用“有限时间位移”替代“瞬时切线”

pp.1–3 · Abstract, §§1–3

在只有一次网络调用时，重要的不是如何用更高阶公式积分一个既有速度场，而是网络究竟被训练来预测什么。普通 Flow Matching 学的是给定当前位置和时间后的条件平均瞬时速度。Shortcut 另加一个所需步长 d，让同一个网络直接学习跨越这一有限区间的平均位移率。若 d=1，单次预测就承担完整噪声到数据的映射；若 d→0，又退回普通流的局部极限。

这本手册围绕这一对象变化组织推导：先分清训练插值线与学得的边缘 ODE 轨迹，再推导两半步构造整步监督，最后回到本机 S7 和 S3，区分生成分布、数值端点误差和闭环控制成功率。原论文的自然图像 FID 不是本机光学图像 SWD，更不是机器人恢复成功率。

底本为 Kevin Frans、Danijar Hafner、Sergey Levine、Pieter Abbeel 的 arXiv:2410.12557v3（2025-06-23，17页，ICLR 2025）。arXiv 页面“view license”指向 non-exclusive license to distribute；它只授予 arXiv 分发权，不能据此声称获得公开逐段翻译授权。因此此处是原创中英双语数学深读，不是全文替代译本。关键原图仅用于紧贴方法的有限学术分析，原版完整保留。

$$x_{t+d} = x_t + d\, s_\theta(x_t, t, d), \qquad \lim_{d\to 0} s(x,t,d) = v(x,t)$$

English · original analysis

A one-call generator needs a different prediction target, not merely a different numerical integrator. Flow matching estimates an instantaneous conditional mean velocity. A shortcut network additionally receives a duration and estimates a finite-interval displacement rate. This distinction explains why a single architecture can serve both long and short sampling budgets. The reading route is: conditional velocity, finite-time composition, stopped bootstrap targets, image evidence, then local engineering counterexamples. The source is the seventeen-page arXiv v3. Its non-exclusive distribution license is recorded separately; this companion contains original analysis rather than a paragraph-by-paragraph translation.

随机配对的直线，并不等于边缘流轨迹是直线

pp.2–3 · Eqs.1–2, Fig.2

设 x_0 来自标准高斯，x_1 来自数据，二者独立随机配对。每一条训练插值线的导数都是 x_1-x_0；但网络只看到 x_t 与 t，不知道是哪一对端点。因此平方回归的最优解不是某一条线的导数，而是所有可行端点对的后验加权平均。

推导只需条件平方误差分解：任意候选函数与随机目标的风险，等于它与条件均值的距离，加上不依赖函数的条件方差（向量时取协方差的迹）。最小化风险便给出边缘速度 $\bar{v}$。不同时间的后验配对集合会变化，因而其积分曲线通常弯曲。不能把“训练路径是直线”错读为“一次 Euler 必定正确”。

最极端的 t=0 处，独立配对使 x_1 与 x_0 无关，条件平均等于数据均值减噪声。因此理想普通 FM 的一次 Euler 恰好把所有噪声送到数据均值，无法保留多模态。实际有限容量模型会偏离该理想极限，可能产生非平凡图像；这不推翻平方损失的最优解，而是提醒我们区分理论对象与有限训练模型。

可手算的双峰例子：令数据等概率取±a，噪声方差σ^2。给定 $x_t=x$，两个高斯条件密度的对数比为 $2tax/[(1-t)^2\sigma^2]$，所以后验数据均值为$a\cdot\tanh[tax/((1-t)^2\sigma^2)]$。再由$x_0=(x-tx_1)/(1-t)$消元，得到下式的边缘速度。取a=σ=1、t=0.5、x=0.25，均值为$\tanh(0.5)\approx0.462117$，速度≈0.424234。这个正速度来自当前位置已偏向正峰的后验；t=0则速度恰为−x，一大步仍回到0。条件分布随t变化，正是边缘轨迹弯曲的来源。

$$x_t=(1-t)x_0+tx_1,\quad u=x_1-x_0,\quad \bar{v}(x_t,t)=\mathbb{E}[u\mid x_t,t]$$

$$\mathbb{E}[\|f-u\|^2\mid x_t,t]=\|f-\mathbb{E}[u\mid x_t,t]\|^2+\operatorname{tr}\operatorname{Cov}(u\mid x_t,t)$$

$$x_0+\bar{v}(x_0,0)=x_0+\mathbb{E}[x_1]-x_0=\mathbb{E}[x_1]$$

$$\bar{v}(x,t)=\frac{a\tanh\left(\frac{tax}{(1-t)^2\sigma^2}\right)-x}{1-t},\qquad t<1$$

English · original analysis

The straight object is each sampled training interpolant. The learned marginal ODE follows a posterior average over many such interpolants. Conditional least squares separates approximation error from irreducible target variance. At zero time, independent noise–data pairing makes the optimal one-step Euler endpoint the data mean. This is an exact statement about the ideal objective, not a prediction that every finitely trained implementation returns the same image. A good finite-step model must learn more than that local tangent. A checkable two-point example gives a posterior mean of tanh(0.5)=0.462117 at time 0.5 and position 0.25, hence velocity 0.424234. The posterior, not the individual straight interpolation, controls the marginal dynamics.

两段半步为什么构成一个整步目标

pp.3–4 · Eqs.3–5, Fig.3

令 $\Phi_{t,t+d}$ 表示学得 ODE 在区间内的流映射，定义 $s(x,t,d)=(\Phi_{t,t+d}(x)-x)/d$。注意 s 是平均位移率，不是终点，不是未经归一化的位移。流映射的组合性质立即给出：长度2d的位移率，等于第一段 d 与第二段 d 的位移率平均。

关键在第二次查询：输入必须是第一段预测实际到达的 x′，时间必须推进到 t+d，而不是继续沿真实端点对线性插值。这样监督的是模型自身流的组合，而不是两条含随机配对歧义的经验切线。将整步输出回归到两个半步构成的目标，就把小步能力递归传播到大步。

论文式(4)与算法1一致地使用 t+d；PDF式(5)紧接的目标说明中第二次查询印成 t，和前文组合恒等式不一致。此处按算法1与式(4)实现，并在源映射记录这一排印差异。训练允许的二分尺度不意味着任意非二分步长都自动泛化。

$$x' = x + d\,s(x,t,d)$$

$$s(x,t,2d) = \frac{s(x,t,d) + s(x',t+d,d)}{2}$$

$$\mathcal{L} = \mathbb{E}_{d=0}\|s_\theta - u\|^2 + \mathbb{E}_{d>0}\left\|s_\theta(x_t,t,2d) - \mathrm{sg}\,\frac{s_{\mathrm{EMA}}(x_t,t,d) + s_{\mathrm{EMA}}(x',t+d,d)}{2}\right\|^2$$

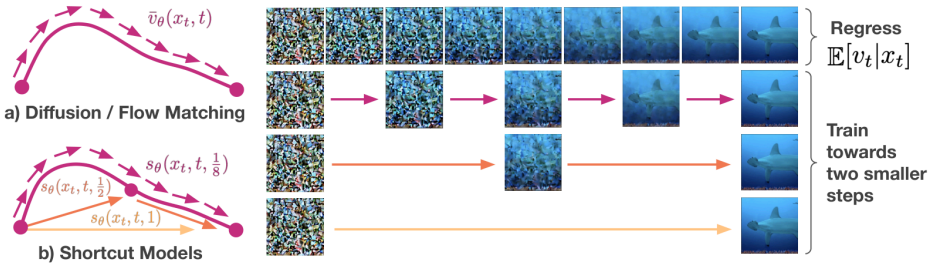

图3局部裁图：同一训练阶段混合经验 FM 与自生成两半步目标；原图见PDF第4页。

English · original analysis

A shortcut is the chord of a flow map divided by its duration. Composition of two flow maps yields the arithmetic mean of their equal-duration chord rates. The second query must use the predicted intermediate state and advanced time. Otherwise the target no longer describes the same composed trajectory. Equation 4 and Algorithm 1 specify the advanced time; the explanatory line following Equation 5 omits it in the source PDF. This companion follows the internally consistent algorithm and flags the discrepancy instead of silently copying it.

算法逐行：截断的是目标梯度，不是在线预测梯度

p.5 · Algorithm 1–2; §3.1

第1—3行采样端点、合法步长与时间，并产生训练输入。经验分支把 d 标为0，监督随机端点差；它提供来自真实数据的锚点。自举分支在无梯度上下文用 EMA 网络算两段预测，将平均结果停止梯度；在线网络仍接收该输入和2d，并正常反向传播。

若不停止目标梯度，在线输出与目标可同时移动，监督就不再是固定目标的自蒸馏。若第二步仍喂线性插值 x_{t+}d，监督的是经验耦合而非模型轨迹。若把输入 d 和预测目标2d 混淆，尺度语义会直接错位。最小步长1/128 构造目标时，论文改为查询 d=0 的 FM 分支。

采样时令 d=1/M，循环 M 次；M=1 请求完整区间的有限位移率。M=128 对应最细训练尺度。不要给“步长 d”绑定优化器更新次数，也不要把自举目标所消耗的额外前向隐藏在训练预算之外。

```
# Train: schematic; q is the bootstrap subset
x = (1-t)*noise + t*data
duration = d.clone()
target[~q] = data[~q] - noise[~q]
duration[~q] = 0
with no_grad():
    a = ema(x[q], t[q], d[q])
    mid = x[q] + d[q]*a
    b = ema(mid, t[q]+d[q], d[q])
    target[q] = (a+b)/2
prediction = net(x, t, 2*duration)
loss = mean(square(prediction-target.detach()))
loss.backward(); optimizer.step(); update_ema()
```

English · original analysis

The empirical branch grounds the model in data, while the bootstrap branch transfers that competence across durations. Stop-gradient applies only to the two-half-step target. The online full-step prediction still requires gradients. EMA target evaluation and sampling are separate from optimizer updates. The minimum-duration target uses the zero-duration flow branch. At inference, one specifies a step budget and follows the corresponding finite-duration predictions. A reproduction should test time advancement, intermediate-state construction, and actual field-call counts before comparing quality.

张量流：不是给输出多接一层线性头

pp.6,16 · DiT-B, Appendix B

原文的256×256 RGB图像先经 sd-vae-ft-mse 编为32×32×4潜变量。DiT-B使用2×2块，因此变为16×16=256个 token；每块含2×2×4=16个标量，投影为768维隐藏表示。t 和 d 各自做时间嵌入，再与类别条件一起调制 Transformer。输出反投影回潜变量速度 / 位移率，与输入 x 同形，最终由 VAE 解码成图像。

这条维度链说明：步长条件改变每个块的整体计算，不只是最后一层缩放。直接把普通 FM 输出乘更大 d，没有让网络看到 d，也没有训练有限时间位移目标，因而不构成 Shortcut。类别条件的引导还会改变目标流；训练大步目标时已蒸入的 CFG 尺度，不等价于采样期可任意修改尺度。

本机 S7 的 MicroDiT 处理32×32程序图，参数745,136，与论文潜空间 DiT-B不同。共享“DiT”名字不构成架构与数据等价。S3则输出未来动作块，最后两轴是动作时域和动作维度，不该沿用图像 H/W 的解释。

$$[B,256,256,3] \xrightarrow{\text{VAE}} [B,32,32,4] \xrightarrow{p=2} [B,256,16] \xrightarrow{\text{linear}} [B,256,768]$$

$$[B,256,768] \xrightarrow{\text{12 blocks,12 heads}} [B,256,768] \longrightarrow [B,32,32,4]$$

轴 / 条件	论文设置	不能混同的对象
t	当前噪声时间	序列物理时间
d	期望积分跨度	迭代计数
hidden	768；MLP比率4	原图通道
depth	12层	采样步数
EMA	0.999	第二个独立教师阶段

English · original analysis

The source DiT-B operates on VAE latents, not directly on RGB pixels. Patchification maps a 32×32×4 latent to 256 tokens of 16 scalars, then to 768 hidden channels. Time, duration, and class embeddings condition the transformer; unpatchification restores the input shape. Duration therefore changes the computed finite-step map throughout the network. Local MicroDiT and action-head experiments reuse the principle but differ in representation, capacity, and output semantics. A dimension audit prevents an apparent implementation match from concealing a different learning problem.

稳定性与16.7%额外训练计算从哪里来

pp.4–5,16 · §3.1, Table3

原文把批量的75%用于经验目标、25%用于自举目标。设一次前向成本为1、反向近似为2，普通更新成本约3。自举的四分之一批量多做两次无梯度前向，平均额外0.5，总成本3.5；相对3多约16.7%。它不是“参数多16%”，也不是对所有硬件成立的墙钟倍率。

经验目标有随机配对带来的较高方差，自举目标在固定输入、固定权重时是确定性的，所以更多经验样本既稳定又省算。EMA降低 d=0 的噪声向 d=1 输出传播的振荡；权重衰减0.1抑制初始化阶段追逐无意义自生成目标。论文说明这些选择帮助绕开严格热身与时间离散日程，但没有提供一个证明任意数据上都收敛的定理。

离散时间先采 d，再只在实际会查询的 d 倍数上采 t。尺度为1/128至1的8个二分长度。固定学习率10^{-4}；CelebA-HQ批量64、40万步，ImageNet批量256、80万步。完整比较要同时记优化样本数、EMA目标前向、以及 Reflow 数据生成开销；只对齐优化步数并不能让总计算相同。

用本机批量64核算：每步48个经验目标、16个自举目标；自举目标额外算16×2=32个样本前向。一次普通优化约等于64+2×64=192个前向样本成本，Shortcut为224，因此(224−192)/192=1/6。做6400步时，额外目标前向为32×6400=204800样本，正好对应S7报告中的Shortcut额外计数。这项会计不包含数据预处理、优化器、缓存或编译开销；这些影响墙钟时间，但不会改变该额外目标调用数。

$$C_{\mathrm{FM}} = 1 + 2 = 3, \quad C_{\mathrm{shortcut}} = 1 + 2 + \tfrac{1}{4}(2) = 3.5$$

$$\frac{C_{\mathrm{shortcut}}}{C_{\mathrm{FM}}} - 1 = \frac{0.5}{3} \approx 16.7\%$$

批量64的一步更新	等效前向样本数
在线前向	64
在线反向（近似）	128
额外EMA目标前向	16×2=32
Shortcut合计 / 普通FM	224 / 192

English · original analysis

The advertised training overhead is a compute-unit estimate. With a backward pass approximated as two forwards, the ordinary update costs three units and a quarter-batch two-query bootstrap adds half a unit. EMA and weight decay address noisy self-targets; they are practical stabilizers rather than a global convergence theorem. Time sampling is restricted to the discrete locations used at inference. Fair budgets include target generation and any separately produced Reflow dataset, not just optimizer iterations. With batch 64, sixteen bootstrap examples require thirty-two extra forward examples per update. Over 6,400 updates that is 204,800 extra target forwards, matching the local S7 accounting. This counts work; it does not predict device utilization or wall-clock ratios.

同架构主表：一步质量、任意预算与阶段数是三个维度

pp.7–8 · Table1

下表逐项复盘原文 ImageNet-256 的 FID-50k，均采用相同 DiT-B 主干和统一实现。Shortcut 一步40.3，明显好于普通 FM 的324.8；但两阶段 Progressive Distillation 一步35.6更低。Shortcut 的卖点不是在所有列绝对最优，而是在单阶段训练同时保留一步、少步、多步能力。

这张表是目标函数与训练流程的对照，不是严格只改一个开关的 EMA / weight-decay 消融。不能把它包装成“独立证明 d 条件的唯一因果贡献”。原文还列 CelebA-HQ：FM 为7.3/63.3/280.5，Shortcut 为6.9/13.8/20.5（128/4/1步）。多步略好被作者解释为可能的隐式正则化，属于待验证假设。

原表括号标出非设计使用条件，例如 Progressive Distillation 用128步和4步、普通扩散做1步。将这些数值与方法原本支持的预算分开读，才能避免用超出目标定义的失败夸大胜利。

ImageNet-256 / FID↓	128步	4步	1步
Progressive Distillation	(201.9)	(142.5)	35.6
Consistency Distillation	132.8	98.01	136.5
Reflow	16.9	32.8	44.8
Flow Matching	17.3	(108.2)	(324.8)
Consistency Training	42.8	43.0	69.7
Live Reflow	46.3	95.8	58.1
Shortcut	15.5	28.3	40.3

English · original analysis

Table 1 is a controlled comparison of objectives and training procedures, not a one-switch component ablation. Progressive distillation wins the one-step ImageNet column, whereas Shortcut combines one-stage training with support for all tested step budgets. The many-step improvement over ordinary flow matching is small and the proposed regularization explanation remains a hypothesis. Parenthesized source entries denote evaluation outside the intended operating regime. Those qualifications determine what can be concluded from the numbers.

扩规模与策略实验：证据能走到哪一步

pp.8–10,16–17 · §§5.2–5.5, Tables2–3

原文 DiT-XL Shortcut 达到 ImageNet-256 一步 FID10.6、4步7.8、128步3.8，676M参数、250轮。表2中的其他生成器使用不同架构和计算预算，不能据此断言在同等算力超过所有 SOTA。结果支持更窄的结论：该自举方法在作者测试规模下没有因自训练而失去随模型增大改善的趋势。

同一起始噪声下，多步和单步通常保留全局内容；一步剩余错误更偏细节。噪声空间插值使用保方差组合，两独立高斯样本的系数平方和为1，所以中间点仍服从相同高斯先验。这种生成插值不是对既有真实图像的精确反演。

机器人实验沿 Diffusion Policy，在 Push-T 和 Transport 上保留网络及大部分超参，仅加 d 条件、改为 FM目标，并将 AdamW权重衰减从0.001改至0.1。因此策略结果包含多个共同变化。原文支持其任务上的一步策略可行性，不能推出任何开环专家数据经 Shortcut 都能自动产生闭环恢复能力。

$$x_0(\lambda) = \lambda x_0^{(a)} + \sqrt{1-\lambda^2}\, x_0^{(b)}, \quad \mathrm{Var}[x_0(\lambda)] = I$$

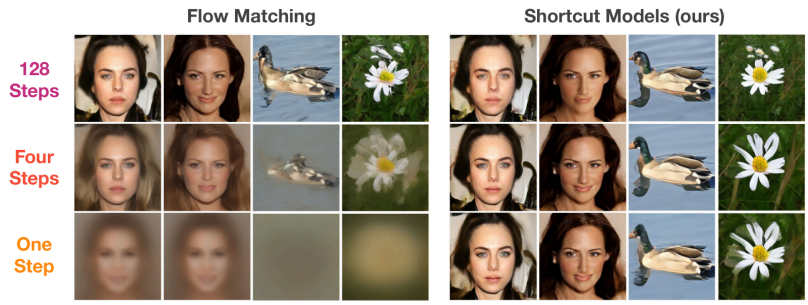

图1局部裁图：相同噪声、不同步数的普通 FM 与 Shortcut；原图PDF第1页。

English · original analysis

Scaling improves the source model within its tested range: the XL shortcut reaches FID 10.6 at one step and 3.8 at 128 steps. The cross-paper comparison is not compute-matched. Variance-preserving interpolation keeps the Gaussian prior stationary because squared coefficients sum to one. Robotic transfer changes the objective, duration conditioning, and weight decay together. The control results establish feasibility on the reported tasks, not automatic recovery on an arbitrary expert dataset.

S7反例：2 NFE 下高阶方法为什么更差

本机 S7 · DELIVERY_REPORT.md; image_metrics.csv

S7 主实验统一745,136参数 MicroDiT、6400×64优化样本预算，保留375个图像质量配置。其 Feature SWD来自独立分类器的256维特征，128个固定方向上的平均一维 Wasserstein-1；不是论文 FID，也不是二维实验使用的 sliced-W_2。

在2 NFE预算下，Heun、预算化RK3与ABM的实现实际都退化为一次Heun更新，故三者同得0.7929，并非三个独立算法都完成完整高阶步。两次Euler得0.1692；Heun输出31.14%通道越界，两次Euler为0.012%。Heun先从t=0预测一大步，再在t=1对该预测端点求速度；该端点通常不在真实数据流形附近，也不是训练时t=1的输入分布。端点速度误差被半步权重直接灌入结果，局部截断误差的渐近阶数此时无保护作用。

Shortcut 一步0.1819、两等长步0.1587；四次Euler0.1436更好。非均匀两步Shortcut却到0.6466，因为它请求未训练的非二分d。这是步长分布外泛化与网格误差共同作用，不能只归因于曲率选点。

$$\hat{x}_1 = x_0 + v_\theta(x_0, 0), \quad x_1^{\text{Heun}} = x_0 + \tfrac{1}{2}\left[v_\theta(x_0, 0) + v_\theta(\hat{x}_1, 1)\right]$$

$$x_{1/2} = x_0 + \tfrac{1}{2} v_\theta(x_0, 0), \quad x_1^{\text{Euler2}} = x_{1/2} + \tfrac{1}{2} v_\theta(x_{1/2}, 1/2)$$

English · original analysis

At two field calls, the budgeted Heun, RK3, and ABM implementations all execute the same single Heun update. Their identical failure is one counterexample, not three independent replications. A one-unit predictor lands away from the trained endpoint distribution before the corrector queries time one. The measured feature SWD and out-of-range channel rate expose that failure. Uniform Shortcut uses trained dyadic durations; a nonuniform grid additionally tests duration generalization. At four calls ordinary Euler is best in this particular measured grid.

3.24×来自编译执行，不来自把逻辑NFE减半

本机 S7 · image_latency.csv, image_precision_canary.csv

S7 在 RTX4060Ti、BF16自动混合精度和TF32开启条件下，用 CUDA Event测每批完整采样器延迟，预热及编译首调用另计。普通1-RF、4步、batch1，从Eager29.57ms降到Inductor9.14ms，约3.24×。这是同一完整网络和相同步数下的执行模式比较；不能说成Shortcut训练使推理快3.24倍。

小网络容易受Python调度、算子启动和中间张量读写支配，编译图可减少边界、融合逐元素运算并消除部分中间存储。但实际速度来自测量，不能从理论 FLOPs 单独推得。显式CFG教师8有效NFE、学生4有效NFE，在batch16编译后是11.12→10.41ms，仅1.07×；减少分支工作不等于毫秒减半。

默认编译可能合并低精度舍入边界，改变累积误差；emulate_precision_casts=True用于更贴近Eager的舍入语义，不是“更高精度”开关。最差配对RMSE0.386降至0.119，代价是该检查中位延迟增2.5%。正式编译300行中位RMSE0.0088，最大仍0.171，所以Eager质量和编译延迟不能当作同一份无差异结果。

可核算的舍入反例：BF16在1附近的间隔为2^{-7}。令a=1，b=c=2^{-8}，并用最近偶数舍入；逐次舍入时，1+b恰在中点而回到1，再加c又回到1。若中间不舍入，最后1+b+c=1.0078125可精确表示。这不是对某个Inductor融合内核的逐指令复现，而是说明为什么保留或删除低精度中间量会改变结果。多步生成把这种差异反复送回网络，故误差取决于具体轨迹；既不能从一步小差异推出全程无差异，也不应把编译后的质量直接假定为Eager质量。

$$29.57/9.14 \approx 3.24, \quad 11.12/10.41 \approx 1.07$$

$$\mathrm{round}_b(\mathrm{round}_b(a+b)+c) \neq \mathrm{round}_b(a+b+c)$$

$$\mathrm{RN}_{\mathrm{BF16}}\left(\mathrm{RN}_{\mathrm{BF16}}(1+2^{-8})+2^{-8}\right) = 1$$

$$\mathrm{RN}_{\mathrm{BF16}}(1+2^{-8}+2^{-8}) = 1+2^{-7} = 1.0078125$$

English · original analysis

The 3.24× result compares execution modes for a fixed four-step ordinary-flow sampler. Inductor can reduce launch and memory overhead, but logical NFE alone does not determine milliseconds. CFG distillation halves the counted branch work yet yields only 1.07× in the measured compiled batch-16 pair. Precision-cast emulation preserves more eager-style rounding boundaries; it improves a numerical canary without making all outputs identical. Quality and timing evidence must retain their execution-mode labels. Near one, BF16 spacing is 2^{-7}. Two half-ULP increments each disappear under separate round-to-nearest-even operations; accumulated before rounding, they produce 1.0078125. This toy calculation explains why removing a rounding boundary can change results without changing the mathematical real-valued expression.

S3反例：单步头很快，失去接触后仍无恢复知识

本机 S3 · final_message.md；4200回合主实验

S3 的4200条配对闭环回合把动作生成质量与任务恢复分开。主实验单步OT-CFM-DiT仅5/100成功，全部走右侧；Shortcut主实验0/100。独立恢复数据对照中Shortcut达13/100（左12、右1），而普通Flow10/100；差3个百分点的配对95%区间为−6至+12，不足以认定方法优势。

教师第32控制tick的第一动作误差，在专家状态约17.8mm，在自己闭环状态约234.3mm，相差约13.16倍。这里的比值比较两类输入分布，不是宣称同一个开环误差经已证明的动力学常数放大13倍。非抓持推块一旦失去接触，继续沿专家接触轨迹前进会制造从未见过的状态；低延迟或更直的生成轨迹不能填补这一覆盖空洞。

可操作的下一步是收集可达恢复状态接管，并用物体相对坐标表达动作与接触几何，再在共同初态、噪声和双侧路线下评估。训练 Reflow 可以降低速度偏差，却继承教师终点及数据支撑；应同时报告接触保持、恢复成功、碰撞与左右模式覆盖，而不只报告动作头毫秒数。

$$234.3/17.8 \approx 13.16$$

$$\|\Delta x_{n+1}\| \lesssim (1 + hL)\|\Delta x_n\| + h\|e_\theta(x_n)\|$$

S3单步策略	主实验成功/100	恢复对照/100
OT-CFM-UNet	0	10
OT-CFM-DiT	5	6
Reflow	0	1
Shortcut	0	13

English · original analysis

A fast action head does not create recovery demonstrations. In the local main experiment, one-step DiT succeeds five times out of one hundred and only on one route. Shortcut improves to thirteen successes in a separate recovery-data experiment, but its paired difference from ordinary flow is inconclusive. The 17.8-to-234.3 mm gap compares expert and closed-loop input states. It points to distribution shift after contact loss rather than a theorem about error amplification. Recovery-state coverage and relative geometry change the learning problem more directly than reducing sampler latency.

与S3/S7/S8/S9的工程接口：先定对象再选加速

原文 §§3,5; 本机S3/S7/S8/S9日志

S3接口是观察条件→动作块：d是生成时间跨度，不能混成真实控制周期。保留动作归一化、相对坐标和可达恢复状态的数据契约；相同噪声下比较左右路径及失接触后恢复，而不是只看离线动作MSE。

S7接口是噪声→程序光学图像：在1—2调用预算选Shortcut，在4调用且优先分布质量时，当前证据偏向普通Euler。必须验证非二分d、t=1预测端点、越界比例，并把编译速度与相应数值差异并列。

S8接口是事件张量→光流 / 流体量：CUDA Graph可加速固定形状计算，但Shortcut不是事件去噪或光流监督的自动替代。若用其做条件生成，须保留事件时间单位、坐标约定与涡量差分的物理尺度；824.5FPS是S8既有端到端图捕获结果，不是本文模型速度。

S9接口是世界系点云→SE(3)动作：平移与旋转应保持其几何表述和尺度。对旋转不能简单把欧氏位移公式当作全局有效的群积分器；SDF近端投影也不能被Shortcut的软监督替代。用分布质量检验生成目标，用几何安全检验碰撞，用闭环回合检验控制，是这组论文联读的最终分工。

术语	本册统一用法
shortcut	有限时间平均位移率；方法名保留Shortcut
NFE	实际网络 / 速度场调用数
stop-gradient	目标停止梯度
EMA	权重指数移动平均
SWD / FID	分别保留本机/论文统计定义

English · original analysis

Choose the object before choosing the accelerator. S3 requires recovery-aware action data; S7 studies a noise-to-image map; S8 measures event processing and flow-derived physical quantities; S9 adds world-frame geometry and collision constraints. Shortcut duration belongs to generative time, not automatically to physical control time. A Euclidean finite-step identity does not by itself define a globally valid rotation integrator. The useful synthesis is modular: finite-step learning for generation, graph compilation for execution, coordinate registration for invariance, and explicit safety constraints for collision handling.