

Course Organization

Flow-Based Generative Models · UFRJ · 2026.2

01 | The course

One course, two programs

Flow-Based Generative Models: From Normalizing Flows to Flow Matching — a graduate-level course offered jointly to two programmes:

Program	Rhythm	What it includes
PGMAT	semester	DL Bootcamp (Uo) + main course
PESC	trimester	main course, joining at the first joint unit

- Two 90-minute sessions per week, alternating: **theory** (T), then **lab** (L).
- Format: theory with live proofs; scaffolded PyTorch labs; problem sets; seminars; a final project.

Course motivation

Ten years of flow-based generative modeling converge on a single simulation-free framework — Flow Matching. Diffusion models are a family of special choices inside it.

The course walks that convergence:

1. **Discrete flows (U1).** Exact likelihood by change of variables; the price is architectural.
2. **Continuous time (U2).** Neural ODEs lift the constraints — but training requires simulating the model.
3. **Flow Matching (U3).** Train the vector field *without* simulating it. Diffusion enters as **detours** from this spine, not as a separate unit.
4. **Frontier (U6).** Student seminars: manifolds, discrete data, graphs.

02 | Reading the map

Every session has an ID

SESSION ID

A session is named `Unit.Kind#` — unit, kind of session, number within the unit. Example: **U3.T2** = second **t**heory session of unit **U3**.

Kind	Meaning	Typical slot
T	theory	first slot of the week
L	lab	second slot of the week
S	seminar	(seminar phase)
W	workshop	(project phase)
A	async document	no live slot

IDs are **stable names**; dates live in a separate calendar layer.

The unit map

Unit	Name	Sessions	Audience
U0	Deep Learning Bootcamp	5 T + 5 L	PGMAT only
UP	Probability for Gen. Models	1 A + 1 T	joint
U1	Discrete Normalizing Flows	2 T + 2 L	joint
U2	Neural ODEs and CNFs	2 T + 2 L	joint
U3	Flow Matching (+ detours)	4 T + 3 L	joint
U5	Scaling and Ecosystem	1 T	joint
U6	Student Seminars	4–5 S	joint
U7	Project and Synthesis	2 W	joint

Small print: **there is no U4**. The numbering jumps from U3 to U5 — a deliberate ruling; session IDs are permanent names, so units are never renumbered.

03 | The units

Uo — Deep Learning Bootcamp (*PGMAT only*)

- **Uo.T1–T5 / Uo.L1–L5:** supervised learning and autodiff; making training work; latent variables and the VAE; architectures for the road (U-Net, attention, time embeddings) and evaluation; numerical ODE solvers.
- The bootcamp builds the **artifacts the whole course reuses**: the training-loop template, the evaluation harness, the U-Net skeleton, the solver notes.
- **PSo** closes the bootcamp — implementation-only, graded on correctness and reproducibility.
- **PESC equivalent:** a CPU-friendly **self-study packet** (ODE solver notes + JAX primer + pointers into UP.A), released at enrollment; PS1 draws on it.

UP — Probability for Generative Models

- **UP.A** — an async, self-contained primer: densities and conditioning, Gaussian algebra, divergences, sampling. Its embedded exercises are graded (they *are* PS1.1).
- **UP.T1** — the first live joint session: **“the Lemma of the course”** — conditional expectation as L^2 projection, proved once in full.
- The Lemma returns three times in U3; the session ends by announcing exactly where. Watch for that slide again.

U1 — Discrete Normalizing Flows

- **U1.T1** change of variables and coupling flows · **U1.L1** coupling flows on 2D toys (the first joint lab) · **U1.T2** the zoo and its limits · **U1.L2** flows on images.
- **The parity promise, stated once and kept everywhere:** no joint session requires having attended U0. Bootcamp artifacts arrive as *given code* with a contract-level description.
- Problem sets start carrying **proof-completions:** results stated in lecture, finished by you.

U2 — Neural ODEs and CNFs

- **U2.T1** Neural ODEs and the adjoint method · **U2.L1** solvers and adjoints in practice · **U2.T2** continuous normalizing flows and FFJORD · **U2.L2** FFJORD in 2D: feel the cost (at-home lab).
- The unit ends on the course's central cliffhanger: *can we train the vector field without simulating the ODE?*

U3 — Flow Matching, with diffusion detours

- **U3.T1–T4** build Flow Matching in full; **U3.L1–L3** implement it: CFM from scratch, diffusion as a special case of your own code, couplings and guidance.
- Diffusion is never a separate unit — it enters through three mapped **detours (D1–D3)** attached to the FM spine.
- The **Rosetta Stone**: a living two-column handout translating FM dialect and diffusion dialect. It grows with each U3 session and ships, complete, with the seminar kit — it is what lets you read the literature in either dialect.

U5 and U7 — Scaling; Project

- **U5.T1** — one session on scale: DiT, latent flow matching, few-step generation, the ecosystem (Stable Diffusion 3, Flux, Meta's FM codebase). Feeds directly into project scoping.
- **U7.W1** — project workshop: bring a one-page sketch, get structured peer feedback, freeze the plan (scope may shrink afterwards, never grow).
- **U7.W2** — synthesis: the course arc retold backwards, and the frontier handoff.
- **Final project:** extend a course lab, or apply Flow Matching to your own research data. Graded on correctness, scoping, and honesty — never on results quality.

U6 — Student Seminars

- Three tracks: **A** manifolds · **B** discrete data · **C** graphs. Two talks per session.
- Per talk: 30–35 min presentation + 10 min discussion, opened by an assigned **discussant** with two prepared questions.
- Every talk's **second slide is a translation slide**: the paper's notation mapped onto course canon. Translating is the learning objective.
- Each talk comes with a **reproduction notebook** at toy scale — with an honest “what is NOT reproduced” statement. A classmate should run it end-to-end in under 30 minutes.

04 | Logistics

Assessment (*weights indicative*)

Item	Weight	Covers
PSo (<i>PGMAT only</i>)	folded into PGMAT grade	bootcamp implementation
PS1	~12%	UP + U1
PS2	~13%	U2 + early U3
PS3	~13%	late U3
Seminar + notebook	~27%	U6: talk, discussant, notebook
Final project	~35%	plan + report + code

Implementation work is graded on **correctness and reproducibility** — seeded runs, logged configs — never on sample quality or benchmark glory.

Materials and tools

- Every session comes with **lecture notes** to read, **slides** to follow in class, and a **printable PDF** of the deck — all on the course site.
- Labs: **PyTorch** primary, JAX for illustrative moments; experiment logging with **MLflow**; everything seeded and configured for exact reruns.
- The house rule: every core object is implemented **from scratch** at least once; libraries (Zuko, TorchCFM, torchdiffeq, ...) are cross-checks, never substitutes.
- Standing references: Lipman et al., *Flow Matching Guide and Code*; Bishop & Bishop, *Deep Learning*; the full library lives in the course bibliography.

Async weeks

A few weeks run **async by design** — a format, not an improvisation:

- a scaffolded notebook with “you should see” checkpoints, plus a checkpoint package (weights + config + seed) so nobody is blocked by compute;
- a **shared Q&A thread**, with blocking questions answered in under 24 hours;
- when a proof matters that week, a **recorded proof clinic** replaces the live derivation.

The rule behind it: **absence weeks are converted, not lost.**

05 | Calendar

Start and end

- **PGMAT:** the course opens **Aug 18** with Uo.T1.
- **PESC:** the trimester window opens **Sep 21**; that week is the async onboarding week (self-study packet + UP.A + PS1 released, recorded welcome lecture).
First live joint session: **Sep 29** — UP.T1, the Lemma.
- Last session: **Dec 17** (U7.W2, synthesis). Final project report target: **Dec 19**.
- No holiday this term falls on a class day — zero sessions lost to holidays.

When I will be away

Three conference windows overlap the term:

Window	Dates	What happens instead
CBCTQ	Sep 21–25	the async onboarding week above
CILAMCE	Oct 26–29	async lab week + recorded proof clinic
Conference #3	TBD	two slots held in reserve

No content disappears: labs go async, seminars run with a delegate chair, theory moves only as a last resort — never out of order.