Aryan Goyal

A short presentation of my previous work,
my research interests, and a bit about me

Indian Institute of Technology Bombay

A Bit About Me



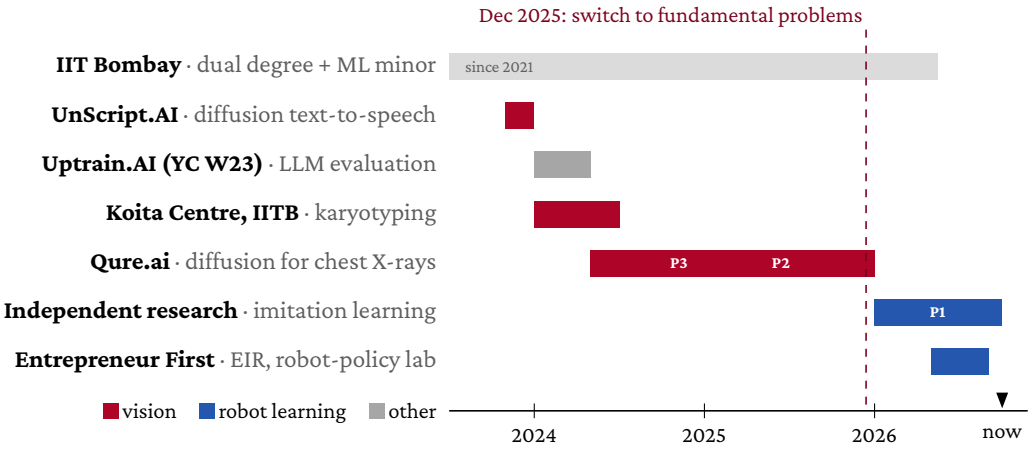
Aryan Goyal
garyan18@gmail.com
aryangoyal7.github.io
X: @arygoy

- **IIT Bombay** (2021–2026): Dual Degree, B.Tech + M.Tech in Environmental Science & Engineering, with a Minor in Machine Learning & Data Science (C-MInDS)
- **Carnegie Mellon**: admitted to the M.S. in Computer Vision, Robotics Institute
- **3+ years of AI research**: started in computer vision with diffusion models, moved into imitation learning in 2026

North star: a generalist robot policy. My current focus is imitation learning.

Outside research: tennis, pour-over coffee, and making playlists for friends.

How I Got Here



Roadmap

1. The stability assumption behind action chunking (independent)
2. Controllable lung nodule synthesis (Qure.ai)
3. Super-resolution of chest radiographs (Qure.ai)
4. What's next

Measuring the Stability Assumption Behind Action Chunking.

arXiv:2610.01626 · independent research

Why Does Action Chunking Work?

A chunked policy predicts k actions, executes them without looking, and then replans. This is widely reported to outperform single-step execution, and seven mechanisms have been proposed to explain why:

- Temporal consistency — Zhao '23; Chi '23
- Horizon reduction — Foster '24
- Representation learning — Chi '23; Torne '25
- Non-Markovian expressivity — Lazzati '26
- Delayed conditioning — Lazzati '26
- Implicit ensembling — Lazzati '26
- Substitute for missing context — Zeng '26

Every one of these is about the error that **enters** the loop: how often the policy makes one, or how large it is. None of them measures what happens to the error **afterwards**, that is, whether the dynamics compress it or expand it over the rest of the horizon.

Zhao et al., RSS 2023 · Chi et al., RSS 2023 · Foster et al., NeurIPS 2024 · Torne et al., 2025 · Lazzati et al., 2026 · Zeng et al., 2026

The Assumption Behind the Theory

The guarantees for chunking assume the true dynamics are open-loop *exponentially incrementally input-to-state stable* (EISS) with one fixed contraction factor $\rho \in (0, 1)$:

$$\|x_{t+1} - x'_{t+1}\| \;\le\; C_{\mathrm{ISS}}\,\rho^{\,t}\,\|x_1 - x'_1\| \;+\; C_{\mathrm{ISS}} \sum_{s=1}^{t} \rho^{\,t-s}\,\|u_s - u'_s\|$$

So any deviation shrinks by a factor of ρ per step, at every state. Under this assumption, long enough chunks restore stability of the learned closed loop, and weaker contraction ($\rho \to 1$) needs longer chunks:

$$k \;\ge\; k^{\star} \;=\; \left(\log \frac{1}{\rho}\right)^{-1} \log \mathrm{poly}\left(L,\, C_{\mathrm{ISS}}\right)$$

Here ρ is assumed and global, never measured, and Lazzati et al. (2026) likewise assume Lipschitz dynamics. Yet an arm in free space and the same arm in contact need not propagate an error at the same rate.

EISS: Simchowitz et al., COLT 2025 · Bound: Zhang et al., arXiv:2507.09061 · Lazzati et al., arXiv:2608.02547

Two Kinds of Stability

Once an action error has entered, only two things can deal with it, and they correspond to the choice the executor makes at every step:

Open-loop stability (longer chunk)

A property of the **plant**: the dynamics and task geometry. The rest of the chunk is replayed unchanged after the error. The state is stable if the deviation shrinks. No policy is involved.

Closed-loop stability (shorter chunk)

A property of the **pair** $(\hat{\pi}, f)$: the learned policy with the true dynamics. The policy replans at every step after the error. The state is stable if replanning shrinks the deviation.

The expert pair $(\pi^\star, f)$ is stable by construction, since a human demonstrator sees the error and corrects it. That does not carry over to $(\hat{\pi}, f)$, so closed-loop stability has to be measured on the trained policy.

Zhang et al., arXiv:2507.09061, 2025

Measuring It State by State

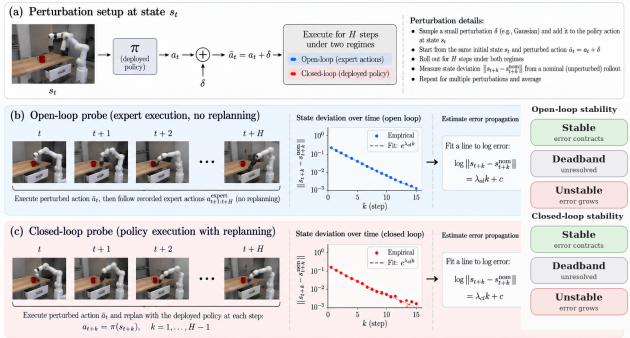


Figure 1: Open-loop probe (b) replays the recorded chunk; closed-loop probe (c) replans at every step. 12 tasks across robomimic, MimicGen and RoboCasa.

1. At state s_t, perturb the first action, $\tilde{a}_t = a_t + \delta$, at the scale of the policy's own action error.
2. Roll out a nominal and N=8 perturbed branches. In closed loop, each pair shares the policy's random seeds.
3. Fit the log-deviation over a window of K steps:

$$\log \left\| y_j - y_j^{(n)} \right\| \approx \alpha_n + \lambda_n \, j$$

   $\lambda > 0$: error grows; $\lambda < 0$: it shrinks.
4. Label unstable or stable only if $|\bar{\lambda}| \geq \ln(m)/K$ (an m-fold change, m=2) and $|\bar{\lambda}| \geq t \cdot \mathrm{SE}$; otherwise deadband.

What the Labels Look Like

Each clip starts at one labeled state of **tool hang**, nudges the first action, and plays the next $K=24$ steps, exactly the window the label is measured over. The white line is the nominal end-effector path; the coloured lines are the $N=8$ nudged copies.

UNSTABLE $\bar{\lambda} = +0.330$



The nudged copies **fan out** away from the nominal path.

demo 108, $t=420$ ▶ clip

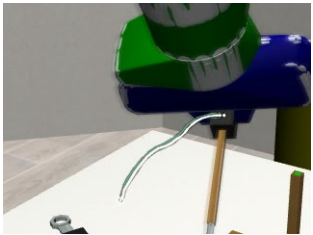
DEADBAND $\bar{\lambda} \approx 0$



The copies neither pull back nor clearly spread.

demo 181, $t=34$ ▶ clip

STABLE $\bar{\lambda} = -0.151$

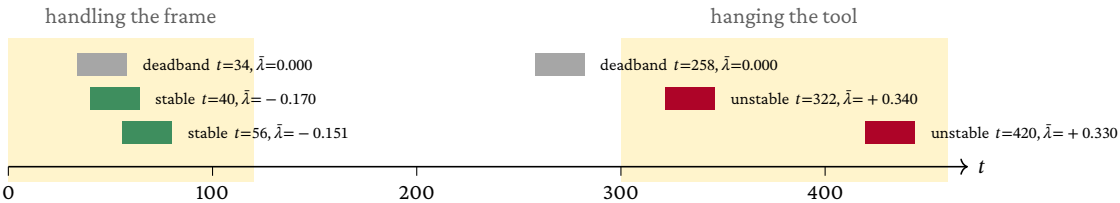


The copies **pull back** toward the nominal path.

demo 187, $t=56$ ▶ clip

An Instant, Not an Episode

$\bar{\lambda}$ is fitted only over the $K{=}24$ steps after the nudge, so a label says how the **next few instants** respond to an error, not how the whole episode behaves. One task passes through all three regimes:



Six labeled tool hang states from different demonstrations, placed on a common episode clock. Each bar is one measurement window of $K{=}24$ steps.

The stability assumption can hold at one moment of a task and fail at another, so a single global ρ cannot describe both.

Result: Rates Across Twelve Tasks

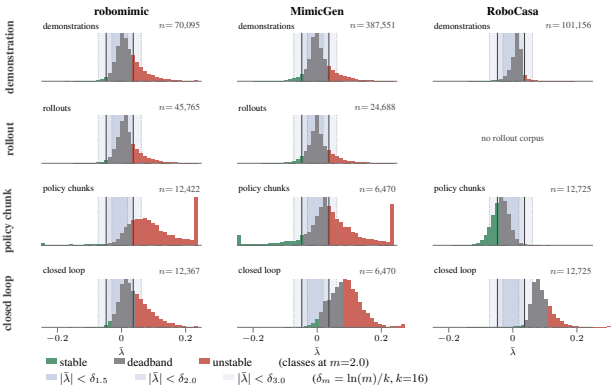
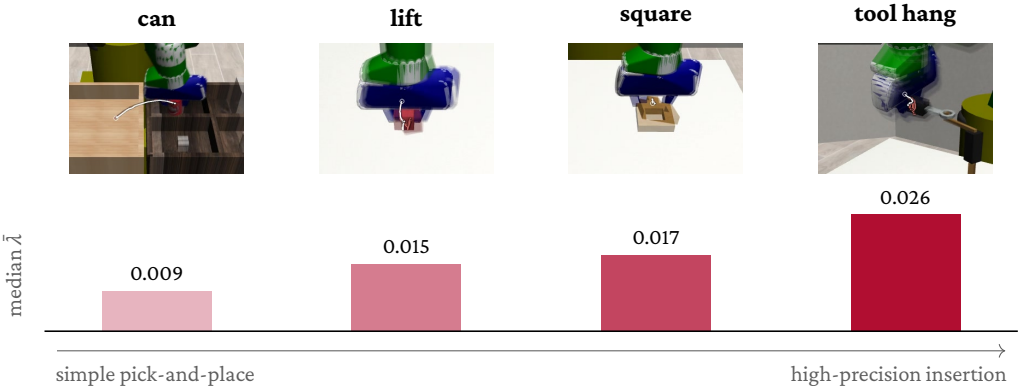


Figure 2: Measured rate $\bar{\lambda}$ per state. Rows: demonstration, rollout, policy-chunk, closed-loop. Dotted: $\pm \ln(m)/16$.

- One panel per suite and state type: green, gray and red are stable, deadband and unstable.
- Most mass sits at or right of zero; the median open-loop rate is positive everywhere, so an error typically grows.
- Policy chunks lie further right: their short window ($K = 8$) captures the early, fast growth.
- Closed-loop rates rarely reach the stable region.

Harder Manipulation, Larger Rates

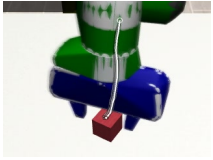


Across robomimic demonstrations the median rate rises with more contact-sensitive, precision-demanding manipulation. This is a trend, not proof that difficulty itself causes it.

Where Each Label Shows Up

DEADBAND

Most states

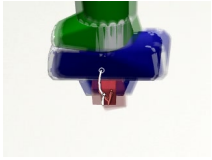


Near-zero rates in free-space reaching and transport: lift approaching the cube, the can or nut being carried.

lift, t=6 ▶ clip

UNSTABLE

Most of the resolved states

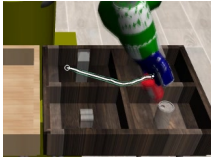


Amplification around grasps, insertions and placements: lift at the grasp, can placement, tool hang at the hook.

lift, t=18 ▶ clip

STABLE

Rare: 0–6% of states



The clearest contraction comes with the object already held, as in can and square mid-carry.

can, t=100 ▶ clip

Deadband is **not** stability: the rate is too small to cross the threshold (78% of lift states have $|\tilde{\lambda}| < 0.029$). Phases read off the most extreme clips per task, not a statistic.

Result: Open-Loop Contraction Is Rare

	$K = 24$			$K = 8$		
Demonstrations	S	D	U	S	D	U
lift	0.0	78.9	21.1	0.0	30.3	69.7
tool hang	1.4	67.4	31.3	1.2	33.2	65.6
coffee (MG)	5.5	76.1	18.4	5.2	52.8	42.0
OpenDrawer (RC)	2.5	93.2	4.3	3.4	19.3	77.3

Table 1: Label shares (%) on the same states at two fitting windows. S/D/U = stable, deadband, unstable.

- Confidently stable states are rare on every task, including the chunks the policy itself plans.
- Most states sit in the deadband because their rate is small, not noisy. Among resolved states, unstable dominates.
- A shorter window turns deadband into unstable: growth is **front-loaded**, fast at first, then flat.

Takeaway: the plant rarely absorbs a small error on its own, and the rate depends on the state and on how long you watch. One global ρ does not describe these tasks.

Result: Replanning Rarely Contracts

open-loop label \ closed-loop label	S	D	U
S	0.06	0.95	0.55
D	0.30	19.59	11.94
U	0.18	33.97	32.45

Table 2: Joint labels, robomimic policy states (%), $m = 2$. Red: open-loop unstable, closed-loop stable.

- The closed-loop stable class is tiny: 3 to 88 states per task at $m = 2$, against hundreds that are not stable.
- Errors that grow open loop but shrink under replanning are only 0.12–0.29% of states across the three suites.
- Replanning often removes *confident* growth (U→D) but rarely turns it into contraction.

Takeaway: behaviour cloning teaches little about recovery once the robot drifts, so train it explicitly with perturbation injection (DART; Simchowitz et al., 2025).

Can the Robot See Its Regime?

If open-loop stability belongs to the state, it may be visible. A frozen V-JEPA 2 encoder reads recent camera frames, a small recurrent net reads proprioception, and a head predicts λ and the unstable class. **P** sees frames and proprioception; **A** also sees the actions.

		Open-loop			Closed-loop		
Task	In	MAE_0	MAE	ρ	MAE_0	MAE	ρ
lift	P	.079	**.054**	.76	.032	**.018**	.82
	A	.079	**.040**	.80	.032	**.016**	.84
tool hang	P	.089	**.069**	.63	.038	.039	.24
	A	.089	**.066**	.64	.038	**.035**	.33
coffee	P	.083	.084	.15	.040	**.031**	.44
	A	.083	**.084**	.22	.040	**.032**	.46

Table 3: Policy-replan states. MAE_0: constant-median baseline; ρ: Spearman correlation. Bold beats the baseline.

- **Open loop:** recoverable without a simulator (AUROC 0.63–0.94). It ranks states better than it estimates the rate.
- Actions help, since λ is defined for those actions.
- **Closed loop:** only partly recoverable, because it also depends on how the policy acts after the perturbation.

V-JEPA 2: Assran et al., arXiv:2506.09985, 2025

Controllable Lung Nodule Synthesis.

Qure.ai · MIDL 2026 · patent filed

Motivation and Related Work

- Nodules in chest X-rays come in various shapes, sizes and textures.
- Assembling datasets that represent all of them is hard.

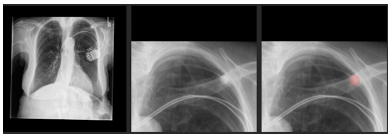
Prior nodule synthesis

- Shen, Ouyang, Xiao, Cheng, Shen & Wang. *Image Synthesis with Disentangled Attributes for Chest X-ray Nodule Augmentation and Detection.* Medical Image Analysis, 84:102708, 2022.
- Gündel, Setio, Grbić, Maier & Comaniciu. *Extracting and Leveraging Nodule Features with Lung Inpainting for Local Feature Augmentation.* MICCAI Workshops, 2021.
- Sogancioglu, Hu, Belli & van Ginneken. *Chest X-ray Inpainting with Deep Generative Models.* arXiv:1809.01471, 2018.

Characteristic Definitions (I)

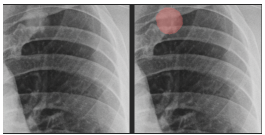
Homogeneous

Uniform density throughout the nodule.



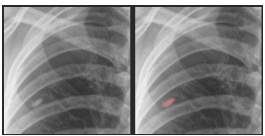
Inhomogeneous

Uneven density with mixed bright and dark regions.



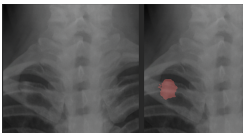
Regular

Smooth, well-defined, and clearly separated from lung tissue.



Irregular

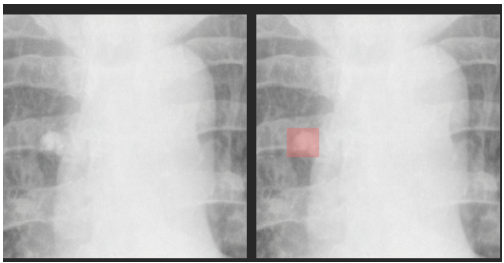
Spiculated, lobulated, or poorly defined margins.



Characteristic Definitions (II)

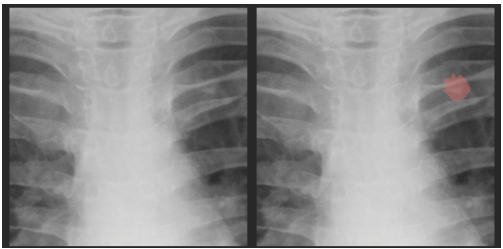
Calcification

Bright, high-intensity regions caused by calcium deposits, commonly seen in **benign nodules**.



Subtlety

Low-contrast nodules that blend with surrounding lung tissue, making them difficult to detect.



Method Overview

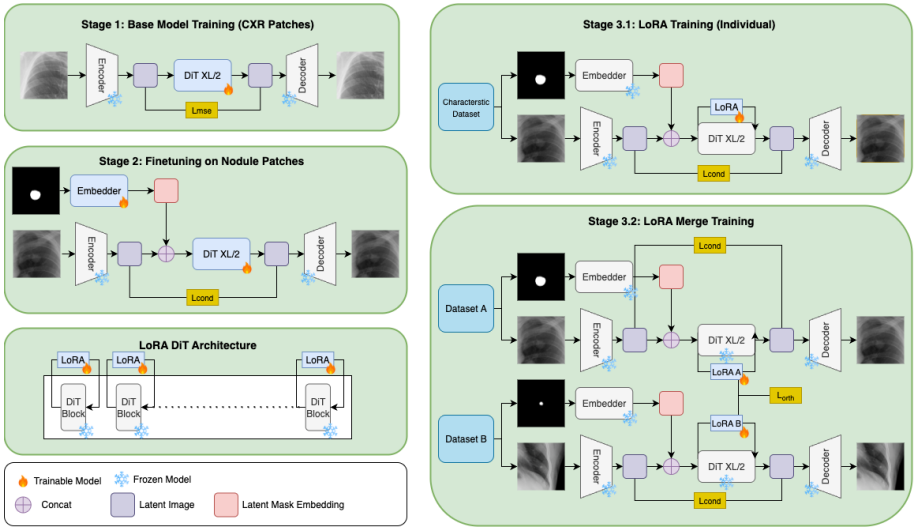
Components

- Mask-conditioned Diffusion Transformer backbone
- Characteristic-specific LoRA adapters
- Orthogonality-constrained adapter merging

Training proceeds in stages

1. Large-scale CXR pretraining
2. Nodule patch finetuning with spatial masks
3. Independent LoRA training per attribute
4. Joint training with orthogonality regularization

At a Glance


Stage 1: Base Model Training (CXR Patches)
Encoder
DiT XL/2
Decoder
Lmse
Stage 2: Finetuning on Nodule Patches
Embedder
Encoder
DiT XL/2
Decoder
Lcond
LoRA DiT Architecture
LoRA
LoRA
LoRA
DiT Block
DiT Block
DiT Block
Trainable Model
Frozen Model
Concat
Latent Image
Latent Mask Embedding
Stage 3.1: LoRA Training (Individual)
Characterstic Dataset
Embedder
Encoder
LoRA
DiT XL/2
Decoder
Lcond
Stage 3.2: LoRA Merge Training
Lcond
Dataset A
Embedder
Encoder
DiT XL/2
Decoder
LoRA A
Lorth
Dataset B
Embedder
Encoder
LoRA B
DiT XL/2
Decoder
Lcond

Mask-Conditioned Diffusion Backbone

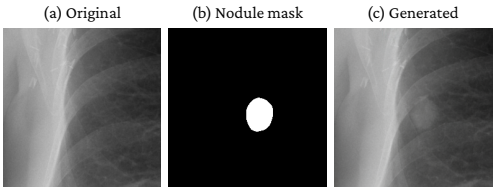


Figure 3: Given an original CXR patch and a binary nodule mask, the model generates a nodule within the masked region.

Forward diffusion process:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\, x_{t-1},\ \beta_t I\right)$$

Training objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0,t,\epsilon}\left[\, \|\epsilon - \epsilon_\theta(x_t, t, c)\|_2^2 \,\right]$$

Conditioning c includes binary nodule masks to enforce spatial control.

Characteristic-Specific LoRA Adapters

For a frozen backbone weight W_0:

$$W = W_0 + \alpha AB$$

where $A \in \mathbb{R}^{d\times r}, B \in \mathbb{R}^{r\times k}, r \ll \min(d,k)$, and α controls attribute strength. Separate adapters were trained for calcification, border regularity, texture homogeneity, and subtlety.

Subtlety via continuous LoRA scaling

Radiologist-annotated subtlety scores $s \in \{1,\dots,5\}$ were mapped as

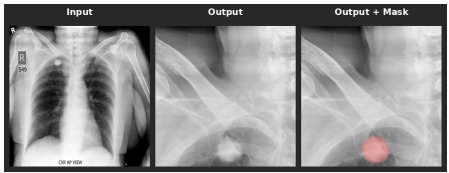
$$\alpha(s) = 22 + s$$

Lower α produces fainter nodules; higher α increases visibility. This gives an ordered perceptual progression and clinically meaningful difficulty control.

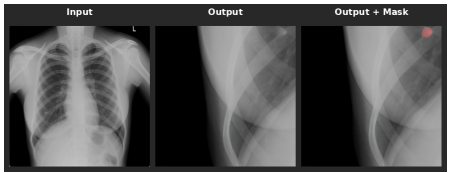
Adapters as controllable concepts: R. Gandikota et al., Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models, arXiv:2311.12092, 2023.

Examples: Texture and Boundary

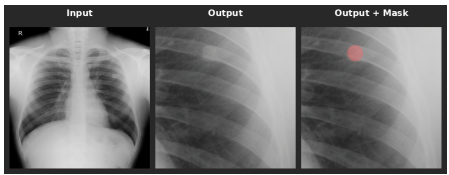
Texture: homogeneous



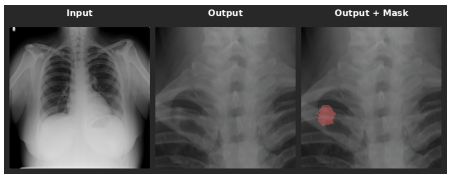
Texture: inhomogeneous



Boundary: regular

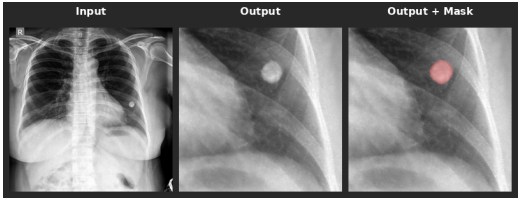


Boundary: irregular

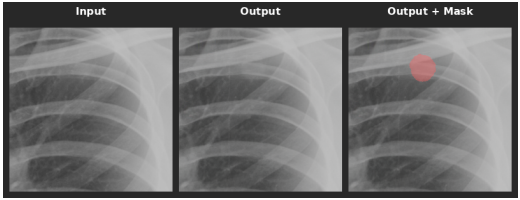


Calcified, Subtle, and Radiologist Checks

Calcified



Subtle



Task 1: Realism. 50 nodules (25 real, 25 from our base model) were shown to 3 expert radiologists. 90% of real nodules were identified as real, and **80% of synthetic nodules were also labeled real**.

Task 2: Controllability. 10 nodules per trait; 3 radiologists checked the intended trait was present.

Characteristic	Agreement (%)
Calcification	80
Regular border	90
Irregular border	100
Homogeneous texture	90
Inhomogeneous texture	100

Can We Control Combinations?

Nodule characteristics are *not mutually exclusive*: a regular nodule can be homogeneous or inhomogeneous while being subtle or calcified. This is where the design choice of LoRA pays off, since each characteristic-specific adapter can steer the same generation.

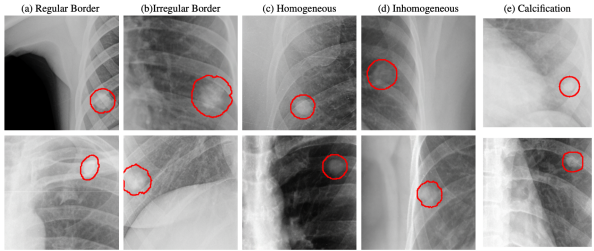


Figure 4: Results of characteristic-specific LoRA training, more than one characteristic.

Multiple adapters on one diffusion backbone: Gandikota et al., Concept Sliders, arXiv:2311.12092, 2023.

Bad Outputs: Merging Is Not That Simple

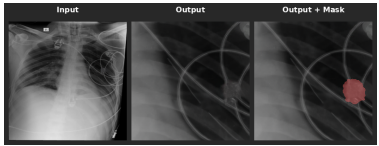


Figure 5: Looks very synthetic, not a good sample.

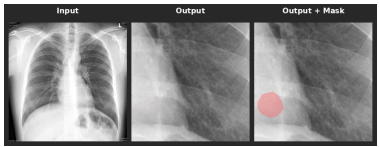


Figure 6: Calcified and homogeneous, but not calcified?

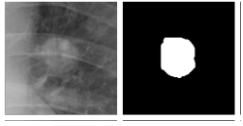
- Conventional merging strategies such as linear merge and switching assume that LoRA updates combine uniformly across layers.
- This can lead to one characteristic being suppressed by another, or to weird and nonsensical outputs.
- A key reason is that LoRA weights are highly sparse (60–70% near zero, $|w| < 0.01$), so a small subset of parameters drives most perceptual changes (K-LoRA; Ouyang et al., 2025).

Why Does This Happen?

Overlapping attention regions

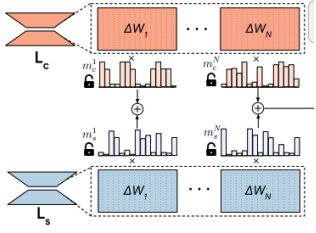


VS



This cannot be fixed in our case, unfortunately.

Non-orthogonal weights



This can be fixed.

Orthogonality-Constrained Merging

ZipLoRA, K-LoRA and Decouple-and-Orthogonalize all resolve LoRA conflicts **at merge time**. We instead constrain the adapters **during training**, because naive merging interferes through correlated updates:

$$\mathcal{L}_{\text{orth}} = \|W_1 W_2^{\top}\|_F^2$$

This encourages decoupled parameter subspaces, stable multi-attribute composition, and reduced spatial competition.

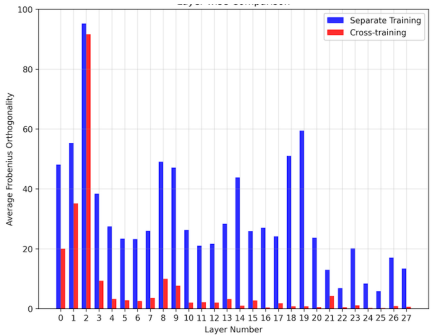


Figure 7: Layer-wise Frobenius norm, cross-trained vs. separately trained adapters.

Shah et al., ZipLoRA, arXiv:2311.13600, 2023 · Ouyang et al., K-LoRA, arXiv:2502.18461, 2025 · Zheng et al., Decouple and Orthogonalize, arXiv:2505.15875, 2025

Qualitative Comparison

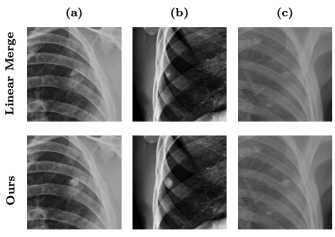


Figure 8: Linear merge vs. ours. (a) calcified + irregular, (b) calcified + homogeneous, (c) regular + homogeneous.

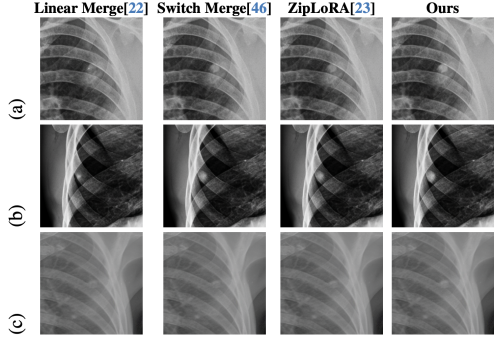


Figure 9: (a) calcified + irregular, (b) calcified + homogeneous, (c) irregular + homogeneous. Rows: configurations; columns: methods.

Downstream: Detection & Segmentation

	In-house		JSRT		ChestX-ray14	
Train data	AUC	IoU	AUC	IoU	AUC	IoU
10k real	0.9705	0.3090	0.8560	0.2475	0.9008	0.5285
10k real + 2k syn	0.9788	0.3222	0.8639	0.2743	0.9168	0.5293
10k real + 4k syn	0.9780	0.3197	0.8780	0.2589	0.9245	0.5500
10k real + 6k syn	**0.9802**	0.3247	0.8940	0.2894	0.9315	0.5750
10k real + 8k syn	0.9796	**0.3274**	0.8864	0.2923	**0.9341**	**0.5954**
10k real + 10k syn	0.9801	0.3056	**0.9023**	**0.3091**	0.9318	0.5613

Table 4: Detection (AUC) and segmentation (IoU) with added synthetic data.

- Adding synthetic data consistently improves performance across datasets, especially on **JSRT** and **ChestX-ray14**.
- Best performance is achieved with **6k–8k synthetic samples**, beyond which gains saturate or slightly regress.

Per-Characteristic Gains and Baselines

	Characteristic	5k real	+ 2k syn
AUC	Nodule	0.9531	**0.9647**
	Calcification	0.8338	**0.8454**
	Regular border	0.8955	**0.8994**
	Irregular border	0.6033	**0.6522**
	Homogeneous	0.7533	**0.7649**
	Inhomogeneous	0.7936	**0.8099**
IoU	Nodule	0.2696	**0.3002**
	Calcification	0.2879	**0.3199**
	Regular border	0.2941	**0.3301**
	Irregular border	0.2733	**0.3080**
	Homogeneous	0.2695	**0.3050**
	Inhomogeneous	0.2706	**0.2963**

Table 5: Swin-Tiny with a multi-head classifier, ∼400 synthetic nodules per characteristic.

Augmentation	JSRT	CXR14
10k real	0.8560	0.9008
+ 10k ACGAN	0.8780	0.9281
+ 10k ReACGAN	0.8808	0.9259
+ 10k CRFILL	0.8786	0.9296
+ 10k DiT-XL/2 (ours)	**0.9023**	**0.9318**

Table 6: Nodule AUC with other synthetic-data augmentations.

Augmentation improves both AUC and IoU on every characteristic, and beats GAN and inpainting baselines.

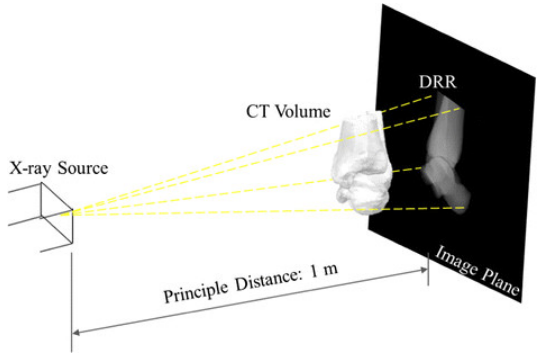
DiffusionXRay: Enhancing Chest Radiographs.

Qure.ai · DEMI @ MICCAI 2025

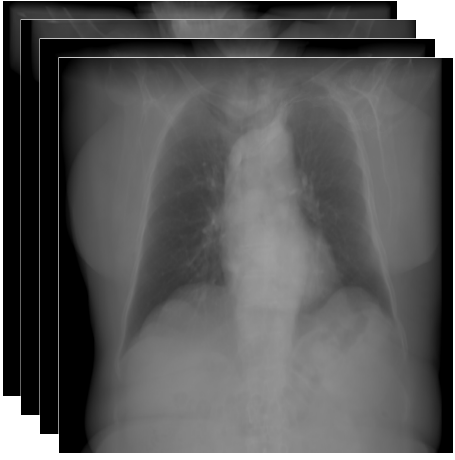
Motivation: DRRs Offer a Solution

- Early lung-cancer cues (subtle nodules) are hard to see on CXRs, and labeled datasets are scarce.
- DRRs (CT-projected X-rays) are scalable, but suffer blur, loss of fine structures, and artifacts.

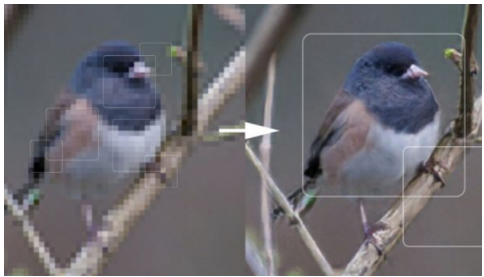
DRR = Digitally Reconstructed Radiograph



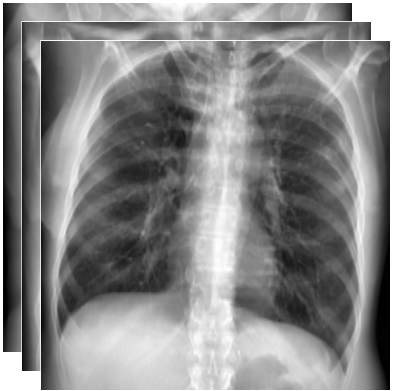
DRRs Obtained via Projection



Super-Resolution: Simple Enough?

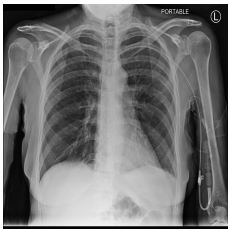


Let's generate paired data using HQ X-rays and train a super-resolution/enhancement model. Simple enough?



Results from transformer-based methods: SwinFIR (Zhang et al., 2023), SwinIR (Liang et al., 2021)

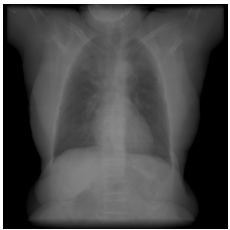
Results from Bicubic Interpolation



Input

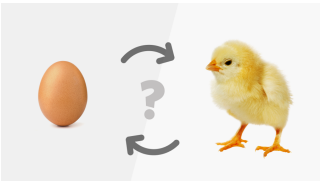


Bicubic interpolation



Reference

Stuck!



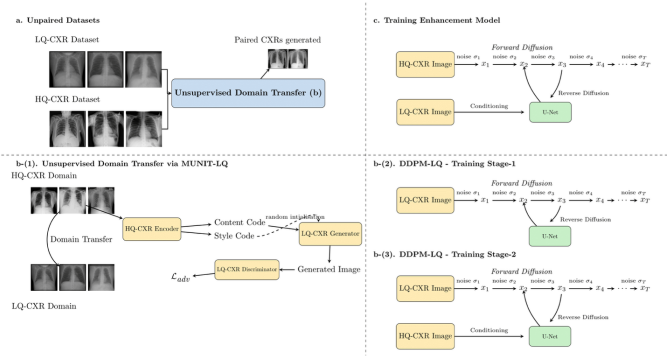
Training an enhancer needs realistic low/high-quality pairs, and making realistic pairs needs a model of how DRRs degrade.

Helpful Papers on Domain Adaptation

- **Synthesizing Realistic Image Restoration Training Pairs.** Tao Yang et al., arXiv:2303.06994, 2023. Diffusion-based generation of realistic LR–HR training pairs.
- **Towards Realistic Data Generation for Real-World Super-Resolution.** Long Peng et al., arXiv:2406.07255, 2025. Addresses the domain gap in real-world SR via realistic data synthesis.
- **Multimodal Unsupervised Image-to-Image Translation (MUNIT).** Xun Huang et al., ECCV 2018, arXiv:1804.04732. Multimodal I2I framework widely used for domain adaptation.

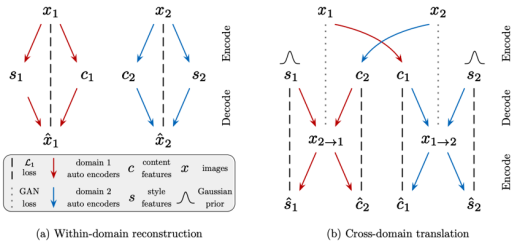
DiffusionXRay: A 2-Stage Pipeline

1. Generate realistic LQ CXRs from HQ:
   - MUNIT-LQ (unpaired style transfer)
   - DDPM-LQ (diffusion degradation model)
2. Train a DDPM-HQ enhancer on the resulting paired HQ–LQ data.



Methodology: MUNIT-LQ and DDPM-LQ

MUNIT-LQ (unpaired style transfer)



(a) Within-domain reconstruction (b) Cross-domain translation

Disentangles content from style, and combines HQ content with LQ style.

DDPM-LQ (diffusion degradation model)

- A conditional DDPM learns realistic degradations.
- Stage 1: train on real LQ DRRs (unconditional).
- Stage 2: fine-tune conditioned on HQ to generate paired LQ.

Qualitative Results

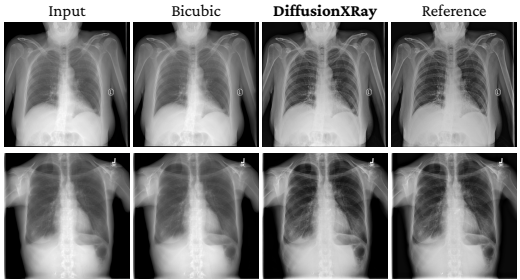


Figure 10: Image enhancement and super-resolution (512×512 → 1024×1024) on synthetically degraded CXRs.

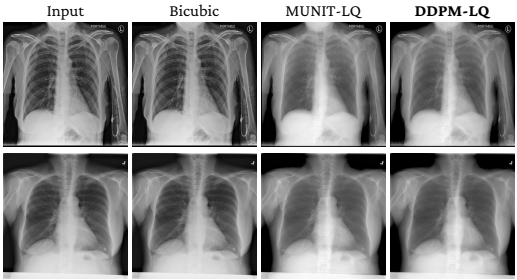


Figure 11: Low-quality CXR synthesis using Bicubic, MUNIT-LQ, and DDPM-LQ (ours).

Quantitative: SR + Enhancement

Setting	PSNR ↑	SSIM ↑
Results with MUNIT-LQ generated data		
Bicubic-DDPM	20.08	0.83
DiffusionXRay (ours)	**27.50**	**0.92**
Results with DDPM-LQ generated data		
DDPM-LQ (baseline)	19.85	0.78
DDPM-LQ (ours)	**22.21**	**0.78**

Table 7: Comparison of low-quality simulation strategies for chest X-rays. Higher PSNR and SSIM indicate better reconstruction quality.

SR = Super-Resolution

Radiologist Evaluation and Limitations

Task	Question	DiffusionXRay (ours)	Bicubic-DDPM
1: Nodule preservation	Nodule easily visible? ↑	**100%**	6.6%
	Confusion with other structures? ↓	**0%**	30%
2: Quality assessment	Lung field clarity improved? ↑	**100%**	66.7%
	Significant noise increase? ↓	72.9%	**25%**

Table 8: Radiologist assessment of enhanced CXRs.

- 100% nodule visibility with 0% confusion: subtle findings are preserved.
- Clarity improved in all cases vs. 66.7% for the baseline, consistent with the higher SSIM.
- 72.9% perceived noise: a deliberate trade-off favouring conspicuity over smoothing.

Limitation: costly training and inference.

Next: lighter backbones or distillation; non-DRR settings such as portable X-rays.

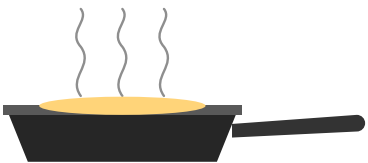
Long-Horizon Planning for Real Robot Actions.

A research problem posed for this talk

Research problem

The Problem Statement

"Design and evaluate a pipeline that can generate long-term planning for the actual robotic action. Long-term planning means that you need to finish a full task, like cooking or cleaning house, instead of only manipulating some objects. The actual robotic action means that you need to consider the future planning is for the actual robotic arms."



Research problem

A Hierarchical Policy

A **high-level VLM** maps the observations $\boldsymbol{o}$ and the task instruction I to a language instruction l. A **low-level VLA** maps l and the current observation o_t to robot actions a.

$$\pi_{\text{HiVLA}}(a \mid [o_i]_{i \le t}, I) = \int_l \underbrace{\pi_{\text{VLA}}(a \mid o_t, l)}_{\text{low-level VLA}} \underbrace{\pi_{\text{VLM}}(l \mid \boldsymbol{o}, I)}_{\text{high-level VLM}} \, dl$$

$$\text{where} \quad \boldsymbol{o} = \underbrace{\text{mem}([o_i]_{i \le t})}_{\text{memory module}}$$

I: task instruction l: language instruction o_t: current observation a: robot action

Equation: Hu et al., What Matters in Orchestrating Robot Policies, arXiv:2606.10267, 2026.
World models and the other components come in later slides.

Research problem

Design Choice: Termination Condition

When should the VLA hand control back to the VLM?

Option	Long-horizon success (%)
VLM predicts the execution time	43.5
Fixed horizon	52.4
Success detector	**57.4**

Takeaway: stop when a success detector says the subtask is done; it stays robust under moderate detection error.

Recipe so far

- **Structure:** high-level VLM + low-level VLA
- **Termination:** success detector

Hu et al., What Matters in Orchestrating Robot Policies, arXiv:2606.10267, 2026.

Research problem

Design Choice: Memory

How much history should the VLM see, and in what form?

Option	Long-horizon success (%)
Raw history: last 1 / 3 / 5 steps / full	59.9 / 58.2 / 57.8 / 59.0
No summary	52.4
Summary of the current episode	50.1
Summary of previous episodes	**60.0**

Takeaway: a short window is enough; what helps is experience from previous episodes, summarized.

Recipe so far

- **Structure:** high-level VLM + low-level VLA
- **Termination:** success detector
- **Memory:** short window, cross-episode summary

Hu et al., What Matters in Orchestrating Robot Policies, arXiv:2606.10267, 2026. The two blocks are separate sweeps.

Research problem

Design Choice: Observations

What should the VLM see of the current state?

Option	Long-horizon success (%)
Image only	38.8
Image + description	35.7
Image + description + bounding boxes	47.9
Image + description + contact info	**52.4**

Takeaway: turn the image into a text description grounded in bounding boxes or contact; boxes need no extra sensing.

Recipe so far

- **Structure:** high-level VLM + low-level VLA
- **Termination:** success detector
- **Memory:** short window, cross-episode summary
- **Observation:** text description with boxes or contact

Hu et al., What Matters in Orchestrating Robot Policies, arXiv:2606.10267, 2026. Contact info is privileged simulator state.

What's Next

Ongoing: when is the implicit structure a policy learns enough for dynamics, and when does explicit prediction help?

- Prior work tests this only under **visual shift**: appearance changes, physics does not.
- We shift the physics instead (mass, friction, centre of mass) while holding appearance fixed.
- Two arms that differ in **exactly one loss term**, so any gain comes from the prediction objective alone.
- We also ask which tasks benefit: long-horizon, contact-rich, or failure recovery.

Thank you.

garyan18@gmail.com aryangoyal7.github.io X: @arygoy