

Yuefeng Yin | Teaching Statement

I teach the way I do research: **students should learn to ask good questions and to judge whether an answer is right**, not only to follow a procedure. I designed and delivered a complete university course in *Computational Materials Science* (2017–2019, 2023–2024), run density-functional-theory training for Master's and PhD students, and have mentored nine PhD candidates. AI makes that judgement more important, not less: when a model can write the code or the derivation, the engineer's job is to know when it is wrong.

Teaching experience

Designing and delivering a unit. Within research appointments, I took full responsibility for a unit of study. For the Monash–Central South University "2+2" program I designed *Computational Materials Science* from first principles and taught it in 2017–2019 and 2023–2024: curriculum, lectures, tutorials and assessment. In three intensive weeks it takes students from statistical sampling through density functional theory and high-performance computing to machine learning, ending in an **assessed group mini-assignment** that makes them combine all three (structure, right). The full materials are openly available on [GitHub](#).

Research training. At Monash I run **DFT workshops and tutorials for Master's and PhD students** across research groups. I teach students to select, justify and validate methods for their own research questions, using visualisation, worked examples and physical reasoning to make abstract concepts accessible.

Supervision. I have mentored or co-supervised **nine PhD candidates** (five graduated), supervised three Bachelor's and Master's projects and examined three PhD theses. Mentees have first-authored papers in *Advanced Science*, *ACS Nano* and *Nano Letters*; one now holds a joint postdoctoral position at Cambridge and King's College London.

How I teach

Question first, method second. Each topic opens with an engineering question — *why does this transistor leak? which material should we grow?* — and the theory arrives as the tool that answers it.

Make the abstract visible. I use interactive visualisations, worked examples and live calculations, so that band structures, wavefunctions and transport become something students can see and change rather than only read.

Practise judgement. Students compare methods, estimate before they compute and check answers against limits, units and experiment, so they leave able to defend a result as well as produce one.

Learn by building. Assessment centres on open-ended, team-based design problems with design reviews, mirroring how engineering work is actually done.

1
unit designed and delivered

5
years delivered (2017–19, 2023–24)

9
PhD candidates mentored

3
UG / Master's projects supervised

3
PhD theses examined

THE UNIT I DESIGNED

- WEEK 1**
Foundations & Monte Carlo
What computational materials science is; hands-on Monte Carlo
- WEEK 2**
Atomic simulation & DFT
What DFT can and cannot do; linking to experiment
- WEEK 3**
DFT, HPC & AI
Supercomputing, methods beyond DFT, machine learning
group mini-assignment

Computational Materials Science, Monash–CSU "2+2" program: an intensive 3-week structure.

Teaching with AI

AI is already changing how my field works, and my teaching should prepare students for that rather than pretend otherwise. My approach fits the University's two-lane model of assessment (right).

Use AI openly, and check it. In tutorials and projects students may use AI assistants for code, literature and derivations, but every AI-produced result must be **verified** — against a limiting case, a conservation law, a hand estimate or an experiment — and students keep a short **AI-use log** recording what they asked, what came back and what was wrong.

Secure the core skills. Oral design checks and in-class derivations assess what students understand without help, so the open work builds on a foundation they genuinely own.

Teach AI as engineering content. Students learn where materials AI comes from — data provenance, surrogate models, uncertainty — and why it fails outside its training data: the same questions my research asks.

Units I can contribute to

I can teach and coordinate units in **engineering materials, semiconductor and electronic materials and devices, computational and numerical methods for engineers, solid-state and quantum physics for engineers**, and **data-driven and AI methods for engineering design**, at undergraduate and postgraduate level. I would welcome coordinating a core unit early, and would develop *AI-Driven Materials and Device Design* (right) with colleagues across Schools as a senior undergraduate or postgraduate elective.

Supervision and inclusive practice

My supervision starts by helping students frame a question they care about, then trains the core skills rigorously while leaving room for their own ideas. I set clear expectations, give regular written feedback and build co-supervision across theory and experiment, so students see more than one way of working. In class I use varied formats — visual, written, oral and hands-on — and make assessment criteria explicit, so that students from different backgrounds, including international students joining through articulation programs like the one I taught, can succeed. At Sydney I would review each offering with student feedback and peer observation of teaching, and change what does not work.

TWO-LANE ASSESSMENT

LANE 1 · SECURE

What students own

Oral design checks and in-class derivations, without AI.

LANE 2 · OPEN

What students can do with AI

Projects and tutorials with AI allowed — every result verified, every use logged.

A UNIT I WOULD OFFER

WEEKS 1–2

Electrons in solids & semiconductors

WEEKS 3–4

First principles: DFT on HPC

WEEKS 5–6

From materials to devices

Band alignment, contacts, transport

oral design check (lane 1)

WEEKS 7–8

AI for design

Data, surrogate models, uncertainty, AI agents

WEEKS 9–12

Team design project

Discover and justify a material for a device

project + AI-use log (lane 2)

AI-Driven Materials and Device Design (proposed, 12 weeks).