ALD Growth on Polymer Materials

Undergraduate research · Lee Research Group, Stony Brook University · August 2026 to present

I work in a two-person polyurethane thin-film team. We spin-coat films, run ALD and vapor-phase infiltration on them at a national laboratory, and characterize the results.

01 Developing the spin-coating recipe

When film quality fell short of spec, I designed a 3x3 grid of concentration and spin speed and used it to seed a Bayesian-optimization loop. We reached a 100 nm target with under 5% nonuniformity, checked by 5-point ellipsometry.



The Lee Group ALD system used in my research. Edited photo.

02 Deposition and characterization

We run ALD and vapor-phase infiltration with TMA/water and DEZ/water on polyurethane films and textiles, comparing flow-through and hold recipes. Our group found the hold recipe gave about 11 times more mass gain. I follow precursor uptake with in-situ QCM and characterize films with ellipsometry, ATR-FTIR, XRD, SEM, and STEM.

03 Lab tools and partner work

I built Crystal Tools, a Python pipeline that cleans instrument exports and compares QCM runs across conditions. I also prepared ASTM D412 tensile specimens, 25 to 30 dog-bones across 5 to 6 ALD conditions, for mechanical testing by an industry partner. Unpublished data is not shown here.

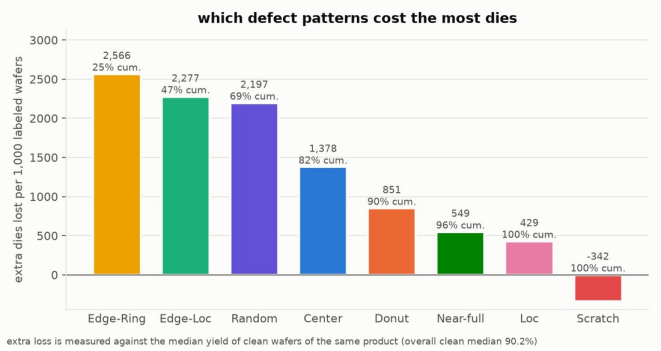
WM-811K Wafer Map Yield Analysis

Independent project · Public WM-811K dataset · October 2026 to present · github.com/mehnajjimy/wm811k

Using the public WM-811K wafer maps, I worked out which defect patterns cost the most dies and whether they come from single wafers or whole lots.

01 Ranking patterns by die loss

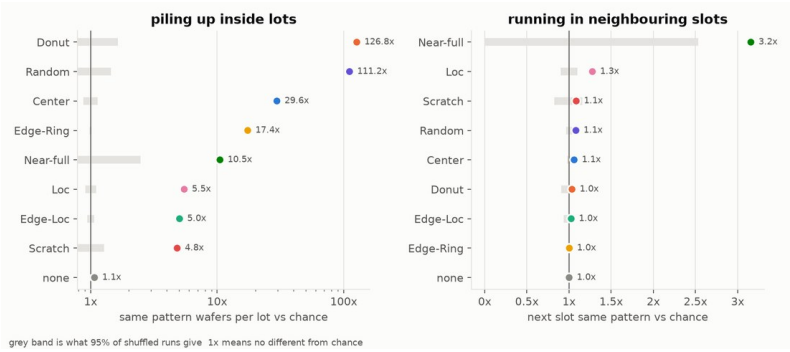
I measured extra die loss against clean wafers of the same map size across 172,950 labeled maps. Edge-Ring and Edge-Loc account for 47% of the excess loss, so edge-related process conditions come first.



Extra dies lost per 1,000 labeled wafers, with cumulative share above each bar.

02 Lot-level clustering

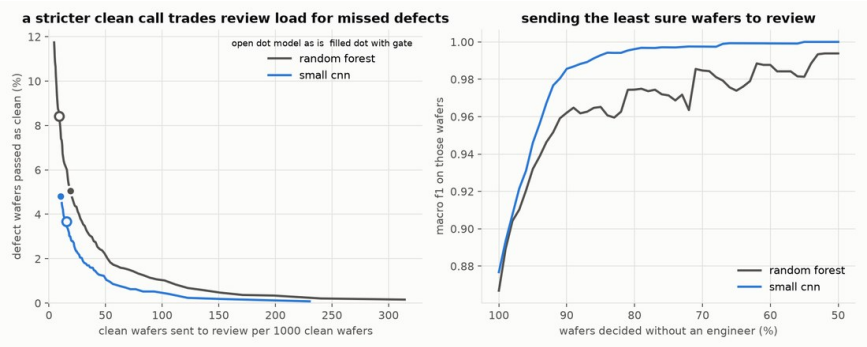
All 8 defect patterns cluster within lots far above chance (200 label shuffles, p of 0.005 or lower). 502 of 7,025 lots are excursion lots, 361 of them Edge-Ring, which points to lot-level causes rather than isolated wafer events.



Left: same-pattern wafers per lot against chance. Right: next-slot repetition.

03 Classifiers with a review gate

On a lot-disjoint split, a small CNN reached 0.877 macro-F1. Holding missed defects near 5%, it sends 10.3 clean wafers per 1,000 to review, against 19.0 for a Random Forest on 24 engineered features.



Review load against missed defects for both models on the test set.

Mehnaj Jimy
mehnajjimy.com · mehnajjimy@gmail.com

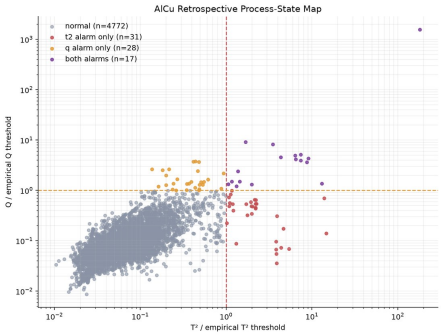
PVD Process Monitoring & Virtual Metrology

Independent project · Public Infineon AlCu and WTi data · August 2026 ·
github.com/mehnajjimy/semiconductor-pvd-process-monitoring

I used Infineon's public PVD release to flag multivariate process excursions in 97-channel equipment sensor data and predict 17 film-thickness metrology targets.

01 Flagging excursions

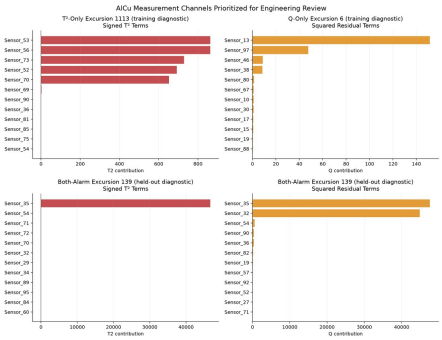
PCA-based SPC (Hotelling T^2 and Q/SPE) compares each record with the training distribution. An alert is a prompt for review, not a confirmed fault.



AlCu process-state map: all records, training and held-out.

02 Ranking the channels

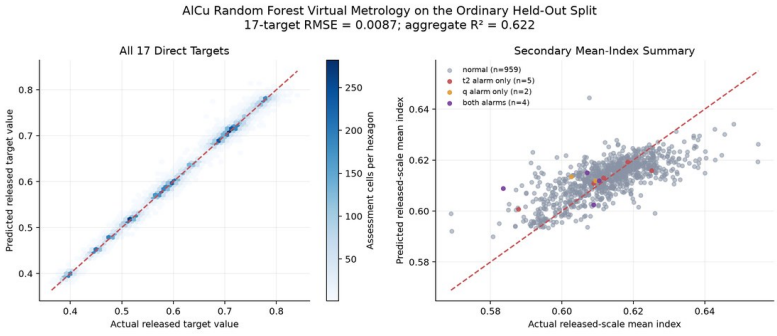
Contribution analysis ranks the anonymous sensor channels behind each excursion, which sets the order for root-cause review.



Contribution plots for flagged AlCu excursions.

03 Virtual metrology

A Random Forest predicted all 17 targets (held-out R^2 0.622 AlCu, 0.604 WTi) and cut RMSE 41 to 43% against Ridge regression. Keeping identical target profiles together gave 0.543 and 0.690.



AlCu ordinary-split predictions on the released scale.

Mehnaj Jimy
mehnajjimy.com · mehnajjimy@gmail.com

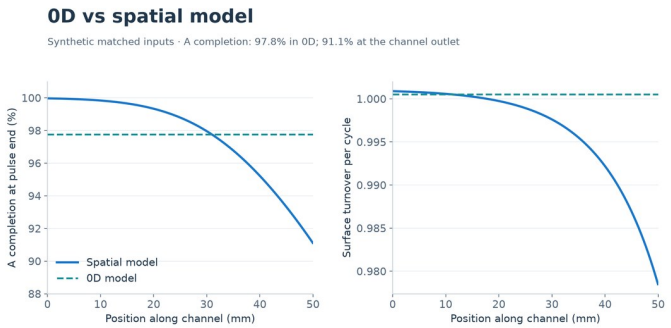
Physics-Based ALD Reactor Model

Independent project · Python modeling · July 2026 to present · github.com/mehnajjimy/ald-reactor-digital-twin

My Python model follows precursor transport and surface reactions through an ALD cycle and compares a well-mixed reactor with a spatial channel model.

01 Where the models differ

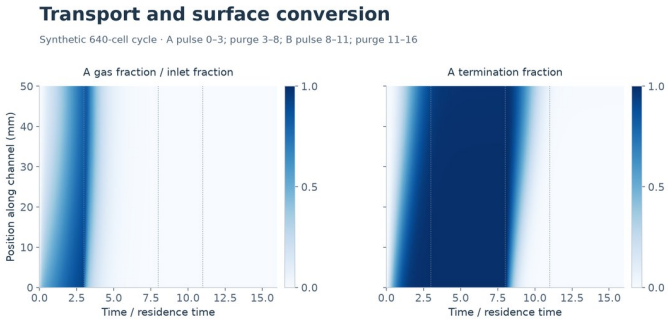
On a synthetic recipe, the well-mixed model predicts 97.8% completion everywhere, while the spatial model falls to 91.1% at the outlet. The well-mixed assumption hides under-dosing near the outlet.



Matched well-mixed and spatial predictions. Synthetic example.

02 Transport along the channel

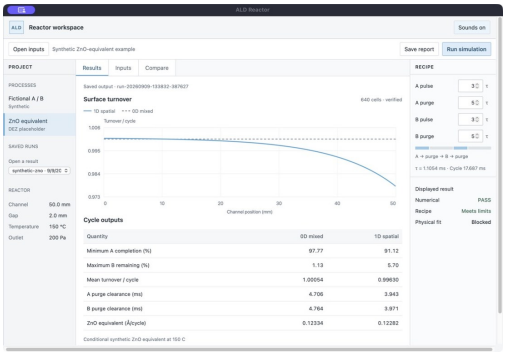
The spatial model shows the gas clearing after the A pulse while the surface keeps its A termination until the B exposure.



Gas fraction and surface termination over one synthetic cycle.

03 App and checks

A desktop app and command-line tool share one solver, with 22 automated tests. The DEZ/water ZnO case predicts about 1.68 Å per cycle against 1.65 Å reported at 150 °C. No example has been checked against experiments yet.



Desktop app showing a saved synthetic result.

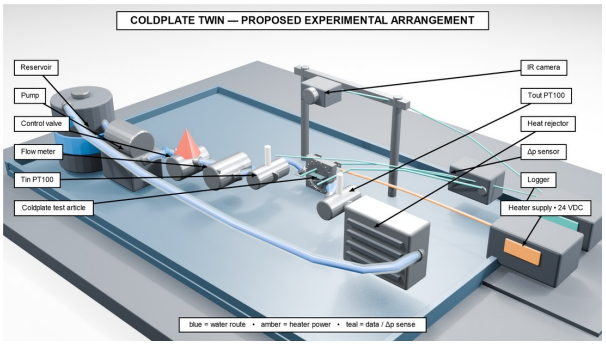
ColdPlateTwin: Water-Cooled Testbed Design

Independent project · Preliminary CAD and synthetic simulations · September 2026 to present · github.com/mehnajjimy/ColdPlateTwin

I’m designing a water-cooled cold-plate testbed with interchangeable flow inserts, using CAD and Python to examine heat balance and pressure losses before physical testing.

01 The proposed experiment

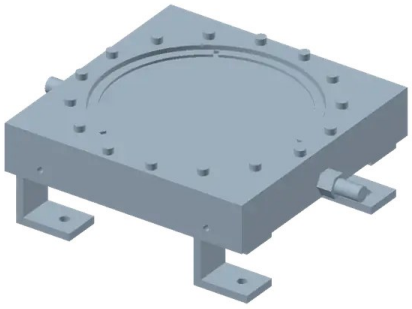
A water loop connects the cold plate to flow, temperature, and pressure measurements. The apparatus is still a proposed design.



Repository render. Equipment layout is schematic, not an as-built photo.

02 The CAD assembly

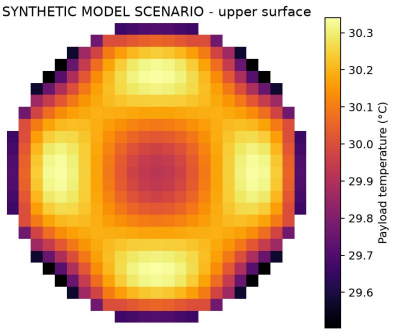
The preliminary geometry supports interchangeable inserts. Machining approval and pressure rating are still ahead.



Preliminary Rev A CAD. Not built hardware.

03 Analysis workflow

Python tools screen steady periods in logger data, then compute heat transferred to the water, energy balance, pressure drop, and uncertainty.



Synthetic surface-temperature map. Not a measurement or CFD result.

Earlier Modeling Projects

Independent Python projects · 2026 · github.com/mehnajjimy

Two smaller chemical engineering models in Python, each checked against its own balances or exact solutions.

01 Heat and Mass Exchange Simulator

The heat calculation uses the effectiveness-NTU method for a counter-current exchanger, and the membrane calculation uses an exponential concentration model with NTU = KA/Q. Energy and solute balance residuals are saved with every case. The two calculations share inputs but are not coupled.

02 Batch Reactor Simulator

I solve zero-, first-, and second-order batch kinetics with SciPy and compare each case with its analytical solution. Over 500 s, the largest concentration errors in the saved cases are 1.11e-16, 3.57e-4, and 8.30e-4 mol/L. This verifies the numerics, not a real reactor.