

U2.T2 — CNFs, FFJORD, and the Cliffhanger

Flow-Based Generative Models · UFRJ · 2026.2

01 | The continuity equation

The question on the board

Since U1.T2, one question has stood on the left board:

What does $\log|\det J|$ become in continuous time?

Today:

1. the equation that governs a density moved by a field;
2. the identity that answers the board;
3. the algorithm that uses it, and the estimator it needs;
4. a family of penalties that can make the solve easier;
5. the price of the answer.

The two costs, version 2, annotated

THE TWO COSTS (V_2 , ANNOTATED – $U_2.T_1$)

Family	Density	Sampling	Trick, and its price
NICE / RealNVP	exact, triangular	direct	coupling; depth
Glow	exact	direct	mixing, multi-scale
MAF	parallel	sequential	AR, data side
IAF	sequential	parallel	AR, source side
Residual flow	direct; series log-det	iterative	contraction, traces
Neural ODE	inverse: ODE solve / NFE; $\log \det J$???	ODE solve / NFE	no inverse layer; Lipschitz field

One cell is marked. It is today's question.

One letter changes

- Until today the field was f_θ : a generic network in a generic ODE.
- From now on the ODE carries **probability**. Its field is u_t , and u_t^θ when learned.
- The object is the same: the Neural ODE of U2.T1, renamed.

When a field pushes densities around, we call it u — remember the symbol table.

The standing hypothesis

Today's statements are pointwise equations for p_t . They need p_t differentiable in x , which asks for one more derivative of the flow than Tier 2. Here $\partial_x u$ is the Jacobian of u in x .

STANDING HYPOTHESIS (H) – U2.T2

u is continuous on $[0, 1] \times \mathbb{R}^d$, C^2 in x with $\partial_x u$, $\partial_x^2 u$ continuous on $[0, 1] \times \mathbb{R}^d$, and $\partial_x u$ bounded. The initial density is $p_0 = \mathcal{N}(0, I_d)$.

Then ψ_t is a C^2 diffeomorphism, and the density p_t of $\psi_t(X_0)$ is positive and C^1 in (t, x) .

Every box that uses (H) says so. The weak form needs only Tier 1; the integrated identity needs only Tier 2.

The density of the moved samples

DEFINITION (THE DENSITY OF THE MOVED SAMPLES)

Under (H), let $X_0 \sim p_0$ and $X_t = \psi_t(X_0)$. The density p_t of X_t satisfies, for every bounded measurable φ ,

$$\mathbb{E}_{X \sim p_t} [\varphi(X)] = \mathbb{E}_{X_0 \sim p_0} [\varphi(\psi_t(X_0))].$$

Since ψ_t is a diffeomorphism, U1.T1's change-of-variables theorem gives $p_t(x) = p_0(\psi_t^{-1}(x)) |\det J_{\psi_t^{-1}}(x)|$.

That formula needs the flow map, which is the solution of an ODE. Does p_t satisfy an equation written with u alone?

Example: a linear field on the line

EXAMPLE (A LINEAR FIELD ON THE LINE)

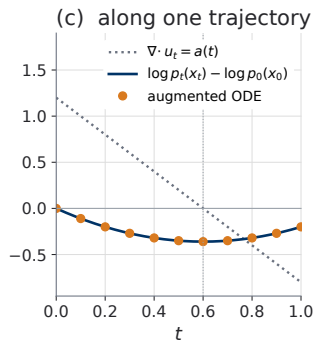
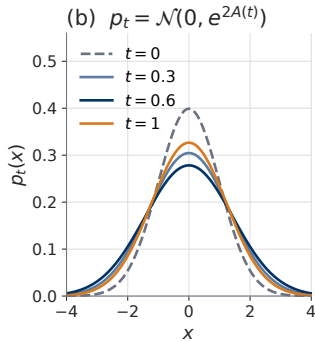
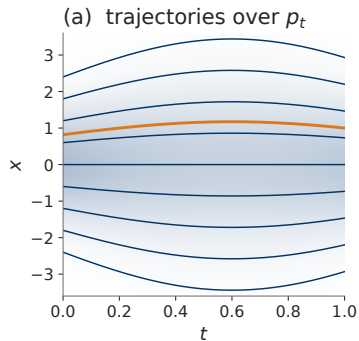
$d = 1$, $u_t(x) = a(t)x$ with $a(t) = 1.2 - 2t$, and $p_0 = \mathcal{N}(0, 1)$. Then

$$\psi_t(x_0) = x_0 e^{A(t)}, \quad A(t) = 1.2t - t^2,$$

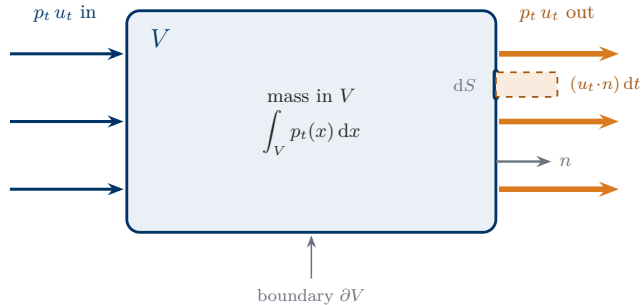
so $p_t = \mathcal{N}(0, e^{2A(t)})$.

The field expands the line until $t = 0.6$, then contracts it: the scale $e^{A(t)}$ peaks at 1.4333 and ends at 1.2214.

The example, pictured



Derivation 1: a control volume



$$\frac{d}{dt} \int_V p_t dx = - \oint_{\partial V} p_t u_t \cdot n dS$$

Derivation 1: the balance

$$\frac{d}{dt} \int_V p_t dx = - \oint_{\partial V} p_t u_t \cdot n dS$$

1. The left side is $\int_V \partial_t p_t dx$.
2. The divergence theorem turns the right side into $-\int_V \nabla \cdot (p_t u_t) dx$, where $\nabla \cdot g = \text{tr}(\partial g / \partial x)$ for a vector field g .
3. So $\int_V [\partial_t p_t + \nabla \cdot (p_t u_t)] dx = 0$ for **every** region V : the bracket vanishes at every point.

This is the picture. It is not yet a proof: “flux” is a modelling statement.

Derivation 2: the weak form

For a smooth, compactly supported test function φ :

$$\begin{aligned}\frac{d}{dt} \mathbb{E}_{x \sim p_t}[\varphi(x)] &= \mathbb{E}_{x_0 \sim p_0} [\nabla \varphi(\psi_t(x_0)) \cdot u_t(\psi_t(x_0))] \\ &= \mathbb{E}_{x \sim p_t} [\nabla \varphi(x) \cdot u_t(x)]\end{aligned}$$

It uses nothing but the trajectory definition: Tier 1 is enough.

Derivation 2: two gaps

Write the right side as an integral and move the derivative onto $p_t u_t$:

- **[C1]** *Integrate by parts.* Which boundary term? Why does it vanish? Which regularity of $p_t u_t$ is used?
- **[C2]** *Localize.* Why does “zero against every test function” mean “zero at every point”?

Both gaps are filled, in full, in the recorded clinic. It is **required viewing before U3.T1.**

The continuity equation

BOX U2.T2-1 – THE CONTINUITY EQUATION, UNDER (H)

$$\partial_t p_t + \nabla \cdot (p_t u_t) = 0$$

p_t is positive and C^1 . The weak form needs only Tier 1.

This exact equation is the foundation of everything after the break. U3 begins by reading it in the other direction.

Check on the example

EXAMPLE (DIRECT DIFFERENTIATION)

$$\begin{aligned}\partial_t p_t &= a(t) \left(x^2 e^{-2A(t)} - 1 \right) p_t, \\ \nabla \cdot (p_t u_t) &= a(t) \left(1 - x^2 e^{-2A(t)} \right) p_t.\end{aligned}$$

The two terms cancel at every (t, x) .

02 | Instantaneous change of variables

The computation, steps 1 and 2

Along a trajectory $x_t = \psi_t(x_0)$, with $\log p_t$ of class C^1 :

$$\frac{d}{dt} \log p_t(x_t) = (\partial_t \log p_t)(x_t) + \nabla \log p_t(x_t) \cdot u_t(x_t)$$

Expand the continuity equation and divide by $p_t > 0$:

$$\partial_t \log p_t = -\nabla \cdot u_t - u_t \cdot \nabla \log p_t$$

Step 3: the transport terms cancel

$$\begin{aligned}\frac{d}{dt} \log p_t(x_t) &= (-\nabla \cdot u_t - u_t \cdot \nabla \log p_t) \\ &\quad + \nabla \log p_t \cdot u_t \\ &= -(\nabla \cdot u_t)(x_t)\end{aligned}$$

Every term on the right is evaluated at x_t . The first bracket is the continuity equation; the last term is the observer riding with the sample. What is left is **compression and expansion**.

Instantaneous change of variables

BOX U2.T2-2 – INSTANTANEOUS CHANGE OF VARIABLES

Under (H), along every trajectory $x_t = \psi_t(x_0)$:

$$\frac{d}{dt} \log p_t(x_t) = -(\nabla \cdot u_t)(x_t)$$

and, integrating over $[0, 1]$ with $x_1 = \psi_1(x_0)$:

$$\log p_1(x_1) = \log p_0(x_0) - \int_0^1 (\nabla \cdot u_t)(x_t) dt$$

Equivalently, $\log |\det J_{\psi_1}(x_0)| = \int_0^1 (\nabla \cdot u_t)(x_t) dt$.

Cited as (Chen et al. 2018, sec. 4, Thm. 1).

The question that has stood on the board

What does $\log|\det J|$ become in continuous time?

U1.T2 wrote it. U2.T1 circled the cell. Today it is answered.

$$\log|\det J_{\psi_1}(x_0)| = \int_0^1 (\nabla \cdot u_t)(x_t) dt$$

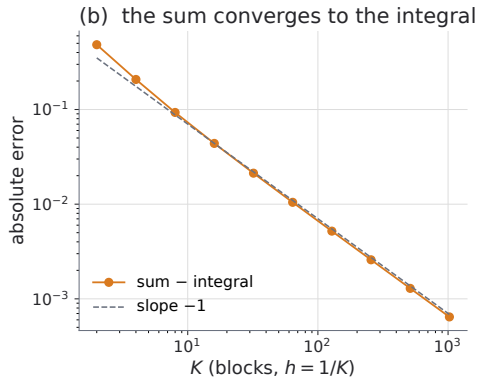
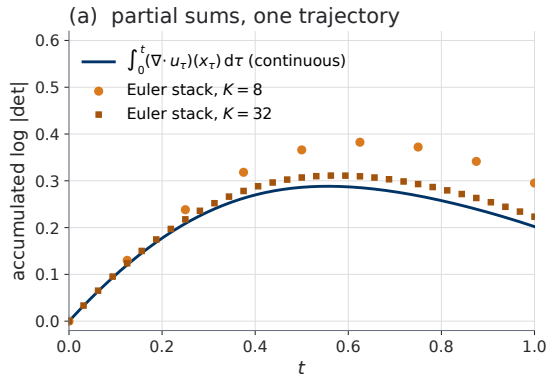
A determinant became a trace, accumulated in time. In the density it enters with a minus sign, as in U1. **Checked off the board.**

Discrete against continuous

	U1: a stack of K blocks	U2: a flow
log-determinant	$\sum_k \log \det J_{\psi_k}(z_{k-1}) $	$\int_0^1 (\nabla \cdot u_t)(x_t) dt$
why tractable	the architecture	nothing is designed: a trace
ancestor	change of variables (U1.T1)	same theorem, differential form

Same object, discrete against continuous. U1 as in U1.T1: block k maps z_{k-1} to $z_k = \psi_k(z_{k-1})$.

The sum converges to the integral



The solve direction

To evaluate a density, x_1 is given at $t = 1$ and a backward solve finds x_0 .

DEFINITION (LOG-DENSITY CORRECTION)

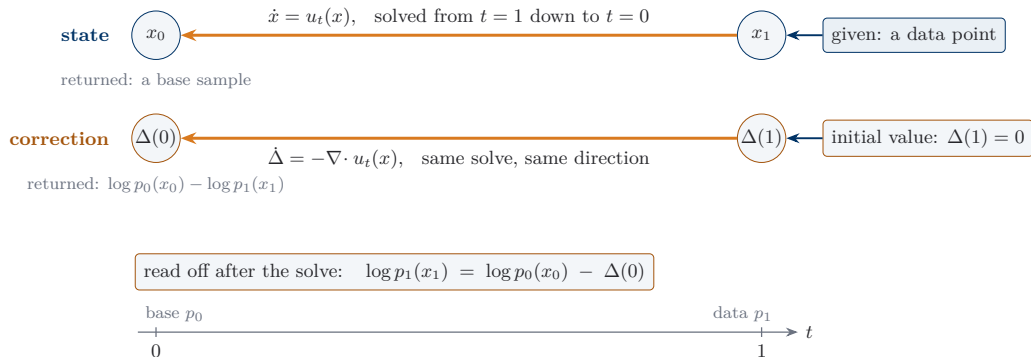
Along the trajectory through the data point x_1 :

$$\Delta(t) := \log p_t(x_t) - \log p_1(x_1)$$

Then $\Delta(1) = 0$, $\dot{\Delta} = -(\nabla \cdot u_t)(x_t)$, and $\Delta(0) = \log p_0(x_0) - \log p_1(x_1)$:

$$\log p_1(x_1) = \log p_0(x_0) - \Delta(0)$$

One backward solve, two quantities



The trap: do not start a variable at 0 and call its terminal value “the log-likelihood”. $\Delta(0)$ is a *difference* of log-densities.

The example, with numbers

EXAMPLE (THE LINEAR FIELD, WITH THE SECOND IDENTITY)

$\nabla \cdot u_t = a(t)$ at every x , so $\Delta(o) = A(1) = 0.2$ for **every** data point. For $x_1 = 1.0$:
 $x_0 = 0.8187$, $\log p_0(x_0) = -1.2541$, so

$$\log p_1(x_1) = -1.2541 - 0.2000 = -1.4541,$$

the log-density of $\mathcal{N}(o, e^{2A(1)})$ at x_1 .

A second proof, and what it needs

- $J_t = J_{\psi_t}(x_0) = \partial \psi_t(x_0) / \partial x_0$ solves $\dot{J}_t = (\partial_x u)(t, x_t) J_t$, $J_0 = I$ (Tier 2).
- Jacobi: $\frac{d}{dt} \log \det J_t = \text{tr}(J_t^{-1} \dot{J}_t) = (\nabla \cdot u_t)(x_t)$.
- That is the identity along trajectories, with **Tier 2 only** and no continuity equation.

The continuity route is run under (H), and gives more: the pointwise equation, and uniqueness.

PS2.2, assigned at the board (1 of 2)

Setting: u satisfies (H); $p_0 = \mathcal{N}(0, I_d)$, smooth and positive; p_t is the density of $\psi_t(X_0)$ and satisfies the continuity equation pointwise.

- **(a)** Justify the total-derivative expansion of $d/dt \log p_t(x_t)$ along a trajectory (which chain rule, applied to what function of (t, x) ?).
- **(b)** Expand the continuity equation's divergence term and carry out the cancellation in full, tracking which terms are evaluated at x versus along x_t .
- **(c)** Derive the integrated likelihood formula, including the direction-of-solve bookkeeping and all signs, for a sample given at $t = 1$.

PS2.2, assigned at the board (2 of 2)

- **(d)** *The bridge, made rigorous:* Euler-discretize the ODE and, under a small-step condition ensuring $I + hJ$ is nonsingular and the matrix-log expansion is valid, show $\log|\det(I + hJ)| = h \operatorname{tr}(J) + O(h^2)$. Conclude in one sentence how U1's sum of log-dets becomes U2's divergence integral.
- **(e)** *Bonus.* Where exactly did (b) use smoothness of p_t , and what could fail without it? (One paragraph; no measure theory expected.)

In (d), J is the field's Jacobian at the Euler point, $\partial_x u(t_k, x_k)$.

03 | FFJORD: the likelihood as an augmented ODE

From identity to algorithm

Replace the field u_t by a network u_t^θ , and solve from $[x; \Delta](1) = [x_1; 0]$ down to $t = 0$:

$$\frac{d}{dt} \begin{pmatrix} x \\ \Delta \end{pmatrix} = \begin{pmatrix} u_t^\theta(x) \\ -\nabla \cdot u_t^\theta(x) \end{pmatrix}$$

$$\mathcal{L}_{\text{NLL}}(\theta) = \mathbb{E}_{x_1 \sim q} \left[-\log p_0(x_0^\theta(x_1)) + \Delta_{x_1}^\theta(0) \right]$$

This is a **continuous normalizing flow** (CNF), trained by maximum likelihood.

Training through the solve

- Reverse the clock, $\tau = 1 - t$: the state $y = [x; \Delta]$ starts at the **fixed** value $[x_1; 0]$ and ends at $[x_0; \Delta(0)]$, where the loss is read.
- That is the setting of **Box U2.T1-3**: a terminal loss on an ODE with a fixed initial value. Nothing new is derived.
- The augmented field is *triangular*: Δ never feeds back into x . Box U2.T1-3 asks more of u^θ than (H): $\nabla \cdot u^\theta$ globally Lipschitz in x and C^1 in (x, θ) . FFJORD requires the field and its first derivatives to be Lipschitz (Grathwohl et al. 2019, sec. 2.2).

What the divergence costs

A **vector-Jacobian product** (VJP) $w^\top \partial_x u$ is one reverse-mode pass through u , at about the cost of one evaluation of u .

Divergence per RHS call	calls of u	extra passes	returns
exact trace	1	d VJPs	$\nabla \cdot u$ exactly
Hutchinson	1	1 VJP	unbiased, with variance

NFE counts calls of u . The VJPs are extra work, visible in wall-clock time, and are **not** extra NFE.

The Hutchinson debt, paid

BOX U2.T2-3 – HUTCHINSON'S TRACE ESTIMATOR (V2)

$A \in \mathbb{R}^{d \times d}$, $\mathbb{E}_v[vv^\top] = I_d$ (Rademacher or Gaussian), $S = \frac{1}{2}(A + A^\top)$.

(i) Unbiased: $\mathbb{E}_v[v^\top A v] = \text{tr}(A)$.

(ii) Symmetric part only: $v^\top A v = v^\top S v$.

(iii) Variance. Gaussian: $2\|S\|_F^2$. Rademacher: $2(\|S\|_F^2 - \sum_i A_{ii}^2)$.

The Rademacher variance is below the Gaussian one by exactly $2 \sum_i A_{ii}^2$, so it is never above it. Version 1 is U1.T2's box, "Hutchinson's trace estimator"; its proof was deferred to today.

One probe, held fixed during a solve

BOX U2.T2-3 (IV) – THE FIXED PROBE

Let x_t not depend on v , and $A_t = (\partial_x u)(t, x_t)$. With one probe fixed on $[0, 1]$:

$$\hat{l}(v) = \int_0^1 v^\top A_t v \, dt = v^\top \bar{A} v, \quad \bar{A} = \int_0^1 A_t \, dt.$$

$\mathbb{E}_v[\hat{l}(v)] = \int_0^1 (\nabla \cdot u_t)(x_t) \, dt$, and (iii) holds with $\bar{A}$.

Fubini: $\mathbb{E}_v \|v\|^2 = d < \infty$ and $\sup_t \|A_t\| < \infty$. Here A_t is the Jacobian along the trajectory (Box U2.T1-3 writes it $A(t)$); on this page $A(t)$ is the antiderivative of the linear example, and for that field $A_t = a(t)$.

Short proofs

- **(i)** $v^\top A v = \text{tr}(A v v^\top)$, so $\mathbb{E}_v v^\top A v = \text{tr}(A \mathbb{E}_v[v v^\top]) = \text{tr}(A)$.
- **(ii)** A number equals its transpose: $v^\top A v = v^\top A^\top v$. Average the two.
- **(iii)** Gaussian: $v^\top S v = \sum_i \kappa_i \tilde{v}_i^2$ ($S = Q \text{diag}(\kappa) Q^\top$, $\tilde{v} = Q^\top v$), variance $2 \sum_i \kappa_i^2$.
Rademacher: $v_i^2 = 1$ leaves $2 \sum_{i < j} S_{ij} v_i v_j$, with pairwise uncorrelated products.

The proof of (i) needs no zero mean: only $\mathbb{E}_v[v v^\top] = I_d$.

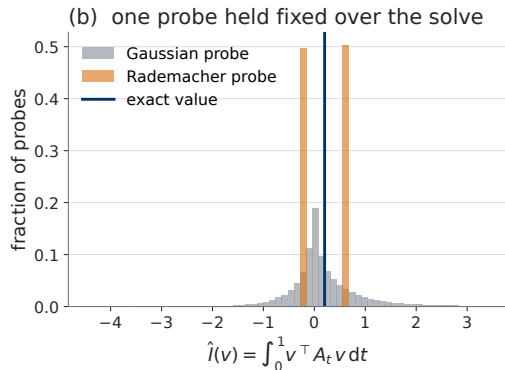
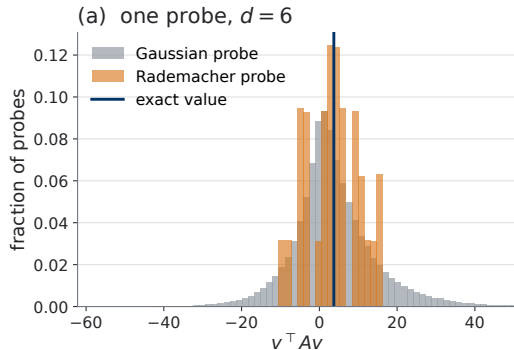
The contract for the probe

Contract. One probe per example, for one likelihood solve, held fixed across every RHS call of the adaptive solver (Grathwohl et al. 2019, sec. 3.1, eq. 8). The course adds: and across every recomputation of the adjoint.

- A deterministic augmented ODE is what a solver expects. A probe redrawn per call is a random field.
- The adjoint reconstructs the state backward and must see the field it saw forward.

Limits. The statement is about the *exact* integral: a solver's steps depend on v . No blanket claim that every training gradient is unbiased. The estimate is unbiased for $\log p_1$, not for p_1 .

Variance, measured



- A **free-form field** u_t^θ : no coupling pattern, no triangular Jacobian, no contraction.
- A **stochastic divergence correction**: one VJP per RHS call, probe fixed for the solve.
- A **sensitivity method**: the adjoint of U2.T1, or direct differentiation of the solver.

The design tension of U1 is dissolved. Each likelihood and each training step still performs a numerical integration.

04 | Regularization: conditioning the solve

An observation protocol

U2.L1, TODO 2.3 and its reference run: the forward NFE of a small Neural ODE classifier, at every training step (seed 0, dopri5 at its defaults).

first steps	at step 4	peak	at step 300
44	20	134 (step 48)	122

A solver's cost depends on the field, and the field can change at every optimizer step. **The count can change during training.** No direction is claimed.

Two penalties

(Finlay et al. 2020): charge the trajectory x_t through x_1 , for the speed and the variation of the field, with weights $\lambda_K, \lambda_J \geq 0$.

$$\begin{aligned}\mathcal{L}_{\text{reg}}(\theta) = & \mathcal{L}_{\text{NLL}}(\theta) + \lambda_K \mathbb{E}_{x_1 \sim q} \left[\int_0^1 \|u_t^\theta(x_t)\|^2 dt \right] \\ & + \lambda_J \mathbb{E}_{x_1 \sim q} \left[\int_0^1 \|(\partial_x u_t^\theta)(x_t)\|_F^2 dt \right]\end{aligned}$$

For $J = (\partial_x u_t^\theta)(x_t)$: $\|J\|_F^2 = \mathbb{E}_v \|v^\top J\|^2$, and $v^\top J$ is the VJP the divergence estimate already computes: **no extra VJP call** (Finlay et al. 2020, secs. 4–5).

The generalized adjoint closes the loop

- Both penalties are integral losses $l(t, x, \theta)$, and the integrand depends on θ **explicitly**, through u_t^θ .
- U2.T1's remark covers exactly this: the adjoint equation gains $-\partial l / \partial x$, **and** the gradient gains $\int_0^1 \partial l / \partial \theta \, dt$.
- It is not enough to add only a source term to the adjoint equation.

Nothing new is derived: the machinery planted in U2.T1 certifies the training of the regularized model.

What the penalties are for

- They control the speed of the field and the variation of its Jacobian along trajectories.
- A field that varies slowly is easier for a solver to follow, and in practice that can make a solve cheaper.
- “Cheaper” is an **empirical** result, not a theorem: (Finlay et al. 2020) report, on unregularized FFJORD flows during training on CIFAR10, a clear correlation between NFE and the Jacobian norm.

Hold on to straightness/easy-to-integrate dynamics as a virtue — it returns later.

05 | The table breaks

The two costs, version 3

THE TWO COSTS (V3) – U2.T2

Family	Density	Sampling	Trick, and its price
NICE / RealNVP	exact, triangular	direct	coupling; depth
Glow	exact	direct	mixing, multi-scale
MAF	parallel	sequential	AR, data side
IAF	sequential	parallel	AR, source side
Residual flow	direct; series log-det	iterative	contraction, traces
CNF (FFJORD)	ODE solve + divergence per call	ODE solve	free-form field; the price is the solve

Where the numerical work sits

WHERE THE NUMERICAL WORK SITS (V3)

Task	integrates	per RHS call	headline
sampling	$x, 0 \rightarrow 1$	u_t^θ	NFE
density, exact	$[x; \Delta], 1 \rightarrow 0$	u_t^θ and d VJPs	NFE; VJPs extra
density, Hutchinson	$[x; \Delta], 1 \rightarrow 0$	u_t^θ and 1 VJP	NFE; VJP extra
training step	solve; backprop or adjoint	as density, then differentiated	NFE, all solves

NFE is the headline count; **wall-clock is the honest total.**

The cliffhanger

MLE training of a CNF places an adaptive numerical solve inside each optimization step. Density-mode RHS calls include both u_t^θ and a divergence calculation; differentiation adds either stored-solver backpropagation or a second adjoint solve. NFE is data-, parameter-, tolerance-, and solver-dependent and may drift substantially during training. The cost of learning sits inside the innermost loop.

The question

**Can we obtain u_t^θ by regression —
without ever simulating during training?**

Housekeeping, and the last sentence

- PS1 is due and PS2 is released with this session (see the calendar-map). PS2.1 and PS2.2 are live; PS2.5 rides with the next lab, U2.L2.
- The asynchronous package (the U2.L2 lab, the clinic recording, the Q&A thread) is released per the calendar-map.
- The clinic recording is **required viewing before U3.T1**.

When we meet in U3.T1, you will have felt what this question is worth.

References

- Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. “Neural Ordinary Differential Equations.” In *Advances in Neural Information Processing Systems*. <https://arxiv.org/abs/1806.07366>.
- Finlay, Chris, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M. Oberman. 2020. “How to Train Your Neural ODE: The World of Jacobian and Kinetic Regularization.” In *International Conference on Machine Learning*. <https://arxiv.org/abs/2002.02798>.
- Grathwohl, Will, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. 2019. “FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models.” In *International Conference on Learning Representations*. <https://arxiv.org/abs/1810.01367>.