> <p>2) 按 P1 的设置从 16x16 与 8x8 两个分辨率插入注意力开始。</p> <p>3) 多头设置用 4 heads 作起点, 并保持总通道不变(P1)。</p> <p>4) 把“4 heads”落实成实现定义(避免读者猜): 对某层通道数为 d_{model} 的注意力层, 每个 head 的通道为 $d_{head} = d_{model} / 4$, 并保持 Q/K/V 投影输出总维度为 d_{model}。</p>
为何这样做	P1 报告这些改动在 ImageNet-64 上带来轻微 FID 改善, 属于“在不改变主干范式前提下”的质量增强。
常见问题	注意力层更容易在混合精度下出数值稳定性问题(尤其 FP16 指数范围)。
解决方式	<p>1) 先在固定采样协议下做 3 组消融: 无注意力 / 只 16x16 / 16x16+8x8, 确认收益与 FLOPs 代价。</p> <p>2) 若混合精度不稳: 先回退到 FP32 把“注意力是否带来收益”验证清楚; 只在吞吐成为瓶颈时再启用 FP16, 并用同一评测协议复核质量不退化(见 4.3/5.6)。</p>

2.3 时间/噪声条件注入方式(Scale-Shift 形式)

引文: P1 p11

怎样做	<p>1) 为每个 residual block 准备时间/噪声嵌入向量 $emb(t)$。</p> <p>2) 用 MLP 从 $emb(t)$ 预测 w, b 两路参数, 并把维度写死: 若该 block 的通道数为 c, 则输出 w, b 都是长度为 c 的向量(对应每个通道一个缩放与平移)。</p> <p>3) 在归一化后做 scale-and-shift: $GroupNorm(h) * (w + 1) + b$ (P1 的明确写法), 不要用 $GroupNorm(h + v)$ 这类“先加再归一化”的写法。</p>
为何这样做	P1 指出该修改对 ImageNet-64 的 FID 有小幅提升; 本质是让条件以仿射形式调制归一化后的特征, 更灵活。
常见问题	条件注入写法不一致会导致复现实验失败; 还可能导致不同实现间的差距被误归因到“别的因素”。
解决方式	<p>1) 把条件注入公式写进配置与日志(与 schedule/采样器同等重要)。</p> <p>2) 若复现失败, 优先排查注入点(GN 前/后)、$w + 1$ 写法、以及 t 的嵌入维度/缩放是否一致。</p>

2.4 EDM 的网络预条件(preconditioning)与噪声嵌入

引文: P3 p45-p46

怎样做	<p>1) 按 P3(EDM)把网络写成显式预条件形式(把公式写死, 读者直接复现即可):</p> <ul style="list-style-type: none"> - 设 $\sigma_{\text{data}} = 0.5$ (P3 的默认常量) - $c_{\text{skip}}(\sigma) = \sigma_{\text{data}}^2 / (\sigma^2 + \sigma_{\text{data}}^2)$ - $c_{\text{out}}(\sigma) = \sigma * \sigma_{\text{data}} / \sqrt{(\sigma^2 + \sigma_{\text{data}}^2)}$ - $c_{\text{in}}(\sigma) = 1 / \sqrt{(\sigma^2 + \sigma_{\text{data}}^2)}$ - $c_{\text{noise}}(\sigma) = \log(\sigma) / 4$ (噪声条件输入的标量化口径) <p>2) 选择噪声嵌入实现并写死: P3 对比过 DDPM positional 与 NCSN++ random Fourier features; 你必须在实现里明确选一种, 不要“随意设定一个 embedding”。</p> <p>3) 配套训练端把 $p_{\text{train}}(\sigma)$ 与 $\lambda(\sigma)$ 写成配置并与预条件一起启用(见 3.4), 否则预条件很难单独带来稳定收益。</p>
为何这样做	P3 把“网络形式 + 预条件 + 训练噪声分布 + loss weighting + 采样器”视为耦合的设计空间, 预条件能改善不同噪声级别上的训练与采样行为。
常见问题	不同噪声级别的误差分布差异很大, 直接训练容易出现某些 sigma 区间支配梯度, 生成表现不均衡。
解决方式	<p>1) 把预条件(2.4)、$p_{\text{train}}(\sigma)/\lambda(\sigma)$(3.4)、采样器(5.3-5.4)绑定成同一个配置组一起开关。</p> <p>2) 固定评测链路(确定性采样器 + 固定 NFE + 固定样本量)后逐项消融, 定位是“训练端”还是“采样端”造成的收益/退化。</p>

2.5 结构容量如何“重新分配”以减少过拟合

引文: P3 p46

怎样做	<p>1) 固定总参数量/训练预算, 只改“容量分布”(避免把收益误归因到算力变大)。</p> <p>2) 按 P3 的再平衡思路: 减少最小分辨率(例如 4x4)的层数/宽度, 把容量挪到中等分辨率(例如 16x16)。</p> <p>3) 用固定采样协议对齐比较, 观察是否更不易出现过拟合/重复样本。</p>
为何这样做	该经验直接来自 P3 的 config 消融, 说明“更深/更低分辨率”不一定更好, 容量该放在更关键的分辨率段。
常见问题	一味增加最低分辨率层数, 训练 loss 可能更低但 FID/视觉质量变差(过拟合或样本多样性下降)。
解决方式	<p>1) 固定总参数量/训练预算, 只改“容量分布”(4x4 vs 16x16 等)。</p> <p>2) 用固定采样器与固定样本量对齐评测, 画出训练进程中的 FID 曲线与样本网格。</p> <p>3) 若过拟合更轻或质量更稳, 再把该容量分布作为新默认并继续调优化超参(见 4.2)。</p>

2.6 Transformer 路线的“动态计算”(DyDiT)

引文: P5 p1,p5-p6

怎样做	<ol style="list-style-type: none"> 1) 明确动态化粒度(P5 提供两条主线): (a) timestep-wise: 不同 t 使用不同计算; (b) token-wise: 对部分 token 允许绕过计算(例如 MLP bypass)。 2) 先从 token-wise 的 MLP bypass 做起点(保留注意力交互), 把路由器/动态 mask 作为额外模块接到主干。 3) 用 STE/Gumbel-Sigmoid 训练路由器, 并引入 FLOPs 约束项把预算写进目标(见 3.5)。
为何这样做	P5 观察到不同时间步与不同空间 patch 的噪声预测难度不均衡; 动态化把算力集中在“更难的步/更难的 token”, 在相同或更低 FLOPs 下改善/保持质量。
常见问题	直接用联合损失 finetune 可能不稳定; 动态网络往往因为 sample-dependent graph 导致批量推理效率差。
解决方式	<ol style="list-style-type: none"> 1) 先让动态结构“只依赖 timestep”: 把 mask/activation 预计算, 避免 sample-dependent graph 造成推理吞吐下降(P5)。 2) 若训练不稳: 启用 warm-up, 保留 L_{complete} 一段时间后再移除(见 4.4)。 3) 用 FLOPs 约束项验证预算是否达标, 再看质量是否保住(见 3.5)。

2.6* Transformer 主干(静态 DiT)的“深度/头数/宽度”配置表(可直接复现)

引文: P5 p17 (Table 7)

怎样做	<p>这张表专门回答“Transformer 的 depth 与 heads 选多少”, 不需要读者翻论文。</p> <ol style="list-style-type: none"> 1) 把 Transformer 的 3 个核心结构超参写成显式配置项: <ul style="list-style-type: none"> - layers(depth): block 数 - heads: 多头数 - channel(width, 也可写 d_{model}): token 的通道维度 2) 直接从 P5 的 Table 7 选一个离散档位作为“静态 DiT 主干”的起点(不要自造深度/头数): <ul style="list-style-type: none"> - model params(M) layers heads channel pre-training source - DiT-S 33 12 6 384 5M iter (Pan et al., 2024) - DiT-B 130 12 12 768 1.6M iter (Pan et al., 2024) - DiT-XLI 675 28 16 1152 7M iter (Peebles & Xie, 2023) 3) 若要做 DyDiT: 先把静态 DiT 跑通(训练-采样-评测闭环), 再加路由器做动态化; P5 表里指出 DyDiT 在这些档位上“layers/heads/channel 不变”, 只是引入可控的激活 mask 与少量额外参数。
为何这样做	P5 的贡献是“在既有 DiT 档位上做动态计算并节省 FLOPs”; 因此它把 DiT 的规模定义为可复现的离散档位(层数/头数/宽度), 复现或微调必须从同一档位起跑才可对照。
常见问题	自己主观设定选 layers / head / channel 会导致算力与容量完全不可比, 训练/采样表现差异无法归因; 还容易把“patch/token 设计”与“模型规模”混在一起造成混乱。
解决方式	<ol style="list-style-type: none"> 1) 先固定一个档位(推荐先从 DiT-S: 12 layers, 6 heads, width=384 起步), 用固定采样协议跑通可视化与评测闭环(见 5.0/5.5/5.6)。 2) 升级规模只允许跳档位: S -> B -> XL; 且每次升级都要同时记录训练超参(优化器 /lr/batch/warmup)是否按论文口径联动(P5 p17 的 Table 6; 见附录 A.2)。

2.7 协方差/方差建模作为“加速采样”的结构扩展

引文: P1 p2-p3, P6 p1-p4

怎样做	1) 先写清楚你的目标: 是“少步(NFE 5/10/20)质量”还是“高 NLL/似然”还是两者都要。 2) 若目标包含少步或似然: 把 Σ_θ 作为结构扩展项进入设计清单, 而不是固定启发式。 3) 两条论文内路线二选一或组合: - P1 路线: 用 L_{hybrid} 学习方差(见 2.8/3.2/3.3)。 - P6 路线: 用 OCM 摊销预测最优对角协方差(见 3.7/5.8)。
为何这样做	P6 明确指出协方差选择在少步采样时影响巨大; 更好的对角协方差能提高少 NFE 下的质量、recall 与似然。
常见问题	直接用启发式 β 或 β 的固定方差在小步数下会很差; 直接用 Hessian 估计对角项计算成本太高(需要大量网络评估)。
解决方式	1) 如果你的瓶颈是少步质量: 优先先换采样过程(DDIM/EDM/Heun)确认“链路可加速”(见 5.1-5.4)。 2) 若仍差: 再引入方差/协方差学习。 - 学方差: $L_{\text{hybrid}} +$ 有界插值(见 3.2/2.8)。 - 学对角协方差: OCM 训练协方差头(见 3.7/5.8)。

2.8 学习方差的参数化: 在上下界之间插值

引文: P1 p3-p4

怎样做	1) 定义上下界: $\sigma_t^2 = \beta_t$ 与 $\sigma_t^2 = \beta$ (P1 把它们作为可行边界)。 2) 让网络输出插值系数 v , 在 log-variance 空间把方差约束在二者之间 (保持正定与数值稳定)。 3) 启用 L_{hybrid} 训练方差(否则 L_{simple} 对方差无学习信号, 见 3.2)。
为何这样做	直接预测 unconstrained 方差容易出现数值不稳定或越界; 在合理区间内学习能更稳定地改进采样步数减少时的质量与似然。
常见问题	学习到的方差过大导致短轨迹噪声过重; 过小导致多样性/覆盖下降或采样不稳。
解决方式	1) 确认方差头有学习信号: 必须启用 L_{hybrid} 而非仅 L_{simple} (见 3.2)。 2) 训练时同时画“少步步数-质量曲线”: 10/20/50/100 步(DDIM/确定性)作为验证面板(见 5.1/5.5)。 3) 若出现噪声过重/多样性下降: 回到插值范围与 λ 权重, 先缩小 VLB 权重再复核。

2.9 噪声调度(Linear vs Cosine)是训练与采样的核心旋钮

引文: P1 p4-p5

怎样做	1) 实现 Linear 与 Cosine 两套调度并可切换, 确保 $\bar{\alpha}_t$ 计算一致。 2) 在同一结构/同一采样协议下做 schedule 对照: 同时报 FID 与 bits/dim(NLL)。 3) 若用 Cosine: 把 dropout/训练时长作为联动超参一起调(见 4.5), 并用更密集 snapshot 曲线检查是否更快进入过拟合区间(见 5.6)。
为何这样做	P1 观察线性调度在后 1/4 步骤很快变成几乎纯噪声, 余弦调度更慢地加噪; 在其消融中, 余弦调度与 L_{hybrid} 的组合整体更优。
常见问题	余弦调度可能更容易过拟合(论文讨论其可能原因包括“噪声更少导致正则更弱/优化更快导致过拟合更明显”)。
解决方式	1) schedule 改动必须联动正则: 在 Cosine 下做 dropout 小网格, 不要只换 schedule(见 4.5)。 2) 用 snapshot+确定性采样器评测曲线定位过拟合窗口: “loss 降但 FID 升”时优先考虑正则/早停而非继续堆算力(见 5.6)。

3) 损失函数选择

3.0 本章上手例子: 用 `L_hybrid` 学方差的最小实现(输出头 + loss 计算)

引文: P1 p3-p4

怎样做	把它当作“实现清单”, 从模型输出到 loss 都写死。 1) 模型输出至少包含两路: - 噪声预测 $\varepsilon_\theta(x_t, t)$ (用于 L_{simple}) - 方差插值系数 $v_\theta(x_t, t)$ (用于构造 Σ_θ , 见 2.8) 2) 用 2.8 的“有界插值”构造对角方差 Σ_θ (把方差限制在 β_t 与 β 之间)。 3) 计算两项 loss 并加权: - $L_{\text{simple}} = E \ \varepsilon - \varepsilon_\theta\ ^2$ - L_{vlb} (与 variational lower bound 对应的项, 其梯度用于训练方差分支) - $L_{\text{hybrid}} = L_{\text{simple}} + \lambda * L_{\text{vlb}}$, 并从 $\lambda = 0.001$ (P1) 起步。 4) 日志必须拆开记录: L_{simple} 、 L_{vlb} 、 L_{hybrid} 三条曲线; 否则你无法判断是“主干学坏了”还是“方差项在抖”。
为何这样做	P1 明确指出 L_{simple} 不给方差学习信号; L_{hybrid} 用很小的权重把 VLB 加回去, 让方差能学到而不显著伤害样本质量。
常见问题	只开了方差头但仍只训练 L_{simple} , 导致方差头永远不收敛; 或 λ 过大使 VLB 抖动压垮训练。
解决方式	1) 若 L_{vlb} 抖动大: 先启用时间步重采样/重要性采样稳定它(见 3.3)。 2) 若样本质量下降: 先把 λ 再降一档并复核; 不要先去改网络结构。

3.1 `L_simple` 与 VLB 的关系(以及何时用谁)

引文: P1 p2-p3

怎样做	<ul style="list-style-type: none"> 1) 默认先用 $L_{simple} = E \ \varepsilon - \varepsilon_\theta\ ^2$ 把“样本质量基线”跑通(最少变量)。 2) 明确你的额外目标是否包含: 学习方差/协方差, 或关注对数似然/NLL。只有这些目标出现时才加入 VLB 相关项。 3) 把训练目标写成可切换配置: $loss = L_{simple}$ 或 $loss = L_{hybrid}$, 并在固定采样协议下比较(见第 5 章)。
为何这样做	P1 总结: 直接优化 VLB 往往样本质量差; L_{simple} 作为重加权形式实践更好, 但它不训练方差。
常见问题	只盯 L_{simple} 会让“损失很低但采样仍差”的情况难以解释, 尤其当采样器或方差设定不匹配时。
解决方式	<ul style="list-style-type: none"> 1) 当采样质量差但 L_{simple} 已很低: 优先检查采样链路(DDIM/Heun/随机性)是否与训练设定匹配(见 5.1-5.4)。 2) 当目标包含方差/似然: 用 $L_{hybrid} +$ 时间步重采样稳定 VLB(见 3.2-3.3); 或直接走 OCM 路线改协方差(见 3.7)。

3.2 `L_hybrid = L_simple + lambda * L_vlb` 用于学习方差

引文: P1 p3

怎样做	<ul style="list-style-type: none"> 1) 实现 L_{vlb} 并在日志里单独记录其数值(不要和 L_{simple} 混在一起看)。 2) 启用混合目标: $L_{hybrid} = L_{simple} + \lambda * L_{vlb}$, 从 P1 的起点 $\lambda = 0.001$ 开始。 3) 配套方差参数化: 采用 2.8 的“有界插值”让 Σ_θ 始终在合理范围。
为何这样做	P1 明确指出 L_{simple} 不给 Σ_θ 信号; 混合目标能同时保持样本质量与更好似然/方差建模。
常见问题	直接加 VLB 会让梯度噪声/训练不稳定, 甚至让 VLB 项“淹没” L_{simple} 。
解决方式	<ul style="list-style-type: none"> 1) 若训练抖动: 先确认 λ 足够小且 L_{vlb} 没有数值爆炸; 再启用时间步重采样(见 3.3)。 2) 若方差头不收敛: 确认你确实在训练 L_{hybrid} (仅 L_{simple} 不会给方差信号)。

3.3 用重要性采样/重采样降低 VLB 项的方差

引文: P1 p4

怎样做	1) 当启用 $L_{\text{vlb}} / L_{\text{hybrid}}$ 时, 把“时间步采样分布”从均匀改为可估计/可重采样的分布。 2) 统计每个 t 的 VLB 项(均值/方差或代理量), 据此分配更高采样概率给高方差/高影响的时间步。 3) 把该采样策略写入配置并固定, 否则不同实验间 VLB 抖动不可比。
为何这样做	P1 观察到 VLB 各时间步项的贡献差异大, 均匀采样会引入不必要噪声; 重要性采样能显著降低目标方差并改进优化。
常见问题	VLB 曲线很抖, 训练看起来“收敛不了”或需要更小学习率; 相同算力下效果不稳定。
解决方式	1) 先把 timestep 重采样开关接入训练管线, 并把“重采样分布的估计口径”写进配置(避免每次训练都不一致)。 2) 若 VLB 仍抖: 降低 λ 、增大 batch 或先延后启用 VLB 项(先用 L_{simple} 预热), 直到主干稳定再开启。

3.4 EDM: 噪声分布 `p_train(sigma)` 与 loss weighting `lambda(sigma)`

引文: P3 p45

怎样做	直接可抄的“配置组”。 1) 训练噪声分布(把默认常量写死, 读者直接复现即可): - $\ln(\sigma) \sim N(Pmean, Pstd^2)$, P3 的默认为 $Pmean = -1.2$, $Pstd = 1.2$ - $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$ - $\sigma_{\text{data}} = 0.5$ 2) loss weighting(把公式写死): - $\lambda(\sigma) = (\sigma^2 + \sigma_{\text{data}}^2) / (\sigma * \sigma_{\text{data}})^2$ 3) 与预条件联动: $p_{\text{train}}(\sigma)$ 、 $\lambda(\sigma)$ 、预条件(2.4)必须一起启用并一起消融; 不要只改其中一个旋钮。
为何这样做	P3 强调不同 sigma 的“去噪难度/误差结构”不同; 不合理的分布或权重会让模型把容量浪费在无效区间或忽略关键区间。
常见问题	生成结果出现系统性偏色/过饱和 drift; 或只在某些噪声级别下好, 但整体采样链路不稳。
解决方式	1) 固定网络与优化超参不变, 先只扫 $p_{\text{train}}(\sigma)$ 与 $\lambda(\sigma)$ 的配置组, 避免把收益误归因到别处。 2) 每组配置都用确定性采样器 + 固定 NFE 评测 FID/NLL, 并保存一张固定 x_T 的样本网格用于质检(见 5.2/5.6)。

3.5 DyDiT: FLOPs 约束损失与扩散主损失的联合

引文: P5 p5-p6

怎样做	1) 定义静态基线 FLOPs: 测量 F_{static} (固定结构一次前向)。 2) 统计动态 FLOPs: 训练时按 batch 统计 F_{dynamic} (实际执行子图的 FLOPs)。 3) 设定目标预算比 λ : 你希望 $\text{avg}(F_{\text{dynamic}} / F_{\text{static}})$ 接近多少。 4) 联合训练: $L = L_{\text{DiT}} + (\text{avg}(F_{\text{dynamic}} / F_{\text{static}}) - \lambda)^2$, 并在日志里单独记录预算项是否达标。
为何这样做	动态路由会自然倾向“多算力以降损失”; FLOPs 约束项把训练目标显式绑定到推理预算, 让效率-质量 tradeoff 可控。
常见问题	只用扩散损失训练动态结构会导致推理 FLOPs 不可控; 或者为降 FLOPs 牺牲关键模块(质量崩)。
解决方式	1) 如果预算达标但质量掉: 优先减少动态化范围(先只做 MLP token bypass, 保留 MHSA 交互)(P5)。 2) 如果质量达标但预算不达标: 调整目标比 λ 或提高 FLOPs 约束项权重, 并检查统计 F_{dynamic} 的口径是否一致。

3.6 “少步采样质量差”常常是方差/协方差选择问题

引文: P6 p1-p3

怎样做	1) 把少步目标写成默认评测面板: 5/10/20 步(NFE)三档固定比较。 2) 在该面板下对比协方差策略: 启发式 vs 学习方差(P1) vs OCM(P6)。 3) 同时记录 FID 与 NLL/recall, 避免只看单指标误判(见 5.5/5.8)。
为何这样做	P6 直接指出: 当 T 小时, 协方差选择影响很大; 更准确的对角协方差能显著改善少步采样的质量、recall 与似然。
常见问题	用启发式方差在 5/10/20 步下 FID 极差或生成发散; 但把步数加回去就“看起来好了”。
解决方式	1) 先换少步更稳的过程: DDIM(eta=0)或 EDM/Heun, 把“少步仍能成图”的基本盘打稳(见 5.1/5.3)。 2) 再引入协方差学习: 用 OCM 或学习方差来提升少步质量/似然(见 3.2/3.7)。

3.7 OCM: 用无偏目标回归最优对角协方差(对角 Hessian)

引文: P6 p3-p4

怎样做	<ul style="list-style-type: none"> 1) 明确直接估计的代价: P6 指出 CIFAR-10 上要获得理想对角近似需要 $M >= 100$ 个 Rademacher 样本; 且每个样本需要一次前向+一次反传, 总成本约 $2M$ 次网络评估, 推理不可用。 2) 增加一个对角 Hessian/协方差预测头 $h_\phi(\tilde{x})$, 与 score/均值预测并行。 3) 用无偏 OCM 目标训练该头: 用 $v \sim \text{Rademacher}$ 构造 $v \odot H(\tilde{x}) v$ 的蒙特卡洛项, 让 h_ϕ 去匹配其期望(摊销估计)。 4) 采样时用 h_ϕ 输出构造对角协方差, 在 5/10/20 步面板下评测 FID/NLL 是否改善(见 5.8)。
为何这样做	该目标避免“小 M 直接回归均值”带来的偏置, 同时推理时只需一次网络前向即可得到对角项。
常见问题	直接用 Rademacher 估计要 $M >= 100$ 才有可接受近似, 会把采样速度拖到不可用; 或者小 M 估计偏差大导致协方差更糟。
解决方式	<ul style="list-style-type: none"> 1) 训练期: 先把 OCM 头训练稳定(监控其对角项误差或相关代理指标), 再把它接入采样链路。 2) 评测期: 固定少步面板 5/10/20 步, 同时报 FID 与 NLL, 只要在少步 regime 稳定改善才算“值回训练复杂度”(见 5.8)。

4) 优化器选择

4.0 本章上手例子: Adam + EMA 的“最小训练协议”(你应该固定什么)

引文: P1 p11; P3 p45

怎样做	<p>把它当作训练脚本的 checklist。</p> <ul style="list-style-type: none"> 1) 优化器与关键超参固定为起点: Adam, batch = 128, lr = $1e-4$, EMA = 0.9999(P1)。 2) 每一步更新都同时更新 EMA 权重: $ema = ema * decay + params * (1-decay)$。 3) 评测/采样只用 EMA 权重: 任何 FID/NLL 与可视化网格都必须写明“用的是 EMA 权重”还是“训练权重”。 4) 训练日志至少包含: loss、学习率、EMA decay/half-life、每次 snapshot 的评测结果(见 5.6)。
为何这样做	P1/P3 都把 EMA 当作默认训练组件; 不固定“是否用 EMA”会导致同样 loss 下的采样质量不可比。
常见问题	训练看似稳定但采样很飘, 往往是因为评测时混用了权重(有时用训练权重, 有时用 EMA)或 EMA 衰减设置不一致。
解决方式	<ul style="list-style-type: none"> 1) 先统一评测用 EMA 权重, 再讨论是否需要调 EMA 强度(见 4.1 的网格)。 2) 把 EMA 的配置写进 config 与日志, 不允许“靠记忆”复现。

4.1 Adam + EMA 是论文体系内的默认起点

引文: P1 p11; P3 p45

怎样做	<ul style="list-style-type: none"> 1) 优化器用 Adam 作为默认起点。 2) 从训练第一步就维护 EMA 权重, 并明确“评测/采样用 EMA 权重”(P1/P3 都把 EMA 当作默认组件)。 3) 以 P1 的起点超参开跑: $lr = 1e-4$, $batch = 128$, $EMA = 0.9999$; 若按 P3 表达, 把 EMA 写成 $half-life(M_{img})$ 并记录。 4) 日志里对比: 原始权重 vs EMA 权重在同一采样协议下的结果, 避免误判 EMA 是否有效。
为何这样做	EMA 能显著平滑训练噪声并提升采样稳定性; 两篇论文都把 EMA 当作关键训练组件并写入默认超参数表。
常见问题	不用 EMA 时, 同样 loss 下采样更差且不稳定; EMA 过强会让模型“跟不上”学习进度, 过弱又起不到平滑效果。
解决方式	<ul style="list-style-type: none"> 1) 固定采样协议后对 EMA 做小网格: EMA 值/half-life 只改一个维度, 观察采样稳定性与 FID/NLL 的变化。 2) 任何报告都同时写出“训练用权重”与“采样用 EMA 权重”, 避免复现实验时用错权重。

4.2 学习率/批量与模型规模联动(不应将超参视为常数)

引文: P1 p11; P3 p45

怎样做	<p>给出可直接复现的两套默认起点。</p> <p>1) 默认起点 A(P1, 作为大多数实验的基线): Adam + EMA, 并固定:</p> <ul style="list-style-type: none"> - batch=128 - lr=1e-4 - EMA=0.9999 <p>2) 吞吐导向起点 B(P3, CIFAR-10 的“Ours”训练表): 直接复现其训练协议(把每个数字写死, 不需要翻论文):</p> <ul style="list-style-type: none"> - 训练总量: 200 Mimg - GPU 数: 8 - batch: 512 - 学习率: 1e-3 - LR ramp-up: 10 Mimg - EMA half-life: 0.5 Mimg - 梯度裁剪: 关闭(P3 经验是关掉不影响且更省事) - Dropout: 13%(P3 在该表中给出其最终选择) - 数据集 x-flips: 关闭(配合 non-leaky augmentation 思路) - Augment probability: 12% <p>3) 当改变模型规模(例如把 C 变大)或改变并行度时, 不允许“只改模型不改超参”: 必须至少重新跑一次 (A 或 B) 的对照, 再继续细调。</p>
为何这样做	扩散训练的梯度噪声与有效 batch、目标项方差耦合明显; 固定 lr 往往会导致收敛变慢或不稳定。
常见问题	增大模型后 loss 下降变慢、震荡, 或者出现训练发散; 不同配置之间对比失真。
解决方式	<p>1) 扩大模型后先做“只改优化超参”的对照: 固定模型与采样协议, 扫 batch/lr/warmup/EMA 时间常数。</p> <p>2) 若启用 VLB/协方差学习: 同时启用时间步重采样或 OCM 等稳定化策略, 否则训练抖动会掩盖结构收益(见 3.3/3.7)。</p>

4.3 梯度裁剪与混合精度: 以“是否有收益”为准

引文: P3 p45-p46

怎样做	<p>1) 梯度裁剪: 严格按 P3 的结论落地成“默认关闭”:</p> <ul style="list-style-type: none"> - 把开关写进 config(例如 <code>grad_clip_norm: null</code> / <code>grad_clip_norm: 1.0</code>)。 - 固定其余一切(模型/损失/采样器/NFE/样本量)只做一次开/关对照; 若无稳定收益, 就沿用 P3 的默认: 关闭。 <p>2) 混合精度: 严格按 P3 的使用场景落地:</p> <ul style="list-style-type: none"> - 仅当大规模训练中需要吞吐(例如 P3 的 ImageNet-64 配置)才启用 FP16。 - 日志必须写明: 是否 FP16、是否有 loss scaling、以及是否出现 NaN/Inf(出现即回退 FP32 先跑通基线)。 <p>3) 对任何“稳定化手段”都用同一评测协议确认“真的提升了采样质量/稳定性”, 不要只看 loss(见 5.6 的确定性采样选点协议)。</p>
为何这样做	扩散训练稳定性来自多因素; “习惯性开裁剪”不一定有用; 而混合精度带来吞吐但会引入数值风险。
常见问题	盲目裁剪导致有效学习率被压低; FP16 在某些层出现数值不稳定导致训练崩或质量退化。
解决方式	<p>1) 先在 baseline 配置上把裁剪开关做消融, 只保留“确实提升稳定性/质量”的设置。</p> <p>2) 若混合精度下出现 NaN/发散: 先回退到 FP32 把训练-采样闭环跑通; 然后再按 P3 的思路把 FP16 只作为吞吐优化重新启用, 并再次用同一评测协议验证质量不退化。</p>

4.4 动态结构的端到端训练稳定性

引文: P5 p6

怎样做	<p>1) 路由器使用 STE/Gumbel-Sigmoid 等可微近似以训练离散决策(P5)。</p> <p>2) 若端到端 finetune 不稳: 先引入 warm-up, 训练初期保留完整模型监督项 $L_{complete}$。</p> <p>3) 当路由器不再塌陷且预算项达标后, 再逐步移除 $L_{complete}$, 进入纯联合目标训练(与 3.5 联动)。</p>
为何这样做	动态 mask 引入离散决策与额外优化难度; warm-up 让主干保持可靠梯度, 再逐步让路由器接管。
常见问题	训练出现不稳定震荡、路由器塌陷(全开或全关)、质量显著下降。
解决方式	<p>1) 先用 warm-up 把训练稳定下来, 再逐步移除 $L_{complete}$; 不要一开始就让路由器完全接管(P5)。</p> <p>2) 用 FLOPs 约束项保证推理预算达标, 并在固定采样协议下复核质量没有系统性掉点(见 3.5/5.6)。</p>

4.5 Dropout 与 EMA 需要按 schedule/数据集做小网格

引文: P1 p11; P3 p45

怎样做	<p>1) Dropout 的“起步网格”直接复现 P1 的 CIFAR-10 网格: {0.1, 0.2, 0.3}(每次只改 dropout, 其余不动)。</p> <p>2) Dropout 的“单点起步”也可直接复现 P3 Table 7 的最终值(预算不足时先用它们跑通闭环):</p> <ul style="list-style-type: none"> - CIFAR-10: 13%(P3 的 improved 配置) - FFHQ: 5%, AFHQv2: 25%(P3) - ImageNet-64: 10%(P3) <p>3) EMA 的“单点起步”也直接复现 P3 Table 7 的 half-life(用 half-life 写法更不易和 batch/lr 混淆):</p> <ul style="list-style-type: none"> - CIFAR-10/FFHQ/AFHQv2(改进配置): 0.5 Mimg - ImageNet-64: 50 Mimg <p>4) 对每种 noise schedule(Linear/Cosine)分别做 dropout 网格(P1 指出“最优 dropout 随 schedule 变”)。</p> <p>5) dropout/EMA 的比较必须在固定采样协议下完成(确定性采样器 + 固定 NFE + 固定样本量)(见 5.6)。</p>
为何这样做	扩散训练的有效正则强度与噪声调度、增强、采样器选择都耦合; dropout 与 EMA 决定了“稳定性-拟合度”的平衡点。
常见问题	dropout 太小导致过拟合或样本多样性下降; 太大导致欠拟合、细节糊; EMA 设置不当使得采样权重落后或波动大。
解决方式	<p>1) 把 dropout 与 EMA 同时纳入小网格, 但每次只改少量组合以控制实验成本。</p> <p>2) 选择标准以固定采样协议下的 FID/NLL/样本多样性为主, 不要只看训练 loss。</p>

5) 可视化与微调(把“采样链路”纳入调参闭环)

5.0 本章上手例子: 每个 epoch 生成可比对的可视化(固定 `x_T` + 固定采样器)

引文: P2 p7-p8; P3 p45

怎样做	<p>把“可视化”变成可复现的评测协议, 而不是随意抽取少量图像。</p> <ol style="list-style-type: none"> 固定一组初始 x_T(例如保存一个 shape 为 [64, 3, H, W] 的噪声张量, 或固定随机种子)。 固定采样器作为可视化基线: <ul style="list-style-type: none"> - 少步: 用 DDIM, $\eta = 0$ (确定性) - 固定步数: 本手册默认固定 $N = 50$ (全程不变; 若要更稳定可再加一条 $N = 100$ 的对照面板) 每个 epoch 结束都用“同一个 x_T”生成同一张 8x8 网格图并保存, 这样你看到的变化只来自模型权重变化(P2 的一致性诊断思想)。 若调 EDM 的数值积分: 把求解器固定为 Heun + 固定 NFE, 否则可视化会被采样噪声污染(P3)。
为何这样做	P2 的一致性诊断强调 x_T 是信息性 latent code; 只有固定 x_T , 可视化才能反映“模型是否真的学到了更好的反向映射”而不是随机波动。
常见问题	每次可视化都重新采样 x_T , 导致你无法判断质量是变好了还是只是抽到了更好的噪声种子。
解决方式	<ol style="list-style-type: none"> 强制固定 x_T 与采样器, 把它写进代码为默认。 若要展示多样性, 另存一组“多 seed 网格”, 但不要替代固定 x_T 的主面板。

5.1 DDIM: 训练不变, 只改采样即可获得少步高质量

引文: P2 p6-p8

怎样做	<ol style="list-style-type: none"> 保持训练不变: $T = 1000$ 与训练目标不动。 采样改为 DDIM: 选择子序列 τ, 按 P2 的两种规则二选一(不要自己发明第三种, 先复现论文): <ul style="list-style-type: none"> - Linear: $\tau_i = \text{floor}(c * i)$, 选 c 使得 τ_{N-1} 接近 T。 - Quadratic: $\tau_i = \text{floor}(c * i^2)$, 选 c 使得 τ_{N-1} 接近 T。 P2 的经验选择: CIFAR-10 用 quadratic, 其余数据集用 linear。 固定步数 sweep(默认面板): $N \in \{10, 20, 50, 100\}$。 先固定 $\eta = 0$ 作为确定性基线; 再尝试 $\eta > 0$ 看是否改善多样性。 按 P2 的闭式更新式实现每一步 (见 0.3 的公式); 实现后先用 $\eta = 0$ 复核“同一 x_T 下不同 N 的高层语义一致性”(见 5.2)。 每档步数都在同一评测协议下对比: 固定采样器、固定样本量、固定是否使用 EMA 权重(见 5.5-5.6)。
为何这样做	P2 证明在少采样步下, DDIM 往往显著优于 DDPM; 并指出 DDIM 在较短轨迹也能接近 1000 步质量, 带来 10x-50x 加速。
常见问题	用 DDPM 的高噪声方差策略(如更大的 σ_{hat})在短轨迹下会产生明显噪声扰动, FID 急剧变差。
解决方式	<ol style="list-style-type: none"> 先用 $\eta = 0$ 跑通少步基线(50/100 步), 确认实现正确且曲线单调合理。 再在固定 x_T 与固定评测协议下扫 η, 观察多样性与质量是否存在稳定 tradeoff(见 5.2/5.5)。

5.2 DDIM 的一致性诊断: 固定同一个 `x_T` 看不同轨迹

引文: P2 p7-p8

怎样做	1) 固定一组初始 latent x_T (固定随机种子或直接缓存张量)。 2) 对同一 x_T 分别用不同 τ (步数/时间点分布不同)采样。 3) 做一致性检查: 高层语义(主体/构图)应基本一致, 细节只随步数变化。 4) 若一致性很差: 把它当作采样实现/数值误差/调度不匹配的红旗, 先查实现再调参。
为何这样做	P2 观察到 DDIM 在固定 x_T 下, 不同轨迹生成的高层特征相似, 暗示 x_T 是信息性 latent code; 这为调参提供“结构性信号”。
常见问题	如果不同轨迹下高层语义完全不一致, 说明采样数值误差/模型误差在放大, 或噪声调度/采样实现存在 bug。
解决方式	1) 先切到 $\eta = 0$ 并使用更平滑/更密集的 τ 子序列, 排除随机性与过粗步长的影响。 2) 若仍不一致: 优先排查 $\bar{\alpha}_t$ 取值、均值/方差项实现与广播维度是否正确(这是最常见的“看起来像噪声”原因之一)。

5.3 EDM: 2 阶 ODE 求解器(Heun)往往比 Euler 更划算

引文: P3 p45

怎样做	1) 把采样器实现抽象成“给定噪声序列 $\{\sigma_i\}$ 的数值积分器”, 并按 P3 写死默认时间步离散化(EDM 公式): <ul style="list-style-type: none"> - 设采样步数为 N, P3 使用 ρ-warp 的 σ 序列: $\sigma_i = (\sigma_{\max}^{(1/\rho)} + i / (N-1) * (\sigma_{\min}^{(1/\rho)} - \sigma_{\max}^{(1/\rho)}))^{1/\rho}$, 其中 $i = 0..N-1$, 并设 $\sigma_N = 0$。 - P3 的默认边界选择: $\sigma_{\min} = \max(\sigma_{lo}, 0.002)$, $\sigma_{\max} = \min(\sigma_{hi}, 80)$; 常用默认常量为 $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$。 2) 在同样步数 N 下把 Euler 替换为 Heun(2 阶, 梯形法)。 <ul style="list-style-type: none"> - 代价口径写死: P3 指出 Heun 每步多一次 D_θ 评估; 采用 Heun 时总 NFE 约为 $2N-1$ (最后一步到 $\sigma = 0$ 无需二次校正)。 3) 只改求解器不改其余设置, 在固定评测协议下画出 NFE-质量曲线比较。
为何这样做	P3 通过大量实验展示 Heun 在相同 FID 下所需 NFE 更少, 是更好的“质量-计算”折中。
常见问题	Euler 在某些噪声段局部误差大, 需要很多步才能弥补; 少步时偏离轨迹导致图像更像噪声或细节破碎。
解决方式	1) 把 Heun 固定为默认确定性采样器, 先在同一时间点序列下替换 Euler 验证收益。 <ul style="list-style-type: none"> 2) 若收益有限: 再调整离散时间点/步分布, 并用固定 NFE 对齐比较(按 P3 的消融方式做)。

5.4 EDM: 随机性不是“越多越好”，需要按数据集经验性搜索

引文: P3 p45

怎样做	<p>直接可用的网格来自 P3 的随机采样消融表。</p> <ol style="list-style-type: none"> 1) 把随机性写成参数组: S_{churn}, S_{tmin}, S_{tmax}, S_{noise}, 并确保可以网格搜索。 2) 以确定性采样作为基线: $S_{\text{churn}} = 0$ (P3 视作确定性采样器基线)。 3) 按 P3 的网格集合做起点: <ul style="list-style-type: none"> - S_{churn}: 0, 10, 20, ..., 100 - S_{tmin}: 0, 0.005, 0.01, 0.02, ..., 1, 2, 5, 10 - S_{tmax}: 0.2, 0.5, 1, 2, ..., 10, 20, 50, 80 - S_{noise}: 1.000, 1.001, ..., 1.010 4) 若你需要一个“单点起步”再微调: 直接复现 P3 Table 5 的已用参数(你后续再在网格内做局部搜索): <ul style="list-style-type: none"> - CIFAR-10 (VP/VE): $S_{\text{churn}} = 30$, $S_{\text{tmin}} = 0.01$, $S_{\text{tmax}} = 1$, $S_{\text{noise}} = 1.007$ - ImageNet-64 (P3 用于 pre-trained/ours 的设置): $S_{\text{tmin}} = 0.05$, $S_{\text{tmax}} = 50$, $S_{\text{noise}} = 1.003$; S_{churn} 在不同设定中取 80 或 40 (以复现目标表为准, 但必须把这一点写进实验日志)。 5) 固定 NFE 后再搜随机性(避免归因混乱): 对同一 N(或同一 NFE)只改随机性参数, 选最优点; 不允许同时改时间点离散化/求解器/随机性三者。
为何这样做	P3 指出随机性有助于纠正采样误差, 但也会引入图像退化(例如过饱和 drift); 最优随机性取决于数据集与训练设置, 必须经验性确定。
常见问题	过强随机性导致颜色漂移/过饱和、细节抖动; 或者在某些训练设置下随机性反而伤害 FID。
解决方式	<ol style="list-style-type: none"> 1) 先固定 $S_{\text{churn}} = 0$ 作为对齐基线, 再逐个打开随机性旋钮(避免一次性改太多看不出因果)。 2) 若出现过饱和/漂移: 优先限制 $S_{\text{tmin}} / S_{\text{tmax}}$ 的生效区间, 避免在极低噪声段引入随机性(P3 的启发式动机)。

5.5 评测指标与样本量: FID/NLL/Precision-Recall 的取舍

引文: P1 p1,p11; P6 p3; P4 p37

怎样做	<ul style="list-style-type: none"> 1) 把“评测样本量 + 采样器 + NFE”写成协议并固定(否则不同实验间 FID/NLL 不可比)。 2) 把 FID 的计算口径写死(避免“同名不同口径”): <ul style="list-style-type: none"> - 样本量: 50,000 张生成图像 vs 全部真实图像(P3 在实现细节中明确其默认口径; P1 也以 50K 作为常见基准)。 - 真实图像侧不做增强: 不使用 x-flips 等增强再算 FID(P3 明确说明)。 - 为降低随机波动: P3 报告其 FID 通常会重复计算 3 次并报告最小值, 并指出随机波动量级约为 $\pm 2\%$。 - 快速消融时的降采样: P1 在无条件 ImageNet-64 的大消融中会用 10K 样本加速(承认会带来偏差但用于相对比较); 若你采用 10K, 必须在图注/表注里显式写出并在关键结论处用 50K 复核。 3) 把指标面板固定为多指标: FID(质量), NLL/bits-dim(似然), improved precision/recall(覆盖)同时记录, 避免单指标误判(P1/P4/P6)。
为何这样做	P1 指出仅看 FID 不足以判断覆盖, precision/recall 能揭示扩散模型的高 recall 特性; P4 也强调评价基准仍不完备且易受偏差影响。
常见问题	用 10K 样本算 FID 会带偏差, 但常用于快速消融; 只看 loss 会误判, 因为 loss 与采样器/方差策略耦合。
解决方式	<ul style="list-style-type: none"> 1) 建立“两级评测”: 快速 ablation 用较少样本但固定采样器与固定 NFE; 关键结论用 50K 样本复核(P1)。 2) 每次只允许一个自由度变化(模型/损失/采样器三者择一), 否则 FID 变化无法归因。

5.6 “保存快照 + 用确定性采样器选最优”作为报告协议

引文: P3 p45

怎样做	<ul style="list-style-type: none"> 1) 把 checkpoint 频率与“选最优”的规则写死(写进实验协议, 不要临时主观设定)。 2) 按 P3 的做法: 按训练图像数保存 snapshot(例如每 2.5 Mimg 一次)。 3) 用固定的确定性采样器与固定 NFE 评测 FID, 取最低 FID 的 snapshot 报告: <ul style="list-style-type: none"> - P3 的实现细节里明确: 其报告通常基于“确定性采样器 + 固定 NFE”(例如 NFE=35 或 NFE=79, 取决于分辨率)来选最优点。 - 你不必跟它用同一个 NFE, 但必须固定并在日志/报告里写明。 4) 报告时同步写出: 采样器(确定性/随机性)、NFE/步数、样本量、以及选择规则(最低 FID/其他指标)。
为何这样做	P3 用该协议降低“评测随机性与采样器差异”对结论的污染, 让训练过程的对比更像可复现的数值实验而不是抽样展示。
常见问题	只看训练 loss 选 checkpoint, 可能选到采样质量并不最优的点; 或者随机采样器让评测波动大。
解决方式	<ul style="list-style-type: none"> 1) 用确定性采样器锁死评测链路, 避免随机采样器造成“选点噪声”。 2) checkpoint 选择依赖固定 NFE 下的 FID/质量曲线, 而不是单一训练 loss(P3)。

5.7 少步推理加速: 协方差与少步过程要一起改

引文: P6 p1-p4; P2 p6-p8

怎样做	1) 先定少步目标: 5/10/20 步三档固定为默认评测面板。 2) 过程侧先改采样: DDIM(确定性优先) 或 EDM/Heun, 把步数压下来后仍可稳定成图(见 5.1/5.3)。 3) 统计侧再改协方差: 若少步质量/似然仍差, 引入学习方差(P1)或 OCM(P6)(见 3.2/3.7)。 4) 指标侧必须联动: 同时报 FID 与 NLL/recall, 避免只看 FID 忽略覆盖(见 5.5/5.8)。
为何这样做	P6 论证协方差在小步数时关键; P2 论证少步下换成 DDIM 立刻改善; 组合起来才能既快又不塌质量/覆盖。
常见问题	只做 DDIM 但方差/协方差仍用启发式, 少步质量仍差; 或只学协方差但采样过程仍不匹配, 收益有限。
解决方式	1) 把“少步目标”作为系统目标写进实验协议: 同一套 5/10/20 步面板反复复核。 2) 按 P2/P3/P6 分别对采样轨迹(DDIM)、数值积分(Heun/时间点)、协方差(OCM/学习方差)做联动消融, 并用统一可视化面板记录“步数-质量曲线”和“覆盖/似然曲线”。

5.8 OCM 的可视化目标: 少步场景直接看(步数, FID, NLL)三元组

引文: P6 p3

怎样做	1) 固定同一数据集与模型(保持权重不变), 只改协方差策略(启发式 vs OCM)。 2) 固定少步步数集合: 5/10/15/20 步(或你的目标 NFE 集合)。 3) 每个步数下同时报 FID 与 NLL, 并附上同一批固定 x_T 的样本网格(见 5.2)。 4) 用“(步数, FID, NLL) 三元组曲线”判断 OCM 在少步 regime 的真实收益(P6)。
为何这样做	P6 的主张是“协方差对少步至关重要”; 因此可视化必须直接落在少步 regime, 否则会掩盖协方差改进的核心收益。
常见问题	只在 1000 步下评测, 看不出协方差学习的价值; 或只看 FID 看不出似然/覆盖的变化。
解决方式	1) 把少步评测面板固定为默认: 步数扫一遍, 同时记录 FID 与 NLL。 2) 只有当 OCM 在少步面板上稳定带来收益, 才把协方差头与其训练复杂度纳入长期默认配置(P6)。

附录) 快速查表(只收录 P1-P6 中“可以直接复现”的表与数字)

A.0 三类数据集的“指标→模型”决策流程 (图像/视频/SMPL-H)

引文: P4 p37; P1 p11; P3 p45; P5 p17; P6 p3

怎样做	<p>三类数据集统一使用同一模板: 先统计指标, 再按规则选主干与规模档位。</p> <ol style="list-style-type: none"> 1) 图像数据集(文本条件)必统计: <ul style="list-style-type: none"> - 规模: 样本数 N、分辨率分布、长尾类别/场景覆盖。 - 质量: 坏样本率、重复/近重复率、压缩噪声/水印比例。 - 对齐: 文本-图像匹配分数分布与低分样本占比。 - 决策规则: 小规模或高噪声优先 U-Net 路线(P1/P3); 大规模且高质量可进入 DiT/Transformer 路线(P5)。 2) 视频数据集(文本条件)必统计: <ul style="list-style-type: none"> - 规模: 视频条数、总时长、分辨率/FPS/时长分布。 - 质量: 坏片段率、压缩噪声、重复片段率、镜头切分异常比例。 - 时空复杂度: 运动强度、镜头切换频率、长时依赖比例(长视频占比)。 - 对齐: 文本-视频匹配分数分布与时序语义错配比例。 - 决策规则: 小规模或噪声高先用 Video U-Net/latent video diffusion; 大规模高质量且运动复杂可用 Video DiT/时空 Transformer。 3) SMPL-H 序列数据集(文本条件)必统计: <ul style="list-style-type: none"> - 规模: 序列条数、总帧数、序列长度分布。 - 质量: 参数缺失率、越界率、重建失败率(不可渲染/异常姿态)。 - 动态复杂度: 关节速度/加速度分布、手部高频动作比例、长程动作依赖比例。 - 对齐: 文本-动作匹配分数分布、动作语义错配比例。 - 决策规则: 数据规模有限或噪声偏高优先 1D U-Net/参数扩散; 规模更大且动作语义复杂可用序列 Transformer 扩散。 4) 三类数据统一选型门槛: <ul style="list-style-type: none"> - 仅统计“规模+质量”不足以选型, 还必须加入“复杂度(时空/动力学)”与“文本对齐”。 - 主干比较时冻结采样器/NFE/样本量/随机种子, 只改主干与离散规模档位(见 5.5-5.6)。 - 先做最小两档对照确认趋势, 再扩展规模扫点(P1/P5)。
为何这样做	数据规模与质量决定模型是否可训练, 复杂度决定需要的建模能力, 文本对齐决定条件学习上限; 四者缺一都会导致“模型选错但不可归因”。该流程把三类数据集放在同一可复核框架下, 并与扩散论文中的可复现评测协议对齐(P1/P3/P4/P5/P6)。
常见问题	只看规模与质量就上大模型; 视频任务忽略时空复杂度; SMPL-H 任务忽略参数有效性; 条件任务忽略文本对齐噪声, 结果训练不稳定且结论不可信。
解决方式	<ol style="list-style-type: none"> 1) 先完成每类数据的四组指标统计(规模/质量/复杂度/对齐), 再进入模型选型。 2) 先按规则给出每类数据的“保守基线主干”, 通过固定协议对照验证后再升阶到更大模型。

A.1 DiT/DyDiT 的规模档位表(选 layers/heads/width)

引文: P5 p17 (Table 7)

怎样做	<p>选型原则: 让读者知道什么时候用 S/B/XL, 并能在同一协议下复现对照。</p> <ol style="list-style-type: none"> 将 P1 的 U-Net 替换为 Transformer(DiT)时, 先用下面的“离散档位表”锁死 layers / heads / channel, 保证可复现与可对照(P5)。 做 DyDiT 动态化时, 仍以同一档位的静态 DiT 为主干; 动态化只改变每步激活的 heads/通道组/token, 不改变主干规模定义(P5)。 先确认你是否处在 P5 的“已覆盖分辨率/任务”范围: <ul style="list-style-type: none"> - P5 的主要实验在 ImageNet 256x256, 并额外报告了 512x512(同样以 DiT-XL 为基线模型)(P5)。 - 若你的目标分辨率不是 256 或 512: 本手册不声称哪档“最优”; 你仍可用 S/B/XL 作为离散 sweep, 但结论属于你的新实验, 需要按第 5 章固定协议对齐评测再下结论。 默认决策(严格对齐 P4/P5 的“数据稀缺/预算/对照”语境, 不引入论文外阈值): <ul style="list-style-type: none"> - ImageNet-256/512 且追求最优质量: 选 DiT-XL (P5 报告的最强静态基线规模)。 - ImageNet-256/512 但更在意快速迭代/先跑通闭环: 选 DiT-S; 跑通后再升到 DiT-B/DiT-XL 做对照(P5)。 - fine-tuning/迁移/数据更稀缺: 先从 DiT-S 起步; P4 强调稀缺任务更易过拟合/偏置, P5 也在附录微调语境中以小模型作为常用起点; 用 5.6 的 snapshot 协议确认不过拟合后再升档。 结构档位表(直接复现, 来自 P5 Table 7; 用途: 只要你声明“用的是哪个档位”, 读者就知道你的 depth/heads/width): <table border="1"> <thead> <tr> <th>model</th><th>params (M)</th><th>layers</th><th>heads</th><th>channel (width)</th><th>pre-training source</th></tr> </thead> <tbody> <tr> <td>DiT-S</td><td>33</td><td>12</td><td>6</td><td>384</td><td>5M iter (Pan et al., 2024)</td></tr> <tr> <td>DiT-B</td><td>130</td><td>12</td><td>12</td><td>768</td><td>1.6M iter (Pan et al., 2024)</td></tr> <tr> <td>DiT-XL</td><td>675</td><td>28</td><td>16</td><td>1152</td><td>7M iter (Peebles & Xie, 2023)</td></tr> </tbody> </table> <p>6) “控制变量”的更全表格见下面 A.1a 总表: 固定模型与分辨率, 只扫 λ, 给出同表对照的 FLOPs 与 FID(P5 Table 8)。</p>	model	params (M)	layers	heads	channel (width)	pre-training source	DiT-S	33	12	6	384	5M iter (Pan et al., 2024)	DiT-B	130	12	12	768	1.6M iter (Pan et al., 2024)	DiT-XL	675	28	16	1152	7M iter (Peebles & Xie, 2023)
model	params (M)	layers	heads	channel (width)	pre-training source																				
DiT-S	33	12	6	384	5M iter (Pan et al., 2024)																				
DiT-B	130	12	12	768	1.6M iter (Pan et al., 2024)																				
DiT-XL	675	28	16	1152	7M iter (Peebles & Xie, 2023)																				
为何这样做	P5 的方法是在既有 DiT 档位上做动态计算; 因此如果不沿用这些档位, 结论无法和论文对齐。																								
常见问题	把“分辨率/patch/token 设计”当成 layers/heads 的推导依据, 结果越改越乱。																								
解决方式	先锁死档位(选 S/B/XL), 分辨率与 token/patch 作为单独设计变量在独立实验中改, 不要混进“模型规模定义”里。																								

A.1a 控制变量总表: 固定 ImageNet-256 与评测链路, 按 backbone 扫目标 FLOPs 比例 `λ`

引文: P5 p18 (Table 8)

怎样做	1) 这是把原 A.1a/A.1b/A.1c 合并后的“单一入口总表”: 固定分辨率=256x256、固定评测=FID, 并在 DiT-S / DiT-B / DiT-XL 三档 backbone 下分别扫 λ , 统一看速度-质量 tradeoff(P5)。 2) 写论文/报告时, 可直接引用这张总表进行横向比较: 同一 backbone 内比较 λ 的影响, 不同 backbone 间比较“规模-效率-质量”关系。 表(P5 Table 8; 注: P5 说明其在 V100 32G 上用每个模型的最优 batch 做 batched inference; DyDiT 实际 FLOPs 会在目标 λ 附近波动):						
	model	λ	s/image (↓)	acceleration (↑)	FLOPs (G) (↓)	FID (↓)	FID Δ (↓)
	DiT-S (static)	1.0	0.65	1.00	6.07	21.46	+0.00
	DyDiT-S	0.9	0.63	1.03	5.72	21.06	-0.40
	DyDiT-S	0.8	0.56	1.16	4.94	21.95	+0.49
	DyDiT-S	0.7	0.51	1.27	4.34	23.01	+1.55
	DyDiT-S	0.5	0.42	1.54	3.16	28.75	+7.29
	DyDiT-S	0.4	0.38	1.71	2.63	36.21	+14.75
	DyDiT-S	0.3	0.32	2.03	1.96	59.28	+37.83
	DiT-B (static)	1.0	2.09	1.00	23.02	9.07	+0.00
	DyDiT-B	0.9	1.97	1.05	21.28	8.78	-0.29
	DyDiT-B	0.8	1.76	1.18	18.53	8.79	-0.28
	DyDiT-B	0.7	1.57	1.32	16.28	9.40	+0.33
	DyDiT-B	0.5	1.22	1.70	11.90	12.92	+3.85
	DyDiT-B	0.4	1.06	1.95	9.71	15.54	+6.47
	DyDiT-B	0.3	0.89	2.33	7.51	23.34	+14.27
	DiT-XL (static)	1.0	10.22	1.00	118.69	2.27	+0.00
	DyDiT-XL	0.7	7.76	1.32	84.33	2.12	-0.15
	DyDiT-XL	0.6	6.86	1.49	67.83	2.18	-0.09
	DyDiT-XL	0.5	5.91	1.73	57.88	2.07	-0.20
	DyDiT-XL	0.3	4.26	2.40	38.85	3.36	+1.09
为何这样做	P5 的核心目标是“用动态计算把 FLOPs 压下去但尽量不伤质量”; 用一张总表集中展示更便于做同协议下的横向比较。						
常见问题	同时改 sampler/CFG/步数/NFE 与 λ , 导致你分不清质量变化是动态化带来的还是采样链路带来的。						
解决方式	先按 P5 的对照方式固定采样与评测链路, 只扫 λ ; 采样链路的改动另开一组实验(见第 5 章)。						

A.2 DyDiT 的训练细节表(优化器/批量/迭代/ λ 网格)

引文: P5 p17 (Table 6)

怎样做	<p>1) 当静态 DiT 上加路由器做 DyDiT 微调时, 用这张表直接给出“微调训练协议”的默认起点(优化器与 lr, global batch, warmup, λ 网格)。</p> <p>表(P5 Table 6, ImageNet 微调设置):</p> <ul style="list-style-type: none"> - optimizer: AdamW, learning rate=1e-4 - global batch size: 256 - target FLOPs ratio λ: <ul style="list-style-type: none"> - DiT-S/DiT-B: {0.9, 0.8, 0.7, 0.5, 0.4, 0.3} - DiT-XL: {0.7, 0.6, 0.5, 0.3} - fine-tuning iterations: <ul style="list-style-type: none"> - DiT-S: 50,000 - DiT-B: 100,000 - DiT-XL: 150,000 (for $\lambda=0.7$), 200,000 (for others) - warmup iterations: DiT-XL uses 30,000 (S/B are 0) - augmentation: random flip - cropping size: 224x224
为何这样做	P5 的实验与结论依赖“只做少量微调(约 3%)”即可启用动态架构; 沿用其微调协议能复现同一节奏与同一约束。
常见问题	把 λ 当成“随便取个值”而不是明确网格; 或者不做 warmup 导致 XL 微调不稳定。
解决方式	先按表跑通 λ 网格与 warmup, 再讨论更复杂的稳定化策略(见 4.4)。

A.3 EDM 随机采样参数网格(直接用来搜 S_churn 等)

引文: P3 p43 (Table 5)

怎样做	<p>1) 要做 EDM 随机采样时, 直接用这张表定义搜索空间, 避免自造范围导致不可比。</p> <p>表(P3 Table 5, grid search sets):</p> <ul style="list-style-type: none"> - S_churn: 0,10,20,...,100 - S_tmin: 0,0.005,0.01,0.02,...,1,2,5,10 - S_tmax: 0.2,0.5,1,2,...,10,20,50,80 - S_noise: 1.000,1.001,...,1.010
为何这样做	P3 指出最优随机性取决于数据集与设置; 但网格集合必须固定才能对照。
常见问题	一次同时改求解器/时间点/随机性, 导致 FID 变化无法归因。
解决方式	固定 NFE 与求解器(Heun/确定性)后再只搜随机性(见 5.3-5.4)。

A.4 EDM 训练超参数表(训练时长/批量/lr/ramp-up/EMA/dropout)

引文: P3 p45 (Table 7)

怎样做	<p>1) 复现 P3 的 improved 配置或以其为起点, 用这张表直接设置训练超参; 不需要翻论文。</p> <p>表(P3 Table 7, 关键字段摘录; 单位照论文口径):</p> <ul style="list-style-type: none"> - CIFAR-10 (Ours): duration=200 Mimg, GPUs=8, batch=512, lr=1e-3, LR ramp-up=10 Mimg, EMA half-life=0.5 Mimg, dropout=13%, dataset x-flips=off, augment prob=12% - ImageNet (Ours): duration=2500 Mimg, GPUs=32, batch=4096, lr=1e-4, LR ramp-up=10 Mimg, EMA half-life=50 Mimg, dropout=10%
为何这样做	P3 的结论强调“训练端配置(噪声分布/权重/超参)与采样端耦合”; 你不沿用这些起点就很难复现其曲线。
常见问题	只抄网络不抄训练表, 然后误判“EDM 未达到预期”。
解决方式	先按表跑通 baseline, 再做最小消融(一次只改一个超参)。

A.5 EDM 网络结构细节表(残差块数/注意力分辨率/heads)

引文: P3 p46 (Table 8)

怎样做	<p>1) 复现 P3 的网络族(DDPM++/NCSN++/ADM)时, 用这张表把“resblocks 与注意力插入点”写成确定配置, 避免实现不一致。</p> <p>表(P3 Table 8, 关键字段摘录):</p> <ul style="list-style-type: none"> - Residual blocks per resolution: DDPM++(VP)=4, NCSN++(VE)=4, ADM(ImageNet)=3 - Attention resolutions: - DDPM++(VP): {16} - NCSN++(VE): {16} - ADM(ImageNet): {32,16,8} - Attention heads: - DDPM++(VP): 1 - NCSN++(VE): 1 - ADM(ImageNet): 6-9-12 (按不同分辨率段)
为何这样做	P3 把网络结构作为可控维度做消融; 复现必须对齐这些离散选择。
常见问题	“注意力/残差块”写成口号但不写清分辨率与数量, 复现必失败。
解决方式	把注意力分辨率集合与每分辨率 resblocks 写进 config, 并在日志中打印。

A.6 iDDPM 的 U-Net 规模档位与 lr 缩放规则

引文: P1 p8,p11

怎样做	<p>1) 做 U-Net scaling 时, 用 P1 的 4 档通道数作为唯一档位, 并按其 lr 缩放规则联动调整。</p> <p>表(P1 scaling anchor):</p> <ul style="list-style-type: none"> - C: 64 / 96 / 128 / 192 - params: 30M / 68M / 120M / 270M - lr rule: baseline(C=128) lr=1e-4, others use lr = 1e-4 / sqrt(C/128)
为何这样做	P1 的消融与结论依赖一致的 scaling 口径。
常见问题	随意取中间通道数导致无法对照; 只改模型不改 lr 导致收敛与质量失真。
解决方式	只允许跳档位, 并把 lr 规则写死进脚本。

A.7 DDIM 的时间点子序列(τ)两种规则

引文: P2 p6-p8

怎样做	<p>1) 做少步采样, 用这两种规则生成 τ 子序列, 并用固定 x_T 做一致性诊断(见 5.2)。</p> <p>表(P2 τ rules):</p> <ul style="list-style-type: none"> - Linear: $\tau_i = \text{floor}(c * i)$ - Quadratic: $\tau_i = \text{floor}(c * i^2)$ - P2 经验选择: CIFAR-10 更偏向 quadratic, 其余数据集偏向 linear(作为起步点)。
为何这样做	P2 把“采样时间点分布”当成少步质量关键自由度。
常见问题	τ 取值不合理导致少步生成更像噪声, 但训练端并无明显异常。
解决方式	先用 eta=0 的确定性 DDIM 跑通一致性诊断, 再引入随机性(eta>0)。

A.8 OCM 的直接估计代价提示(为什么需要摊销)

引文: P6 p3-p4

怎样做	<p>1) 若想在少步下提升质量/似然并考虑学对角协方差, 先用这条代价提示判断“直接估计是否可行”。</p> <p>代价提示(P6): CIFAR-10 上要获得理想对角近似需要 $M \geq 100$ 个 Rademacher 样本; 每个样本都要一次前向+一次反传, 总成本约 2M 次网络评估, 推理不可用。</p>
为何这样做	这解释了 P6 为什么要用无偏 OCM 目标做摊销预测, 推理时只需一次前向。
常见问题	直接在推理时做 Hessian/对角估计, 速度不可用。
解决方式	在训练期用 OCM 头摊销学习, 推理期用一次前向输出对角项(见 3.7/5.8)。

版权与使用声明

- 本手册为对 [P1]-[P6] 的结构化二次整理, 不替代原论文; 以原文为准。
- 原论文版权归其作者与出版方所有; 本手册仅用于学习与研究目的。

参考文献 (仅 6 篇)

- [P1] Alex Nichol, Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672, 2021.
- [P2] Jiaming Song, Chenlin Meng, Stefano Ermon. Denoising Diffusion Implicit Models. arXiv:2010.02502, 2021.
- [P3] Tero Karras, Miika Aittala, Timo Aila, Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. NeurIPS 2022. arXiv:2206.00364.
- [P4] Hang Chen, Qian Xiang, Jiaxin Hu, Meilin Ye, Chao Yu, Hao Cheng, Lei Zhang. Comprehensive exploration of diffusion models in image generation: a survey. Artificial Intelligence Review, 58:99, 2025. DOI: 10.1007/s10462-025-11110-3.
- [P5] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, Yang You. Dynamic Diffusion Transformer. ICLR 2025.
- [P6] Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z. Xiao, Yingzhen Li, David Barber. Improving Probabilistic Diffusion Models with Optimal Diagonal Covariance Matching. ICLR 2025. arXiv:2406.10808.