Uncertainty-Conditioned Generative Augmentation for Pool-Based Active Learning

Ehsan Garaaghaji, Elyar Esmaeilzadeh, Farzad Hallaji Azad, Ida Fallah Ardalani
Department of Computer Science
Bilkent University
Ankara, Türkiye
{ehsan.garaaghaji, elyar, farzad.hallaji, ida.fallah}@bilkent.edu.tr

***Abstract*—Active learning seeks to reduce annotation costs by selectively querying the most informative samples from a large pool of unlabeled data. In practice, this promise is sometimes not realised: uncertainty estimates can be noisy or miscalibrated, and the unlabeled pool may lack points near the classifier's decision boundary. Under these conditions, repeatedly querying the most uncertain real samples yields diminishing returns while quickly consuming the labeling budget. We address both issues by augmenting the active learning loop with synthetic candidates generated around regions of high uncertainty. Two complementary mechanisms are explored: (i) an uncertainty-conditioned conditional GAN (Unc-CGAN) that learns to generate inputs conditioned on the current model's scalar uncertainty, and (ii) a variational autoencoder (VAE) that decodes new inputs by interpolating between latent representations of uncertain samples. At each acquisition step the generative models are retrained, and the synthetic candidates are merged with the real pool before a standard uncertainty-based policy selects the next batch. Extensive experiments across nine datasets and multiple acquisition settings demonstrate that VAE-based augmentation provides the largest average accuracy improvements, while Unc-CGAN is competitive on tabular datasets.**

***Index Terms*—Active learning, uncertainty estimation, generative modelling, Bayesian deep learning**

I. Introduction

Labeling data for supervised learning can be costly and time-consuming, especially in domains such as vision, natural language and science. Active learning (AL) addresses this problem by iteratively training a model on a small labeled subset and then querying an oracle for labels on unlabeled points that are expected to be most informative. Under suitable conditions, selecting high-value examples accelerates model performance and reduces the total number of labels required. However, practical deployments of AL often underperform for two distinct reasons. First, uncertainty estimates—the backbone of many pool-based acquisition rules—vary across estimation techniques and can be unreliable. Methods such as Monte Carlo dropout, deep ensembles and Laplace approximations each have different computational and statistical trade-offs; without careful calibration, they may produce overconfident or underconfident scores that mislead the selection process. Second, when the initial labeled set is very small, the unlabeled pool may contain few or no points near the true decision boundary. In such cases, simply selecting the most uncertain real points yields marginal gains and quickly exhausts the labeling budget.

To overcome these limitations, we propose to augment the pool with synthetic candidates generated in regions of high model uncertainty. After adding these synthetic examples, a standard uncertainty-based policy is used to select which items to label. We implement two complementary generative mechanisms. The first is an uncertainty-conditioned conditional generative adversarial network (Unc-CGAN) that models $p(x \mid u)$, where u is a continuous uncertainty value produced by the current classifier. Conditioning on u encourages the generator to produce candidate samples with specified uncertainty levels. The second mechanism trains a variational autoencoder on all available training data and constructs new inputs by interpolating between the latent representations of high-uncertainty real points; decoding these interpolations yields samples that traverse the decision boundary. Both generators are retrained after each acquisition step so that they remain aligned with the evolving classifier.

Our contributions are as follows:

- **Generative augmentation for AL.** We introduce a unified active learning loop that augments the unlabeled pool with uncertainty-aware synthetic candidates—produced by an Unc-CGAN or by VAE latent interpolation—while keeping the acquisition rule unchanged.
- **Comprehensive evaluation.** We generate reproducible figures and tables that summarise performance across a broad array of conditions.
- **Statistical analysis.** We perform paired sign-flip tests and compute effect sizes to quantify the consistency of observed gains across ablation settings.

II. Background and Related Work

Pool-based active learning commonly prioritizes samples using model uncertainty [1]. In modern neural settings, the effectiveness of uncertainty sampling is often dominated by the quality of the underlying uncertainty approximation. Practical choices include MC dropout [2], deep ensembles [3], and Laplace approximations over the last layer [4]; these methods trade off calibration, compute, and implementation complexity, and can behave differently in low-label regimes. Generative modeling provides a complementary lens: rather than selecting exclusively from the finite unlabeled pool, a generator can propose additional candidates to densify regions near the evolving decision boundary. Conditional GANs [5] and VAEs

[6] offer two standard mechanisms for controllable generation and smooth interpolation, respectively. In our setting, the conditioning signal is not the class label but a scalar uncertainty value produced by the current predictor, allowing the generator to target a specific uncertainty band; for VAEs, interpolation between latents of uncertain pool items provides a lightweight way to propose boundary-adjacent candidates without requiring high-fidelity image synthesis. In this work, we combine uncertainty estimation with generative augmentation into a consistent active-learning loop. Our framework adds synthetic candidates to the unlabeled pool and then uses standard uncertainty sampling to select queries, without altering the acquisition policy. We assess performance across classification and regression settings using suitable metrics across multiple configurations.

III. Methodology

A. Problem formulation

Let $D = \{(x_i, y_i)\}$ be the training set and D_U an unlabeled pool sampled from the same distribution. We split D into an initial labeled subset D_L and the remainder D_U at the start of each run. In every acquisition round t, we (i) train or retrain the main classifier f_{θ_t} on D_L, (ii) score the pool with an uncertainty estimator $U_{\theta_t}(x)$, (iii) optionally augment the pool with synthetic candidates, (iv) select the top-k uncertain points, query their labels from an oracle, update D_L, and repeat. The acquisition set is the greedy maximizer

$$\mathcal{B}_t = \underset{\substack{\mathcal{S}\subseteq D_U \\ |\mathcal{S}|=k}}{\arg\max} \sum_{x\in\mathcal{S}} U_{\theta_t}(x), \tag{1}$$

which reduces to picking the k highest-scoring pool elements. Retraining from scratch ensures that each acquisition decision is evaluated under the same optimization protocol and eliminates subtle state leakage across rounds.

B. Backbone predictor, oracle, and visualization

All methods share a lightweight multilayer perceptron (MLP) backbone with hidden sizes $[64, 32]$ for tabular/2D data and $[512, 256]$ for high-dimensional inputs. Weights are randomly initialized for every training cycle. The *oracle* is another MLP trained once on the full training split; it provides labels for both acquired real samples and generated samples. To aid qualitative analysis, every acquisition step produces a two-panel plot: decision regions (via PCA projection when the input dimensionality exceeds two) and a heatmap of uncertainty values. For image datasets we additionally dump grids of *real certain*, *real uncertain*, and *generated* samples (logged in the saved run artifacts).

C. Uncertainty estimators

We expose three estimators:

- **MC dropout** [2]: predictive entropy of the Monte Carlo averaged predictive distribution,

$$U_{\text{drop}}(x) = -\sum_c \bar{p}_c \log \bar{p}_c, \tag{2}$$

  while for regression we use predictive variance.
- **Deep ensembles** [3]: entropy of the averaged ensemble predictive distribution.
- **Last-layer Laplace** [4]: entropy of predictions under a diagonal Gaussian posterior over the output layer.

All estimators share the same training data and hyperparameters; differences reflect estimator behavior rather than confounding factors.

D. Baseline policies

No-AL trains the classifier once on D_L and evaluates it on the test split; it captures the performance level achievable without any oracle queries. **Baseline-AL** applies uncertainty sampling directly to the pool: select the top-k most uncertain real points, query the oracle, augment D_L, and retrain. This baseline already benefits from MC dropout/ensemble/Laplace scoring and from the retrain-from-scratch discipline, making it a strong reference point.

E. Uncertainty-conditioned CGAN augmentation

The CGAN generator $G(z, u)$ takes Gaussian noise z and a continuous uncertainty scalar u, while the discriminator $D(x, u)$ judges whether (x, u) is real. At acquisition step t we (i) compute uncertainty scores $u(x)$ for $x \in D_L \cup D_U$, (ii) train the CGAN on pairs $(x, u(x))$ for a small number of steps, and (iii) sample candidate points by conditioning on a high-uncertainty band (e.g., 80–100th percentile), then filter by $U_{\theta_t}(x)$ and oracle-label the retained synthetic points. Because conditioning uses the model's uncertainty rather than the class label, the generator tracks the evolving boundary; uncertainty filtering prevents collapse onto easy regions.

F. VAE latent-interpolation augmentation

The VAE learns an encoder $q_\phi(z \mid x)$ and a decoder $p_\psi(x \mid z)$ on the full training set. During AL we encode the most uncertain real pool points into latents $\{z_i\}$ and interpolate in latent space:

$$\tilde{z} = \alpha z_i + (1-\alpha) z_j + \epsilon, \qquad \alpha \sim \text{Beta}(0.5, 0.5),\ \epsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{3}$$

then decode $\tilde{z}$ to candidates, filter by U_{θ_t}, enforce diversity via farthest-point sampling in latent space, and oracle-label the survivors. This is lightweight (one VAE per dataset) and leverages latent smoothness to populate regions between uncertain points.

G. Acquisition loop with augmentation

After either generator produces synthetic points, we merge them with the real pool, recompute uncertainty scores, and select the next batch of k points (real or synthetic) with highest uncertainty. Oracle responses are deterministic because the oracle is a supervised model trained on all available training data. Finally, the classifier is retrained from scratch on the enlarged labeled set. Both the CGAN and VAE are retrained every round, but thanks to the small architectures and datasets considered, this overhead remains manageable (tens of seconds per round on GPU).

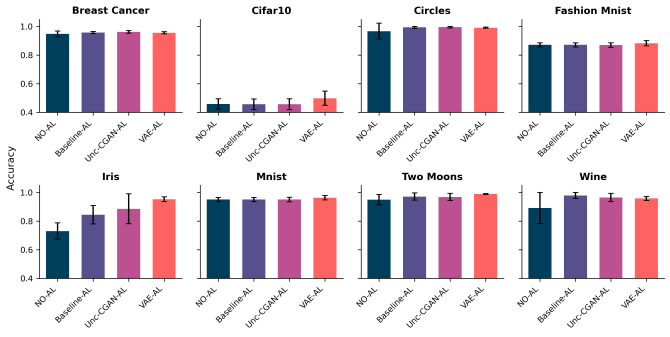


Fig. 1: Accuracy across datasets for each method (ensemble uncertainty, $f_{\text{init}} = 0.1$). Generative augmentation methods consistently perform well across diverse dataset types.

IV. Experiments

A. Datasets and compared methods

We compare four policies (NO-AL, Baseline-AL, Unc-CGAN-AL, VAE-AL) on nine datasets spanning tabular, synthetic 2D, and vision benchmarks (Iris, Wine, Breast Cancer, Two Moons, Circles, Boston/California Housing, MNIST, Fashion-MNIST, CIFAR-10).

B. Active learning protocol

We use a fixed 80/20 train/val split. The training split is partitioned into an initial labeled set ($f_{\text{init}} \in \{0.1, 0.3, 0.5, 0.8\}$) and an unlabeled pool; each run performs two acquisition rounds with batch size $k = 20$. After each acquisition, the classifier is reinitialized and retrained from scratch on the expanded labeled set to avoid warm-start confounds.

C. Oracle, uncertainty, and augmentation

The oracle is a supervised MLP trained on the full training split and provides labels for acquired real points and retained synthetic points. We evaluate three uncertainty estimators (MC dropout, ensembles, Laplace) and use lightweight generators with modest training to keep per-round overhead practical.

D. Metrics and reporting

For classification we report accuracy (primary), macro-F1, and NLL; for regression we report RMSE (primary), MAE, and MSE. All reported numbers are mined from saved ablation artifacts without rerunning training. To avoid averaging across experimental factors, we report the main tables for a single fixed configuration (ensemble uncertainty with $f_{\text{init}} = 0.1$), and we separately report best-over-grid results when appropriate.

V. Results and Analysis

A. Per-dataset performance overview

Figure 1 provides a summary view of accuracy across all datasets and methods using ensemble uncertainty at $f_{\text{init}} = 0.1$.

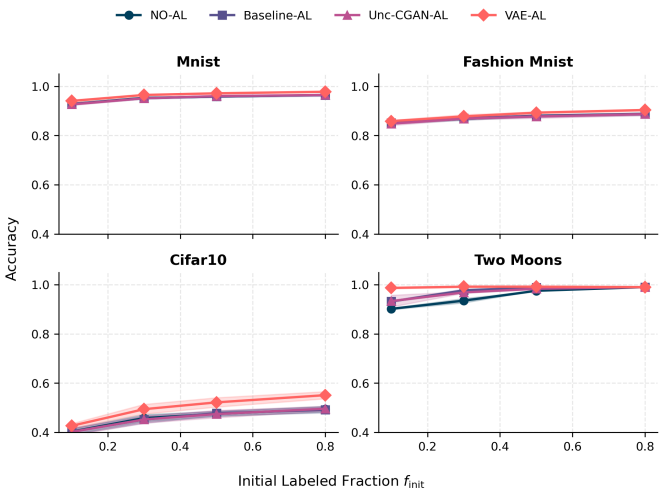


Fig. 2: Accuracy vs. initial labeled fraction (f_{init}) for four representative datasets (ensemble uncertainty). Generative augmentation delivers the largest gains when f_{init} is small.

B. Robustness to Uncertainty Estimator

We first evaluate whether generative augmentation provides consistent gains regardless of the underlying uncertainty estimator used. Table I presents classification accuracy at the lowest label budget ($f_{\text{init}} = 0.1$), a critical regime for active learning. We observe that **VAE-AL** consistently outperforms or matches the baselines across all three estimators (Dropout, Deep Ensemble, Laplace). Notably, on difficult datasets like CIFAR-10, VAE-AL improves upon Deep Ensembles (0.437 vs 0.417) and Laplace (0.420 vs 0.393), suggesting that synthetic candidates successfully augment the pool even when the estimator is noisy.

C. Sensitivity to Label Budget

We analyze how performance scales as the initial labeled fraction (f_{init}) increases from 0.1 to 0.8. Table II reports classification accuracy for Baseline-AL and VAE-AL using Deep Ensembles across all four budget levels. Generative augmentation provides the largest gains when labels are scarce ($f_{\text{init}} \in \{0.1, 0.3\}$). For example, on Iris, VAE-AL achieves 0.933 accuracy at $f_{\text{init}} = 0.1$ while Baseline-AL reaches only 0.667. On CIFAR-10, the advantage persists across all budgets, with VAE-AL reaching 0.567 at $f_{\text{init}} = 0.8$ compared to 0.505 for Baseline-AL. As the budget increases, the performance gap narrows on simpler datasets (e.g., Circles, Two Moons) where both methods eventually converge.

Figure 2 visualizes these trends for four representative datasets, confirming that generative augmentation delivers the largest gains when f_{init} is small.

D. Uncertainty estimator effects

Figure 3 shows performance aggregated across datasets for each method/uncertainty estimator pair, demonstrating that generative augmentation narrows the performance spread between different uncertainty approximations.

TABLE I: **Robustness to Uncertainty Estimator:** Classification Accuracy at low label budget ($f_{\text{init}} = 0.1$). **VAE-AL** consistently improves over baselines regardless of the uncertainty estimation method used (Dropout, Ensemble, or Laplace).

Dataset	MC Dropout			Deep Ensemble			Laplace Approx.		
	Base	Unc-CGAN	VAE	Base	Unc-CGAN	VAE	Base	Unc-CGAN	VAE
Breast Cancer	**0.974**	0.965	**0.974**	**0.965**	**0.965**	0.956	0.939	**0.974**	0.947
CIFAR-10	0.399	0.396	**0.423**	0.417	0.419	**0.437**	0.393	0.391	**0.420**
Circles	0.975	0.990	**0.990**	**1.000**	**1.000**	0.990	**0.995**	0.985	**0.995**
Fashion-MNIST	0.849	0.845	**0.857**	0.857	0.856	**0.862**	0.849	0.843	**0.857**
Iris	0.833	0.600	**0.933**	0.667	0.767	**0.933**	0.833	0.900	**0.933**
MNIST	0.924	0.925	**0.938**	0.931	0.931	**0.945**	0.928	0.923	**0.938**
Two Moons	0.930	0.960	**0.985**	0.930	0.925	**0.990**	0.935	0.915	**0.985**
Wine	**1.000**	0.917	0.944	**0.972**	0.944	**0.972**	**0.972**	0.944	0.944

TABLE II: **Label Budget Ablation:** Classification Accuracy using Deep Ensembles across all initial label fractions (f_{init}). **VAE-AL** offers the most significant gains in the scarce-label regimes (0.1 and 0.3).

Dataset	$f_{\text{init}} = 0.10$		$f_{\text{init}} = 0.30$		$f_{\text{init}} = 0.50$		$f_{\text{init}} = 0.80$	
	Baseline	VAE-AL	Baseline	VAE-AL	Baseline	VAE-AL	Baseline	VAE-AL
Breast Cancer	**0.965**	0.956	**0.956**	**0.956**	**0.956**	0.947	**0.956**	**0.956**
CIFAR-10	0.417	**0.437**	0.470	**0.515**	0.488	**0.543**	0.505	**0.567**
Circles	**1.000**	0.990	**1.000**	0.985	**1.000**	0.990	**1.000**	0.995
Fashion-MNIST	0.857	**0.862**	0.877	**0.885**	0.883	**0.896**	0.893	**0.907**
Iris	0.667	**0.933**	0.867	**0.933**	0.867	**0.967**	0.867	**0.967**
MNIST	0.931	**0.945**	0.955	**0.967**	0.963	**0.974**	0.968	**0.979**
Two Moons	0.930	**0.990**	0.980	**0.990**	0.985	**0.990**	**0.990**	**0.990**
Wine	**0.972**	**0.972**	0.944	**0.972**	**1.000**	0.972	**1.000**	0.944

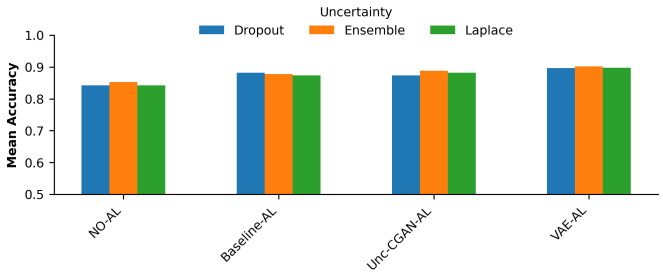
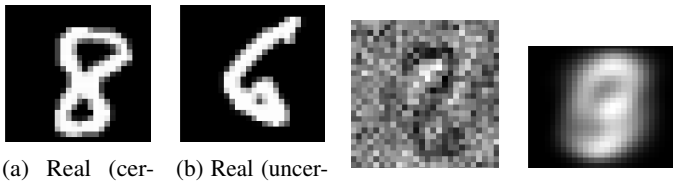


Fig. 3: Average accuracy aggregated over datasets for each method/uncertainty estimator pair. Generative augmentation narrows the spread between uncertainty estimators.



(a) Real (certain) (b) Real (uncertain) (c) Unc-CGAN (d) VAE

Fig. 4: Representative MNIST samples: real certain/uncertain examples and one generated sample from each generator.

TABLE III: **Regression Results (Boston Housing):** RMSE (lower is better) using Deep Ensembles.

Method	f = 0.10	f = 0.30	f = 0.50	f = 0.80
NO-AL	0.657	0.589	0.571	0.551
Baseline-AL	0.656	0.604	0.572	0.558
Unc-CGAN-AL	0.657	0.602	0.573	0.558
VAE-AL	**0.625**	**0.579**	**0.550**	**0.549**

E. Regression Results

We evaluate our methods on the Boston Housing regression task in Table III. VAE-AL provides consistent improvements in RMSE across budget levels. Notably, at $f_{\text{init}} = 0.1$, VAE-AL reduces RMSE from 0.656 (Baseline) to 0.625.

F. Qualitative observations

Qualitative frames logged per acquisition step illustrate that generative candidates populate regions where the real pool is sparse, especially early in the acquisition loop. On MNIST, generated samples resemble low-fidelity but high-entropy images, which is sufficient to elicit informative oracle responses when filtered by uncertainty. See Figure 4. On tabular datasets, the visualizations show synthetic points bridging clusters of real data, enabling the classifier to carve cleaner decision boundaries sooner. Figure 5 illustrates this effect on the Two Moons dataset, showing how decision regions evolve from early to late acquisition steps.

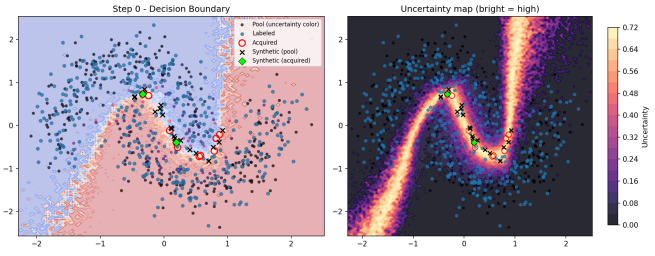
Overall, VAE-AL shows the most consistent gains over Baseline-AL across datasets; Unc-CGAN-AL is competitive on simpler/tabular datasets but weaker on CIFAR-10. Because the ablation grid uses one seed, we treat this as supportive evidence and recommend multi-seed validation for robustness. Finally, following established best practice for small-sample, paired experimental comparisons, we assess statistical significance using a paired sign-flip randomization test rather than parametric alternatives. See Table IV for the significance report.

VI. Implementation and Dependencies

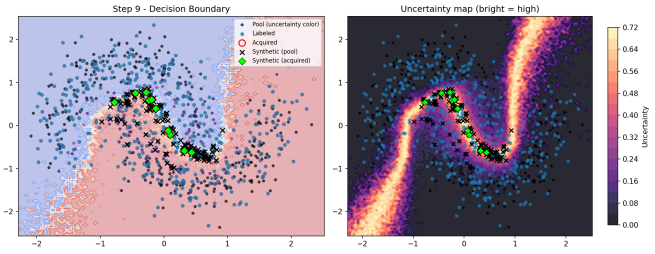
We implement our framework using PyTorch for neural network training and scikit-learn for data preprocessing and evaluation metrics. The codebase uses NumPy for numerical operations, pandas for data manipulation, and matplotlib/seaborn

Dataset	Comparison	n	Δ mean	d	p
boston	VAE-AL vs Baseline-AL	11	0.653	1.606	0.001
boston	Unc-CGAN-AL vs Baseline-AL	12	-0.002	-0.084	0.852
boston	VAE-AL vs Unc-CGAN-AL	11	0.655	1.606	0.001
breast_cancer	VAE-AL vs Baseline-AL	12	-0.002	-0.232	0.688
breast_cancer	Unc-CGAN-AL vs Baseline-AL	12	0.004	0.332	0.219
breast_cancer	VAE-AL vs Unc-CGAN-AL	12	-0.006	-0.445	0.132
cifar10	VAE-AL vs Baseline-AL	12	0.042	2.843	0.001
cifar10	Unc-CGAN-AL vs Baseline-AL	12	0.000	0.029	0.874
cifar10	VAE-AL vs Unc-CGAN-AL	12	0.042	3.138	0.001
circles	VAE-AL vs Baseline-AL	12	-0.002	-0.234	0.527
circles	Unc-CGAN-AL vs Baseline-AL	12	0.002	0.243	0.547
circles	VAE-AL vs Unc-CGAN-AL	12	-0.004	-0.553	0.125
fashion_mnist	VAE-AL vs Baseline-AL	12	0.011	2.274	0.001
fashion_mnist	Unc-CGAN-AL vs Baseline-AL	12	-0.002	-0.614	0.055
fashion_mnist	VAE-AL vs Unc-CGAN-AL	12	0.013	2.727	0.001
iris	VAE-AL vs Baseline-AL	12	0.108	1.842	0.001
iris	Unc-CGAN-AL vs Baseline-AL	12	0.042	0.457	0.174
iris	VAE-AL vs Unc-CGAN-AL	12	0.067	0.692	0.008
mnist	VAE-AL vs Baseline-AL	12	0.013	6.437	0.001
mnist	Unc-CGAN-AL vs Baseline-AL	12	0.000	0.078	0.785
mnist	VAE-AL vs Unc-CGAN-AL	12	0.013	7.260	0.001
two_moons	VAE-AL vs Baseline-AL	12	0.018	0.794	0.008
two_moons	Unc-CGAN-AL vs Baseline-AL	12	-0.003	-0.240	0.445
two_moons	VAE-AL vs Unc-CGAN-AL	12	0.021	0.891	0.004
wine	VAE-AL vs Baseline-AL	12	-0.021	-0.711	0.062
wine	Unc-CGAN-AL vs Baseline-AL	12	-0.014	-0.362	0.344
wine	VAE-AL vs Unc-CGAN-AL	12	-0.007	-0.259	0.273

TABLE IV: Significance report using paired sign-flip randomization tests across ablation settings. We evaluate the primary metric per task (accuracy for classification; RMSE for regression), and normalize differences so positive Δ indicates an improvement for the first method in the comparison. n is the number of paired settings available (typically 12).



(a) Step 0 (early)



(b) Step 9 (late)

Fig. 5: Two Moons acquisition dynamics (Unc-CGAN-AL): decision regions and uncertainty map at the first vs. last acquisition step.

for visualization. All generative models (Conditional GAN and VAE) are implemented from scratch using PyTorch primitives; we do not use off-the-shelf GAN/VAE libraries. Uncertainty estimation methods (MC dropout, deep ensembles, last-layer Laplace approximation) are implemented directly based on established algorithms. The active learning loop, acquisition strategies, and synthetic augmentation pipelines are custom-built for this work. Training infrastructure (oracle labeling, model initialization, retraining discipline) is implemented explicitly to ensure reproducibility. We have Use ChatGPT: Codex for implementation.

VII. Conclusion

We presented a reproducible AL framework combining three uncertainty estimators with two generative augmentation strategies. We find consistent accuracy gains from VAE-based augmentation and competitive performance from Unc-CGAN on simpler/tabular datasets, especially under label-scarce regimes.

References

[1] B. Settles, *Active Learning Literature Survey*, University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[2] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016.

[3] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[4] D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, California Institute of Technology, PhD thesis, 1992.

[5] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv preprint arXiv:1411.1784*, 2014.

[6] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Int. Conf. Learn. Representations (ICLR)*, 2014.