ON UNIFORMLY SCALING FLOWS: A DENSITY-ALIGNED APPROACH TO DEEP ONE-CLASS CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised anomaly detection is commonly approached through two major paradigms: deep one-class methods learn compact representations of nominal data, whereas deep density estimators such as normalizing flows model exact likelihoods. We show that uniformly scaling flows (USFs), normalizing flows with a constant Jacobian determinant, provide a principled bridge between these paradigms. Specifically, we prove that maximum-likelihood training of a USF reduces to a one-class objective whose inherent regularization mechanism prevents representational collapse and incentivizes density-aligned latent representations. This connection implies that USFs inherit both the density faithfulness of flows and the distance-based reasoning of one-class methods: their negative log-likelihood is monotonic in the latent norm, whereas in non-USFs the Jacobian term can entangle likelihood with local volume changes. Consequently, we advocate USFs as drop-in replacements for non-USFs in modern anomaly detection architectures and identify mechanisms that recover expressivity and enhance numerical stability. Across tabular and image benchmarks, this substitution simplifies optimization, markedly reduces run-to-run variability, and consistently matches or improves detection quality, while mitigating severe failure modes. Thus, our results unify two central perspectives on anomaly detection, indicating that normalizing flow-based methods benefit more from density-faithful geometry than from greater nominal expressivity.

1 INTRODUCTION

Unsupervised anomaly detection seeks to identify rare deviations from normality without access to labeled outliers and underpins safety-critical applications, including medical diagnostics Fernando et al. (2021) and industrial inspection Liu et al. (2024). For such complex data, performance and reliability hinge on how well "normal" structure is represented in latent space, where typical samples should occupy compact and semantically meaningful regions that distinguish them from anomalous patterns. Characterizing the underlying data-generating process, implicitly through distances or explicitly through density estimation, provides a principled foundation for this task Ruff et al. (2021).

Within the landscape of leading deep unsupervised anomaly detection methods, there exist two seemingly complementary lines: one-class classification, epitomized by Deep SVDD Ruff et al. (2018), which forgoes explicit density mapping to learn encoders that concentrate normal data into minimal-volume support (e.g., a hypersphere), and distribution mapping, which estimates a distribution over embeddings Liu et al. (2024). Among the latter, a predominant instance are normalizing flows, generative density estimator models which learn bijective transformations between the data space and simple base distributions and enable exact likelihoods Papamakarios et al. (2021). Common flow architectures broaden the admissible transformations through input-dependent volume adjustments Dinh et al. (2017), increasing expressivity but potentially decoupling likelihood from latent distance. One-class methods on the other hand offer geometric simplicity and efficiency, but they risk trivial solutions, and limited density fidelity Chong et al. (2020); Zhang & Deng (2021).

Despite their distinct mechanisms, both paradigms ultimately seek to characterize a high-mass region of normality. What has been missing is a rigorous account of when their objectives coincide, what this connection entails, and how it can be leveraged to further improve anomaly detection methods.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

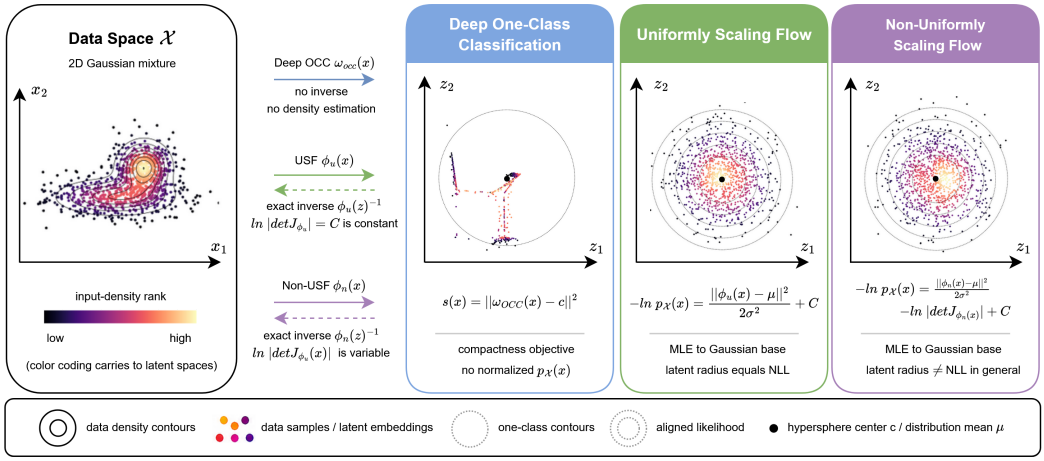


Figure 1: **Conceptual outline of the three paradigms considered.** Seminal deep one-class methods like Deep SVDD learn a mapping $\omega(\cdot, W)$ that concentrates normal samples near a latent center c and employ distance-based scoring. Normalizing flows learn an invertible mapping $\phi(\cdot, W)$ that moves data towards a simple latent distribution, such as an isotropic Gaussian. Uniformly scaling flows connect these views: with a radial monotonic latent distribution, likelihood ranking becomes monotone in latent norm, yielding one-class geometry within an invertible and density-aligned model.

We close this gap by introducing a theoretically grounded bridge via uniformly scaling flows (USFs), normalizing flows whose Jacobian determinant is constant across inputs. Our contributions are threefold. **First**, we show that USFs realize a flow-based analogue of Deep SVDD within an exact density-modeling framework. In particular, for a standard Gaussian base, we formally establish that USF maximum likelihood training reduces to a deep one-class objective within a distinctive regularization scheme. This regularization directly penalizes volume collapse in a way that provably circumvents well-known degeneracy issues of standard deep one-class classification. We further show that the resulting negative log-likelihood increases monotonically with latent norm, connecting one-class geometry to exact density modeling. Figure 1 visualizes this relationship. **Second**, we develop USF instantiations as drop-in replacements for non-uniformly scaling flows in existing anomaly detection architectures and identify mechanisms that improve expressivity and numerical stability. **Third**, across tabular and industrial image benchmarks, these instantiations consistently match or improve upon state-of-the-art one-class and flow-based methods, with better training stability, greater robustness to contamination, and lower run-to-run variability. These results show how aligning likelihood with latent geometry can improve anomaly detection despite reduced nominal expressivity.

2 Preliminaries

To first define unsupervised anomaly detection formally, let X be a random variable governed by a probability density function $p_X(x)$ that models the true distribution of normal data. An observation x is deemed anomalous if its likelihood under this distribution falls below a predefined threshold τ, i.e.,

$$\{\, x \mid p_X(x) < \tau \}. \tag{1}$$

This characterizes anomalies as those points residing in the low–probability tails of the normal data distribution, consistent with the definition that "an anomaly is an observation that deviates considerably from some concept of normality" Ruff et al. (2021).

Deep One-Class Classification Rather than estimating the true density $p_X(x)$ and applying a threshold test as in Equation 1, deep one-class classification methods learn a representation jointly with a latent distance geometry such that normal samples occupy a compact or otherwise structured region in latent space. *Deep Support Vector Data Description (Deep SVDD)* Ruff et al. (2018) is the seminal instance of this paradigm, which learns a mapping that pulls normal samples toward a fixed hypersphere center, such that latent distance provides the criterion for anomaly scoring.

To this end, Deep SVDD parameterizes a deep neural network $\omega(x, w)\colon X \to \mathbb{R}^d$ with layer weights $w = (w^1, \ldots, w^L)$ and learns these weights by optimizing the one-class loss

$$\min_w \frac{1}{n}\sum_{i=1}^{n}\|\omega(x_i, w) - c\|^2 \; + \; \frac{\lambda}{2}\sum_{l=1}^{L}\|w^l\|_F^2 \tag{2}$$

where $\{x_i\}_{i=1}^n$ are the normal training samples and $c \in \mathbb{R}^d$ is the hypersphere center. The second term corresponds to a weight decay regularizer with hyperparameter $\lambda > 0$ where $\|\cdot\|_F$ denotes the Frobenius norm. For inference, the squared distance $\|\omega(x, w) - c\|^2$ is treated as anomaly measure.

Normalizing Flows Flow models are commonly used as density estimators and generative models Kobyzev et al. (2021); Papamakarios et al. (2021). Formally, a flow ϕ transforms a usually complex data (or target) distribution X into a typically simple base (or latent) distribution, $\phi_{\#}p_X = p_B$, using a differentiable invertible map with differentiable inverse, i.e., a diffeomorphism. The map ϕ_w is implemented by an invertible neural network with parameters w. We obtain a density estimator and a generative model:

1. Sampling is performed by first sampling $z \sim B$ and then computing the inverse map $\phi_w^{-1}(z)$.
2. The likelihood $p_X(x)$ is computable with the change of variables formula Folland (1999):

$$p_X(x) = |\det J_{\phi_w}(x)|\, p_B(\phi_w(x)), \tag{3}$$

where $J_{\phi_w}(x) = \frac{\partial \phi_w}{\partial x}$ denotes the Jacobian matrix of ϕ_w, i.e. $|\det J_{\phi_w}(x)|$ describes the local volume change of the transformation. Flows are typically trained to estimate the density of some unknown distribution X by fixing the base distribution and maximizing the log-likelihood of samples from X using Equation 3. In this case, we write $\hat{p}_X^w(x)$ in order to avoid confusion with the true density p_X. The by far most common base distribution is a standard Normal $\mathcal{N}(0, I)$, which motivates the name normalizing flow. However, any distribution that is completely described by a pdf can act as base distribution. The class of *radial monotonic* distributions is of particular interest to us. A distribution B over $\mathbb{R}^d$ is radial with center c, if $p_B(x) = q(\|x - c\|)$ for some function $q : \mathbb{R}_+ \to \mathbb{R}_+$. If such a p_B is monotonic in $\|x - c\|$, i.e. q is monotonically decreasing, we say B is radial monotonic.

As discussed in Section 3, most flow-based anomaly detection architectures rely on various types of coupling layers Dinh et al. (2014; 2017); Kingma & Dhariwal (2018). Another crucial building block are bijective affine transforms Hoogeboom et al. (2019), which are often restricted to a subclass like (soft) permutations or Householder reflections Dinh et al. (2014); Tomczak & Welling (2016).

Uniformly Scaling Flows A core contribution of this work is to reveal a profound theoretical connection between deep one-class classification via the prominent Deep SVDD anomaly detection framework and a well-known subclass of normalizing flows. A *uniformly scaling flow (USF)* Dinh et al. (2014) is a diffeomorphism $\phi : \mathbb{R}^d \to \mathbb{R}^d$ whose Jacobian determinant is constant across the input space:

$$\det J_\phi(x) = \kappa \quad \forall x \in \mathbb{R}^d \tag{4}$$

for some constant $\kappa \in \mathbb{R} \setminus \{0\}$. This property induces uniform volume scaling across all regions of the latent space. Note that if $\phi = \phi_w$ is parameterized, then $|\det J_{\phi_w}(x)| = \psi_{\text{det}}^{\phi}(w)$ for some function ψ_{det}^{ϕ} that depends only on the parameters w. In the flow literature, the term volume preserving is often used synonymously with uniformly scaling. This is strictly speaking a misnomer, since the term actually refers to the special case $|\kappa| = 1$ Draxler et al. (2024). In practice, non-uniformly scaling architectures, like RealNVP Dinh et al. (2017), have largely superseded uniformly scaling variants like NICE Dinh et al. (2014) for their likelihood and sampling performance Kolesnikov et al. (2025), and also predominate flow-based anomaly detection (Section 3). However, stability analyses Behrmann et al. (2021) and recent evidence from tabular anomaly detection Kim et al. (2026) and neuro-symbolic verification Abu Zaid et al. (2026) suggest that USFs retain specific advantages.

In our analyses and experiments in Sections 4 and 5.1, we build our USFlows architecture using uniformly scaling layers from the CJD-Flows library[1]. For comparison, we implement NonUSFlows, which differs only in replacing additive coupling with affine coupling. This permits input-dependent Jacobian determinants Dinh et al. (2017), removing the uniform-scaling constraint and strictly increasing model expressivity Draxler et al. (2024). Full architectural details are provided in Appendix A.

[1] `https://github.com/USFlows/cjd-flows`

3 Related Work

Modern Deep One-class Classification Deep one-class methods, exemplified by Deep SVDD, aim to induce a typicality-preserving geometry, where distance from the nominal region reflects decreasing input-space density or manifold support. Still, Equation 2 provides no explicit mechanism for enforcing such alignment. Follow-up work introduced negative evidence, through synthesized near-manifold anomalies Goyal et al. (2020) or a small number of labeled outliers Ruff et al. (2020), or regularize embeddings using noise injection and batch-level variance penalties Chong et al. (2020). A complementary line preserves fidelity by coupling compactness with reconstruction or probabilistic regularization: DASVDD incorporates an autoencoder objective Hojjati & Armanfard (2024), while related variational formulations impose additional latent structure Zhou et al. (2021). Most recently, DOHSC constrains the one-class geometry through an orthogonal projection layer Zhang et al. (2024). In contrast to these auxiliary objectives and corrections, we show that maximum-likelihood training of a uniformly scaling flow directly yields a one-class objective with implicit density regularization, thereby avoiding representational collapse while inducing density-aligned latent geometry.

Normalizing Flows for Anomaly Detection Normalizing flows are widely used for anomaly detection. For image data, however, raw-input likelihoods can be dominated by low-level regularities Kirichenko et al. (2020), motivating flows on pretrained features Rudolph et al. (2021; 2022). CFlow, FastFlow, and U-Flow develop this paradigm through spatial conditioning, 2D feature-map flows, and multi-scale contextualization, respectively Gudovskiy et al. (2022); Yu et al. (2021); Tailanián et al. (2024). URD further combines reconstruction and density estimation via invertible mappings Wang et al. (2025). All these methods permit input-dependent Jacobian determinants. Yet, even perfect density estimation cannot guarantee successful anomaly detection Le Lan & Dinh (2021), Behrmann et al. (2021) derive more favorable bi-Lipschitz bounds and higher numerical stability on OOD inputs for additive than affine coupling, and recent evidence on tabular data highlights the potential of likelihood-based anomaly detection with USFs Kim et al. (2026). This motivates a central question: when and why is the inductive bias of uniform scaling preferable to additional expressivity?

Intersection of Normalizing Flows and Deep One-Class Classification Prior work combines normalizing flows with one-class reasoning, by imposing compactness losses on flow representations Sendera et al. (2021), learning flow-induced minimal-volume hyperspheres through a quantile-based criterion as in OneFlow Maziarka et al. (2022), or using flows to synthesize pseudo-anomalies Ryzhikov et al. (2021). While these methods highlight synergies between flows and one-class approaches, they rely on custom optimization criteria or additional training stages. Our work reveals a more fundamental connection. This precise theoretical bridge—showing the objective-level equivalence between these seemingly distinct paradigms—has been missing from the literature. Our work addresses: 1) when (and why) do normalizing flows replicate well-principled one-class approaches, and 2) how to leverage this intrinsic connection to improve flow-based anomaly detection.

4 Uniformly Scaling Flows for Deep One-Class Classification

4.1 Maximum Likelihood as One-Class Optimization

Recall that the one-class loss of a Deep SVDD given in Equation 2 consists of two terms: the square distance from the center $\|\omega(x, w) - c\|^2$ and a regularization term. Comparing this objective to the maximum likelihood training of normalizing flows reveals a close relationship: when training a uniformly scaling flow ϕ with base distribution $\mathcal{N}(c, I)$ and parameters w, the objective becomes:

$$\begin{aligned}\arg\min_w \mathbb{E}_{x\sim X}\left[-\ln \hat{p}_X^w(x)\right] &= \arg\min_w \mathbb{E}_{x\sim D}\left[-\ln(\mathcal{N}\left(\phi_w(x); c, I\right)\left|\det J_{\phi_w}(x)\right|)\right] \\ &= \arg\min_w \left(\mathbb{E}_{x\sim D}\left[\|\phi_w(x) - c\|^2\right] - 2\ln \psi_{\text{det}}^{\phi}(w)\right) \end{aligned} \tag{5}$$

Thus, for USFs the MLE objective decomposes into a center-distance term plus a *global* determinant term that depends only on parameters. This is the objective-level bridge: under constant Jacobian determinant, likelihood maximization is equivalent to learning compact one-class representations with an explicit volume regularizer. For non-uniformly scaling flows, this direct equivalence breaks.

The determinant contribution w.r.t. the same derivation on non-US flows becomes sample-dependent:

$$-2\mathbb{E}_{x\sim P_X}\left[\ln\left|\det J_{\phi_w}(x)\right|\right], \tag{6}$$

so it is no longer a pure regularization term in parameter space. Consequently, non-US training optimizes a trade-off: latent centrality and local volume change can be exchanged on a per-sample basis. This distinction is central to the latent representation alignment discussions in Section 4.2– 4.3.

The remaining key distinction to Deep SVDD is that uniformly scaling flows are diffeomorphisms and therefore preserve dimensionality between input and latent space. However, a USF with a radial monotonic base distribution implicitly defines an effective reduction to a one-dimensional score: the latent norm $\phi'_w(x) := \|\phi_w(x) - c\|$ produces an anomaly ranking identical to that of the likelihood.

4.2 Representation Collapse and Exploding Determinants

Deep one-class methods may exhibit *representation collapse*, i.e., latent embeddings that are weakly informative for anomaly scoring. The extreme case is hypersphere collapse, where many inputs map to nearly the same latent point and the objective is minimized trivially. In contrast, *exact* hypersphere collapse is impossible for normalizing flows, as they are bijective, hence distinguish distinct samples. Moreover, maximum likelihood training of flows minimizes the reverse Kullback-Leibler divergence,

$$\mathrm{KL}\left(P_{\phi(X)} \,\middle\|\, \mathcal{N}(c, I)\right) = \mathbb{E}_{x\sim P_X}[-\ln \hat{p}_w(X = x)] + C, \tag{7}$$

driving the transformed data distribution toward the base distribution P_B Papamakarios et al. (2021).

This holds for any base distribution and any flow architecture. Hence, collapse prevention is not a US-specific artifact; it is a consequence of likelihood-based bijective modeling. This also enforces a non-degenerate radial organization in latent space. However, transformation stability differs between US and non-US architectures. In non-US flows, the Jacobian determinant is input-dependent and composes multiplicatively across layers, $|\det J_\phi(x)| = \prod_{\ell=1}^{L} |\det J_{\phi_\ell}(h_{\ell-1}(x))|$, so locally large scale outputs in individual coupling blocks can amplify one another across depth, causing numerical instability Behrmann et al. (2021). Hence, it is common practice to constrain scale outputs (e.g., via clamping or $\tanh$-based restriction) to prevent extreme volume changes from destabilizing training.

In our architecture, volume preservation is enforced at the block level and the global volume change is determined by the final affine transformation; however, LU-based linear layers within each block can still realize local volume changes. For an LU parameterization $W = LU$, the Jacobian determinant satisfies $\det(W) = \det(L)\det(U)$, determined entirely by the diagonal scale terms. Placing a log-normal prior on these diagonal entries induces a log-normal prior on $|\det(W)|$, yielding an explicit probabilistic regularization of intra-block volume changes (see Appendix B.2 for a full derivation). The resulting MAP objective adds a volume regularization term that penalizes both intermediate and cumulative volume changes, refining the one-class regularization discussed in Section 4.1.

4.3 Density-Aligned Latent Space Representations

A remaining key deficiency is shared across both sides of the intersection: general deep one-class methods and non-US flows need not respect the input-density ordering. Lower-density inputs may receive more central latent representations than higher-density inputs, potentially inverting anomaly scores relative to the true data density. This failure mode is now formalized before stating the result.

We call a mapping $F : \mathcal{X} \to \mathcal{X}$ on a normed space $\mathcal{X}$ *density-aligned* with respect to p_X on $M \subseteq \mathcal{X}$ with center c , if for all $x, y \in M$ lower input density implies larger latent distance, and *density anti-aligned* if it implies smaller latent distance:

$$p_X(x) < p_X(y) \implies \begin{cases} \|F(x) - c\| > \|F(y) - c\|, & \text{density alignment,} \\ \|F(x) - c\| < \|F(y) - c\|, & \text{density anti-alignment.} \end{cases} \tag{8}$$

Proposition 1 (Density (anti-)alignment in one-class and flow models)**.**

1. ***Deep SVDD.*** *For $X \sim \mathcal{N}(0, I_d)$ with $d > 2$ and center c, there exists a family of mappings $F_\alpha \colon \mathbb{R}^d \to \mathbb{R}^d$, $\alpha > 0$, whose unregularized Deep SVDD loss converges to 0 as $\alpha \to \infty$, such that for every $\alpha > 0$, F_α is non-constant, its center distances $\|F_\alpha(x) - c\|$, $x \neq 0$, range over all of $(0, \infty)$, and F_α is density anti-aligned on $\mathbb{R}^d \setminus \{0\}$ with center c.*

2. ***Non-uniformly scaling flows.*** *For every $d \geq 2$ and $\epsilon \in (0, 1)$, there exist a smooth, strictly positive density p_X on $\mathbb{R}^d$ and a C^∞ diffeomorphism ϕ of $\mathbb{R}^d$ with $\phi_\# p_X = p_B = \mathcal{N}(0, I_d)$, together with a set M of probability $P_X(M) \geq 1 - \epsilon$, such that ϕ is density anti-aligned with center 0 on M.*

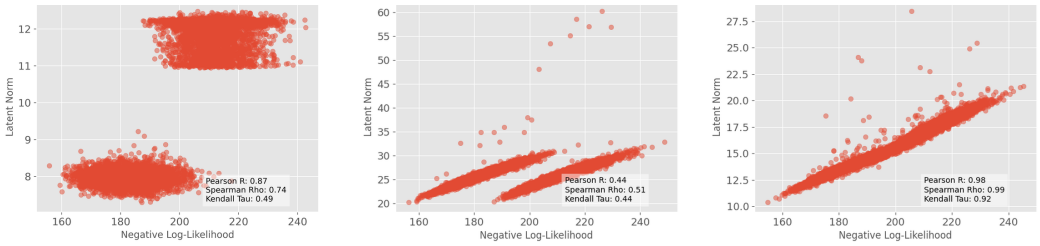
3. ***Uniformly scaling flows.*** *Let ϕ be a USF with $\phi_\# p_X = p_B$. Then $p_X(x) \propto p_B(\phi(x))$. If p_B is also radial monotonic, then ϕ is density aligned with center c on the entire space.*

Items 1 and 2 are derived in Appendix B.3; Item 3 is a direct consequence of the US property via Eq. 3: leaving no input-dependent correction available, the change-of-variables formula directly gives $p_X(x) \propto p_B(\phi(x))$. For any radial monotonic p_B, this implies density alignment with c, as in Eq. 8. Hence, for a fitted USF with a radially decreasing base, latent-distance ordering agrees with its model-density ordering Abu Zaid et al. (2026); agreement with the true density additionally depends on the quality of the fit.

Remark 1 (Scope of Items 1 and 2). Both constructions are worst-case witnesses, not results of typical training behavior. For Item 1, degeneration in the limit is unavoidable: any family driving the unregularized loss $\mathbb{E}|F(X) - c|^2$ to 0 converges to c in $L^2(P_X)$. The key point of the item is the *ordering before collapse*: every finite α yields strictly inverted rankings for inputs of different densities. Thus, excluding exact collapse does not by itself prevent misaligned near-collapse. For Item 2, complete anti-alignment on all of $\mathbb{R}^d$ cannot hold for any homeomorphism of $\mathbb{R}^d$ (see Appendix B.3), so the statement is necessarily probabilistic; the anti-aligned set can, however, carry any probability short of 1.

Hence, without constraints on the hypothesis space, neither the Deep SVDD loss nor MLE training of non-US flows provides an asymptotic guarantee of density-aligned latent representations. However, the failures in Items 1 and 2 of Proposition 1 are due to structurally different reasons. Deep SVDD has no change-of-variables correction and therefore no mechanism to couple its objective to the input-space density p_X: the loss directly minimizes distances to c, indifferent to which inputs are typical or atypical. Non-US flows do provide an exact likelihood through the Jacobian correction $|\det J_\phi(x)|$ in Eq. 3, so the data-space likelihood is exact. Still, precisely because this correction is input-dependent, the map is free to compensate latent density ordering with a matching local volume change: even at perfect likelihood match, the Jacobian term can absorb the density-order information and leave latent norms anti-aligned with input density except on an arbitrarily small fraction of data.

We demonstrate this practical advantage on an asymmetrical Gaussian mixture dataset, comparing the relationship between data density and latent norms across dimensions $d = 2, 8, 32, 128$. Figure 1 visualizes the differences in density alignment for $d = 2$, while Figure 2 provides an evaluation for $d = 128$ in terms of correlation metrics. Appendix B.3 details the exact setup and complete results.



(a) Deep SVDD (b) non-USF (c) USF

Figure 2: **Log-likelihood–norm alignment in Gaussian-mixture experiments for d=128.** The USF preserves a monotonic relation between latent norm and data density, while for non-USF and Deep SVDD this relationship is more susceptible to be broken, or not discovered in the first place.

4.4 Inductive Bias and Expressivity of Uniformly Scaling Flows

Uniform scaling trades increased transformation expressivity for density-aligned latent geometry. Although flexible local volume changes may benefit density estimation and sample generation, their value for anomaly detection must be assessed against the reliability of the resulting scores and the stability of optimization. This motivates evaluating uniform scaling as a task-specific inductive bias.

A USF may rearrange the data manifold nonlinearly and flexibly, but must do so under globally constant volume scaling. Since density amplification via local volume change is not available on a per-sample basis, concentrating in-distribution data near the center of a radial monotonic base distribution can only be achieved through geometric transport. Maximum likelihood training therefore implicitly displaces atypical or OOD inputs toward the outer shells. For anomaly detection, this is precisely the desired behavior: the constraint causes geometrically meaningful latent representations without special treatment of anomalies. This may further help reduce the influence of training contamination.

Even perfect likelihood matching of non-US flows provides no guarantee of density alignment with center c (Eq. 8), because input-dependent Jacobian corrections can absorb density-order information. Still, USFs offer less expressivity Draxler et al. (2024). Adopting general radial base distributions, however, can restore the representational capacity of the USF class Abu Zaid et al. (2026). Two further architectural choices can ensure high expressivity within this class. First, composing additive coupling layers with interleaved affine transforms extends the representable function class from the subgroup $\mathcal{C}$ of fixed-mask additive couplings to the union of its orbits under conjugation by affine diffeomorphisms, i.e., $\{A \circ f \circ A^{-1} : f \in \mathcal{C},\ A \text{ affine}\}$, while preserving the uniform-scaling property. Second, LU-based linear layers can realize feature mixing based on general, potentially non-volume-preserving affine maps, in contrast to Householder reflections, which are volume-preserving. The subsequent experiments assess the tradeoff between expressivity, stability, and detection performance.

5 Experiments

5.1 Experiments on Tabular Anomaly Detection

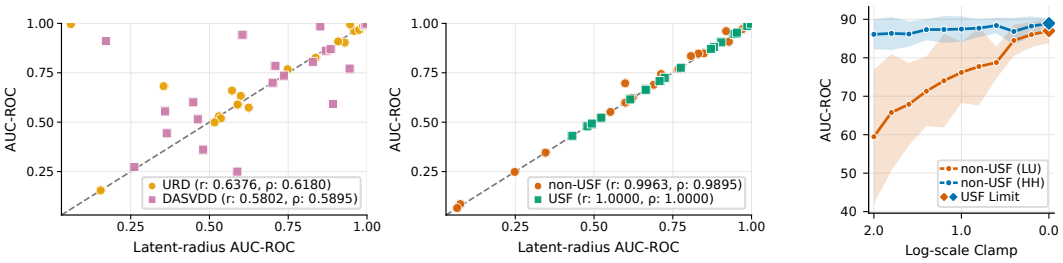
Experimental Setting Following recent work Wang et al. (2025), we evaluate a USF instantiated with the USFlow architecture on 20 tabular benchmarks from ADBench Han et al. (2022) and ODDS Rayana (2016), reporting AUC-ROC (%) mean and standard deviation across three runs per dataset. We use a 50/50 train–test split and report uncontaminated training, where anomalies assigned to training are moved to the test set, and contaminated training, where anomalies remain in their split. We compare USF against its direct non-USF counterpart and methods that isolate or combine the two sides of our analysis: Deep SVDD Ruff et al. (2018) and its follow-up DOHSC Zhang et al. (2024) as deep one-class baselines; DASVDD Hojjati & Armanfard (2024) and URD Wang et al. (2025), which combine deep one-class or flow-based modeling with reconstruction; and OneFlow / OneFlow-Gen Maziarka et al. (2022), which combine one-class and constant / non-constant-Jacobian flows without full density estimation. Hence, each method contains a scoring component that, under idealized alignment, makes the centered latent norm increase monotonically with anomaly evidence.

For USF, non-USF, and the OneFlow approaches, we adopt a shared configuration. We use 8 coupling blocks whose conditioners each consist of 5 linear layers with width 256. These methods are trained with Adam Kingma & Ba (2015) at learning rate 10^{-3} for up to 200 epochs with dataset dependent batch sizes following Wang et al. (2025). In the main results for USF and non-USF we forgo LU-parameterized feature-mixing for Householder reflections to avoid severe instabilities for non-USF, and set the log-scale clamping for its affine coupling to 2.0 (cf. Appendix E). For DOHSC, URD and DASVDD, we follow the architectures and training regimes from the original papers. For Deep SVDD we use the same encoder architecture as DOHSC. Appendix C gives additional details.

Main Results Table 1 highlights the advantages of the USF: it obtains the best mean AUC-ROC in both clean and contaminated training, while maintaining low variance. Compared with its non-USF counterpart, mean performance improves by 2.2 and 1.0 in the clean and contaminated regimes, and with lower variability, indicating that the constant-Jacobian constraint is indeed beneficial. This pairwise advantage is mirrored by comparing OneFlow against OneFlow-Gen, where the constant-Jacobian variant also performs stronger and substantially more stable under both settings.

Table 1: Aggregate AUC-ROC over 3 runs on the 20 tabular datasets with / without contamination.

		Clean		Contam.	
	Method	Avg.	Rank	Avg.	Rank
OC	DeepSVDD	82.1±3.9	5.05	<u>75.2±6.9</u>	4.05
	DOHSC	77.7±7.5	6.55	58.3±9.6	6.95
Hybrid	DASVDD	87.7±1.5	3.25	75.2±3.8	<u>3.50</u>
	OneFlow	<u>87.9</u>±1.4	<u>3.20</u>	74.9±5.0	4.25
	OneFlow-G	85.0±3.5	4.45	72.5±8.3	5.05
	URD	78.2±6.7	5.45	69.2±8.6	5.20
Flow	non-USF	86.1±3.9	4.90	74.7±6.3	3.95
	USF	**88.3**±1.5	**3.15**	**75.7**±4.2	**3.05**



(a) Relationship between latent-distance separation and detection performance. (b) Clamping sensitivity.

Figure 3: **Latent-space separation against performance and log-scale clamp sensitivity analysis.** (a) The x-axis reports the AUC-ROC obtained from the adjusted latent distance alone, while the y-axis reports the final anomaly-score AUC-ROC. For the remaining methods, the score is directly based on latent-radius. (b) The baseline corresponds to the clamp value used in the main experiments and further ablations; the zero-clamp endpoint is the uniformly scaling limit of the same architecture.

Figure 3a probes whether performance is encoded in the learned latent geometry. Just as for the OneFlow and conventional deep one-class approaches, for USF, the latent-radius directly coincides with the final anomaly-score, matching the theoretical argument that the score is radially monotone under a constant Jacobian. The non-USF remains strongly correlated but not identical, as its additional Jacobian term permits deviations from density alignment. Still, performance remains significantly lower than for its fully aligned USF counterpart. Both reconstruction hybrids show a much weaker correspondence, indicating that their final scores are less governed by latent-distance separation.

Ablation Study Figure 3b isolates the effect of the affine log-scale term. As the clamp is tightened, non-USF performance improves almost monotonically, and the zero-clamp endpoint recovers the uniformly scaling limit. Notably, commonly used affine-coupling implementations default to values around 2.0 Akcay et al. (2022). Table 2 shows that both the determinant prior and affine conjugation improve mean performance for both architectures; the former quite drastically. As Householder feature mixing yields ($|\det J_h| = 1$), for USF, the determinant prior acts on the final affine transform only. The full ablation in Appendix E further documents the instability of LU-based non-USFs. In pair-wise experiments, we also evaluate the impact of normalization (cf. Appendix C.4). Here, neither ActNorm Kingma & Dhariwal (2018) nor spectral normalization Miyato et al. (2018) narrow the performance gap, but rather increase it. Switching to a learnable radial base (cf. Appendix C.3) improved USF held-out density fit beyond non-USF, but without uniform gains in anomaly detection.

Table 2: Changes in avg. mean AUC-ROC relative to Table 1 in the uncontaminated setting.

Ablation		Method	
Prior	Aff. Conj.	non-USF	USF
✗	✓	−8.80	−2.95
✓	✗	−0.92	−0.17

5.2 Experiments on Image Anomaly Detection

Experimental Setting Flow-based architectures such as FastFlow Yu et al. (2021), CFlow Gudovskiy et al. (2022), and U-Flow Tailanián et al. (2024) achieve strong image anomaly detection performance, yet they rely on affine coupling layers that are *non-US*. We instantiate a USF layer by (1) replacing affine coupling with *additive* coupling and (2) inserting LU-parameterized bijective affine transforms with fixed diagonal magnitudes to recover per-block expressivity. This variant is evaluated against its affine counterpart across the MVTec AD Bergmann et al. (2019) and VisA Zou et al. (2022) datasets using the three aforementioned model architectures. These experiments were based on Anomalib Akcay et al. (2022) v2.0.0, using its PyTorch Lightning implementations. To isolate the practical impact of this substitution, we adopt the architecture and hyperparameter choices from the original papers and refrain from model- or class-specific tuning. Hence, within-architecture comparisons reflect the causal effect of the flow substitution under various conditions. Each model is trained for up to 100 epochs with batch size 32 using Adam, with early stopping after three epochs of no improvement. The feature extractor backbones are initialized and frozen with ImageNet-pretrained weights. For more details on the experimental setting and additional results, see Appendix D.

Table 3: Mean AUC-ROC and stdev. (%) over three seeds; dataset values are class-averaged. Highest mean per row in bold, second-highest underlined. Class-specific results are provided in Tables 10- 13.

	BASE FLOW			USF		
Dataset (Metric)	FastFlow	U-Flow	CFlow	FastFlow	U-Flow	CFlow
MVTec AD (Pixel)	95.8±0.4	95.9±2.8	95.4±1.0	96.8±0.2	**97.0±0.4**	96.6±0.0
MVTec AD (Image)	91.3±1.5	92.7±4.6	72.5±12.8	91.0±0.9	**94.5±0.6**	90.8±1.1
VisA (Pixel)	96.3±0.7	95.8±3.8	97.8±0.5	**98.8±0.0**	95.7±1.8	98.2±0.0
VisA (Image)	88.0±1.7	84.6±6.7	82.5±8.3	87.7±0.6	85.6±0.6	**88.4±0.9**

Main Results Table 3 summarizes how across datasets, replacing the base flow with uniformly scaling flows (USFs) improves mean performance or leaves it effectively unchanged, while consistently and substantially reducing run-to-run variance. This effect occurs at both image and pixel level.

Noticeably, USFs make the strongest impact where base flows are less stable: they markedly tighten variability (often approaching negligible standard deviations) and lift means most for CFlow and U-Flow, while FastFlow typically maintains its mean and still benefits from improved stability. On MVTec AD, USFs yield the most pronounced consistency gains and deliver the largest mean increase for CFlow; on VisA, the best row-wise means shift toward CFlow (image) and FastFlow (pixel) under USFs, with U-Flow remaining competitive and considerably more stable.

Ablation Study To disentangle which architectural change drives the stability and accuracy gains, we also analyze changes against the affine baseline and the full USF from the main experiments for an additive USF, which only swaps affine for additive coupling, whithout introducing the LU-parameterized general bijective affine transforms. Figure 4 reports the deviations in mean and standard deviation per-class on MVTec AD for pixel-level AUC-ROC (%). The image-level counterpart is given in Appendix D.3. Both USFs benefit CFlow compared to the non-USF version, while U-Flow and FastFlow in particular benefit from the LU transforms. These findings indicate that the uniformly-scaling modification is often the primary driver of improvements, while the general affine transforms are beneficial to recover expressivity without losing stability. For U-Flow, as the best-performing architecture on MVTec AD, we also evaluated the effect of introducing the LU transforms into the affine baseline. This led to significant instabilities, just as in the tabular experiments (cf. Appendix E).

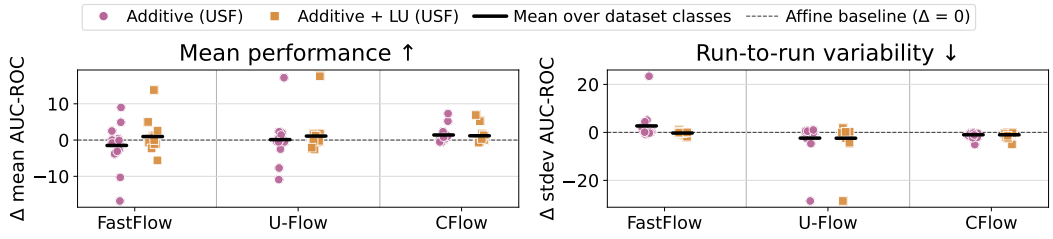


Figure 4: **Ablation on MVTec AD (Pixel).** Per-class changes in pixel AUC-ROC mean and stdev for replacing affine coupling with additive coupling, with / without LU-parameterized affine transforms.

6 CONCLUSION

This work establishes a formal bridge between deep one-class learning and maximum-likelihood training of uniformly scaling flows. USFs occupy the intersection of these paradigms: they retain the bijective density-modeling framework of flows while inducing the score-faithful centrality geometry sought by one-class methods. Empirically, replacing non-US flows with USFs yields a strong robustness–expressivity trade-off, often improving anomaly-detection performance while consistently stabilizing training. Several directions remain open. In particular, the role of the base distribution deserves closer study, including alternative radial families, learned norm distributions $P_{\|B\|}$, and heavier-tailed or multimodal reference measures. More broadly, our results suggest that for anomaly detection, controlling local volume changes is more valuable than maximizing density expressivity.

AI USE STATEMENT

We used generative AI tools, primarily models of the Claude Opus 5 series and, to a lesser extent, GPT, Gemini, DeepSeek, and GitHub Copilot, in interactive sessions for the following tasks requiring disclosure. (i) *Mathematical claims and proofs*: For Item 2 of Proposition 1, the statement, including the necessity of its probabilistic form, and the informal underlying idea of the proof are the authors' own. Generative AI was used interactively to work out from this idea the concrete construction, to formulate the supporting lemmas and refine their assumptions, to write their proofs, and for the numerical evaluation in Appendix B.3. All of these were independently rechecked by the authors and cross-checked. The remaining items of Proposition 1 and their proofs were developed without generative AI. Beyond this, we used AI for proof review and to check the consistency of definitions, statements, and terminology in the theoretical sections. (ii) *Methodology and results*: we used AI for feedback on the research methodology and experiments and for preliminary interpretation of results; the final interpretation is our own. (iii) *Implementation*: AI assistance was used to implement method components, the experimental pipeline, and analysis scripts, all according to our own specifications. This includes the code sampling the synthetic Gaussian-mixture data; the data themselves were not generated by AI. (iv) *Translation*: we used AI to translate parts of the text. We did not use generative AI to develop the research ideas, the theoretical framework, or the main claims, or to formulate hypotheses; the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI for language editing and proofreading throughout the manuscript, to search for additional relevant related work, and to create the legend icons of Figure 1; the experimental data displayed in the figures stems from our own experiments. The authors reviewed all AI-assisted material, checked the mathematical arguments, and reviewed the generated code. We take full responsibility for the final content of this work, including text, claims, proofs, code, and figures produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Architectural specifications are provided in Appendix A, and the assumptions, derivations, and proofs supporting Section 4 are detailed in Appendix B. Sections 5.1–5.2 and Appendices C–D describe the tabular and image experimental protocols and results in detail along with the additional radial-base and normalization studies. Appendix E documents observed training instabilities, before Appendix F specifies the execution environment and Appendix G concludes by outlining limitations of our work. We are including the source code used to conduct our experiments along with their configuration files as supplementary material.

REFERENCES

Faried Abu Zaid, Daniel Neider, and Mustafa Yalçıner. Veriflow: Modeling distributions for neural network verification. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence AAAI*, pp. 28050–28058. AAAI Press, 2026. doi: 10.1609/AAAI.V40I33.40030. URL https://doi.org/10.1609/aaai.v40i33.40030.

Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh A. Ahuja, and Ergin Utku Genc. Anomalib: A deep learning library for anomaly detection. In *IEEE International Conference on Image Processing, ICIP*, pp. 1706–1710. IEEE, 2022. doi: 10.1109/ICIP46576.2022.9897283. URL https://doi.org/10.1109/ICIP46576.2022.9897283.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1792–1800. PMLR, 2021.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec AD - A comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 9592–9600. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00982. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Bergmann_MVTec_AD_--_A_Comprehensive_Real-World_Dataset_for_Unsupervised_Anomaly_CVPR_2019_paper.html.

Robin Kien-Wei Chan, Sarina Penquitt, and Hanno Gottschalk. LU-Net: Invertible Neural Networks Based on Matrix Factorization. In *2023 International Joint Conference on Neural Networks ,IJCNN*, pp. 1–10. IEEE, 2023. ISBN 978-1-66548-867-9. URL https://pub.uni-bielefeld.de/record/2982070.

Penny Chong, Lukas Ruff, Marius Kloft, and Alexander Binder. Simple and effective prevention of mode collapse in deep one-class classification. In *2020 International Joint Conference on Neural Networks, IJCNN*, pp. 1–9. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9207209. URL https://doi.org/10.1109/IJCNN48605.2020.9207209.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. *CoRR*, abs/1410.8516, 2014. URL http://arxiv.org/abs/1410.8516.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR*, 2017. doi: 10.48550/arXiv.1605.08803. URL https://arxiv.org/abs/1605.08803.

Felix Draxler, Stefan Wahl, Christoph Schnoerr, and Ullrich Koethe. On the universality of volume-preserving and coupling-based normalizing flows. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11613–11641. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/draxler24a.html.

Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys, CSUR*, 54(7): 1–37, 2021. doi: 10.1145/3464423. URL https://doi.org/10.1145/3464423.

Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 1999.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. DROCC: deep robust one-class classification. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3711–3721. PMLR, 2020. URL http://proceedings.mlr.press/v119/goyal20c.html.

Denis A. Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, pp. 1819–1828. IEEE, 2022. doi: 10.1109/WACV51458.2022.00188. URL https://doi.org/10.1109/WACV51458.2022.00188.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35:32142–32159, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cf93972b116ca5268827d575f2cc226b-Abstract-Datasets_and_Benchmarks.html.

Hadi Hojjati and Narges Armanfard. DASVDD: deep autoencoding support vector data descriptor for anomaly detection. *IEEE Trans. Knowl. Data Eng.*, 36(8):3739–3750, 2024. doi: 10.1109/TKDE.2023.3328882. URL https://doi.org/10.1109/tkde.2023.3328882.

Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows. *International Conference on Machine Learning ICML*, pp. 2771–2780, 2019. URL https://arxiv.org/abs/1901.11137.

Donghwan Kim, Junghun Phee, and Hyunsoo Yoon. Why the counterintuitive phenomenon of likelihood rarely appears in tabular anomaly detection with deep generative models? *CoRR*, abs/2602.09593, 2026. doi: 10.48550/ARXIV.2602.09593. URL https://doi.org/10.48550/arXiv.2602.09593.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems, Neurips*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems, Neurips*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ecb9fe2fbb99c31f567e9823e884dbec-Abstract.html.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2021. URL https://ieeexplore.ieee.org/abstract/document/9089305.

Alexander Kolesnikov, André Susano Pinto, and Michael Tschannen. Jet: A modern transformer-based normalizing flow. *Transactions on Machine Learning Research*, 2025. arXiv preprint arXiv:2412.15129.

Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12):1690, 2021.

Jiaqi Liu, Guoyang Xie, Jin-Bao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Mach. Intell. Res.*, 21(1): 104–135, 2024. doi: 10.1007/S11633-023-1459-Z. URL https://doi.org/10.1007/s11633-023-1459-z.

Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard Hovy. MaCow: Masked convolutional generative Flow. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 5893–5902. Curran Associates Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/20c86a628232a67e7bd46f76fba7ce12-Abstract.html.

Lukasz Maziarka, Marek Smieja, Marcin Sendera, Lukasz Struski, Jacek Tabor, and Przemyslaw Spurek. Oneflow: One-class flow for anomaly detection based on a minimal volume region. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):8508–8519, 2022. doi: 10.1109/TPAMI.2021.3108223. URL https://doi.org/10.1109/TPAMI.2021.3108223.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22:57:1–57:64, 2021. URL https://jmlr.org/papers/v22/19-1028.html.

Shebuti Rayana. Odds library. stony brook university, department of computer science, stony brook, ny, 2016.

Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, pp. 1906–1915. IEEE, 2021. doi: 10.1109/WACV48630.2021.00195. URL https://doi.org/10.1109/WACV48630.2021.00195.

Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, pp. 1829–1838. IEEE, 2022. doi: 10.1109/WACV51458.2022.00189. URL https://doi.org/10.1109/WACV51458.2022.00189.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 2018. URL http://proceedings.mlr.press/v80/ruff18a.html.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proc. IEEE*, 109(5):756–795, 2021. doi: 10.1109/JPROC.2021.3052449. URL https://doi.org/10.1109/JPROC.2021.3052449.

Artem Ryzhikov, Maxim Borisyak, Andrey Ustyuzhanin, and Denis Derkach. NFAD: fixing anomaly detection using normalizing flows. *PeerJ Comput. Sci.*, 7:e757, 2021. doi: 10.7717/PEERJ-CS.757. URL https://doi.org/10.7717/peerj-cs.757.

Marcin Sendera, Marek Smieja, Lukasz Maziarka, Lukasz Struski, Przemyslaw Spurek, and Jacek Tabor. Flow-based SVDD for anomaly detection. *CoRR*, abs/2108.04907, 2021. URL https://arxiv.org/abs/2108.04907.

Matías Tailanián, Álvaro Pardo, and Pablo Musé. U-flow: A u-shaped normalizing flow for anomaly detection with unsupervised threshold. *J. Math. Imaging Vis.*, 66(4):678–696, 2024. doi: 10.1007/S10851-024-01193-Y. URL https://doi.org/10.1007/s10851-024-01193-y.

Jakub M. Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *CoRR*, abs/1611.09630, 2016. URL http://arxiv.org/abs/1611.09630.

Xiaolei Wang, Tianhong Dai, Huihui Bai, Yao Zhao, and Jimin Xiao. Unifying reconstruction and density estimation via invertible contraction mapping in one-class classification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hassler Whitney. Differentiable even functions. *Duke Mathematical Journal*, 10(1):159–160, 1943.

Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *CoRR*, abs/2111.07677, 2021. URL https://arxiv.org/abs/2111.07677.

Yunhe Zhang, Yan Sun, Jinyu Cai, and Jicong Fan. Deep orthogonal hypersphere compression for anomaly detection. In *The Twelfth International Conference on Learning Representations, ICLR*. OpenReview.net, 2024. URL https://openreview.net/forum?id=cJs4oE4m9Q.

Zheng Zhang and Xiaogang Deng. Anomaly detection using improved deep SVDD model with data structure preservation. *Pattern Recognit. Lett.*, 148:1–6, 2021. doi: 10.1016/J.PATREC.2021.04.020. URL https://doi.org/10.1016/j.patrec.2021.04.020.

Yu Zhou, Xiaomin Liang, Wei Zhang, Linrang Zhang, and Xing Song. Vae-based deep SVDD for anomaly detection. *Neurocomputing*, 453:131–140, 2021. doi: 10.1016/J.NEUCOM.2021.04.089. URL https://doi.org/10.1016/j.neucom.2021.04.089.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Computer Vision - ECCV 2022 - 17th European Conference*, volume 13690 of *Lecture Notes in Computer Science*, pp. 392–408. Springer, 2022. doi: 10.1007/978-3-031-20056-4_23. URL https://doi.org/10.1007/978-3-031-20056-4_23.

A ARCHITECTURAL SPECIFICATIONS (SECTION 2)

In the following, we use $m \odot x$ to denote the component wise (Hadamard) product between two tensors of same topology.

USFlows Architecture According to Abu Zaid et al. (2026), the USFlow architecture is built from masked additive coupling layers Dinh et al. (2014); Ma et al. (2019) and bijective affine transformations parameterized by LU-decomposed matrices Chan et al. (2023).

A masked additive coupling layer $C : \mathbb{R}^d \to \mathbb{R}^d$ (or any other fixed tensor topology) defines a bijection based on an arbitrary *conditioner* function $f : \mathbb{R}^d \to \mathbb{R}^d$ and a $\{0,1\}$-mask m as $C(x) = x + (1-m) \odot f(m \odot x)$. By construction, $m \odot x = m \odot C(x)$ and hence $C^{-1}(y) = y - (1-m) \odot f(m \odot y)$.

An LU layer computes a general bijective affine transform $A(x) = Mx + b$. Bijectivity is guaranteed by imposing constraints on the linear-transform parameterization: $M = LU$, where L/U are lower/upper triangular matrices with $L_{ii} = 1$ and $U_{ii} \neq 0$ for all indices i.

In order to allow uniform handling of different input topologies, USFlows employs one-star-convolutions (convolutions with kernel size 1 along all spatial dimensions, adapted to input topology) for the bijective affine transforms, where an LU-decomposed channel transform ensures bijectivity. For the conditioner of the coupling transforms, general adaptive convolutions (convolutions with homogeneous kernel size along all spatial dimensions, adapted to input topology) with layer normalization and gating are employed.

A USFlow model is built from blocks of shape $B_i = A_i^{-1} \circ C_i \circ A_i$, where A_i is a bijective affine transform and C is an additive coupling layer. The complete flow is of shape $\phi = A_{n+1} \circ B_n \circ B_{n-1} \circ \cdots \circ B_1$, where the final affine transform ensures that the flow can have an arbitrary (constant) Jacobian determinant. Note that each block is volume preserving since determinants of A_i and A_i^{-1} cancel each other and additive coupling layers are volume preserving by design Dinh et al. (2014).

NonUSFlows Architecture The NonUSFlows architecture is identical except that we use masked affine coupling instead of additive coupling for the blocks.

A masked affine coupling layer $C : \mathbb{R}^d \to \mathbb{R}^d$ (or any other fixed tensor topology) defines a bijection based on an arbitrary additive conditioner function $f : \mathbb{R}^d \to \mathbb{R}^d$, a multiplicative conditioner function $g : \mathbb{R}^d \to \mathbb{R}^d$ with $g(x)_i \neq 0$ for all $x \in \mathbb{R}^d$, $i \in \{1, \ldots, d\}$ and a $\{0,1\}$-mask m as $C(x) = m \odot x + (1-m) \odot g(m \odot x)x + (1-m) \odot f(m \odot x)$. Again, since $m \odot x = m \odot C(x)$, the inverse is given by $C^{-1}(y) = m \odot y + (1-m) \odot \frac{y - (1-m) \odot f(m \odot y)}{g(m \odot y)}$. Note that an affine coupling layer with $g(x) = 1$ for all x is equivalent to an additive coupling layer.

Allowing arbitrary non-zero outputs for g can be numerically unstable in practice (locally exploding determinants). Therefore, we adopt the common practice to apply a final scaled $\tanh$ activation with a configurable scaling parameter to our conditioner network g. Moreover, f and g are implemented by a single conditioner that computes both outputs, which further unifies the interface of both models. In all our experiments, the parametrization of the conditioner used in USFlows and NonUSFlows differs only in the final projection layer to produce the required output shapes, respectively.

B Formal Proofs and Omitted Details (Section 4)

B.1 Details Omitted from Section 4.1

Derivation of Equation 5

$$\begin{aligned}
\arg\min_w \mathbb{E}_{x\sim X}\left[-\ln \hat{p}_X^w(x)\right] &= \arg\min_w \mathbb{E}_{x\sim D}\left[-\ln(\mathcal{N}\left(\phi_w(x); c, I\right)\left|\det J_{\phi_w}(x)\right|)\right] \\
&= \arg\min_w \mathbb{E}_{x\sim D}\left[\begin{array}{c} -\ln\exp\left(-\dfrac{(\phi_w(x)-c)^\top I(\phi_w(x)-c)}{2}\right) \\ -\ln\left|\det J_{\phi_w}(x)\right| \end{array}\right] \\
&= \arg\min_w \mathbb{E}_{x\sim D}\left[\frac{1}{2}(\phi_w(x)-c)^T(\phi_w(x)-c) - \ln\left|\det J_{\phi_w}(x)\right|\right] \\
&= \arg\min_w \left(\frac{1}{2}\mathbb{E}_{x\sim D}\left[\|\phi_w(x)-c\|^2\right] - \mathbb{E}_{x\sim D}\left[\ln\left|\det J_{\phi_w}(x)\right|\right)\right] \\
&= \arg\min_w \left(\mathbb{E}_{x\sim D}\left[\|\phi_w(x)-c\|^2\right] - 2\mathbb{E}_{x\sim D}\left[\ln\left|\det J_{\phi_w}(x)\right|\right)\right] \quad (6) \\
&\overset{\phi \text{ US}}{=} \arg\min_w \left(\mathbb{E}_{x\sim D}\left[\|\phi_w(x)-c\|^2\right] - 2\ln\psi_{\text{det}}^{\phi}(w)\right) \quad (5)
\end{aligned}$$

The chain drops the additive normalization constant of $\ln\mathcal{N}\left(\phi_w(x); c, I\right)$ and rescales the objective by 2. Both operations preserve the minimizer but not the value of the minimum, which is why the equalities are stated for the $\arg\min$.

B.2 Details Omitted from Section 4.2

Derivation of Equation 7 We derive the fact that maximum likelihood training of a normalizing flow is equivalent to minimizing the reverse KL-divergence between the latent data distribution and the base distribution Papamakarios et al. (2021).

$$\begin{aligned}
\mathrm{KL}\left(P_{\phi(X)} \,\middle\|\, \mathcal{N}\left(c, I\right)\right) &= \mathbb{E}_{x\sim P_X}\left[\ln\frac{p_X(x)\cdot\left|\det J_\phi(x)\right|^{-1}}{\mathcal{N}\left(\phi(x); c, I\right)}\right] \\
&= \underbrace{\mathbb{E}_{x\sim P_X}\left[\ln p_X(x)\right]}_{\text{const.}} - \mathbb{E}_{x\sim P_X}\left[\ln\mathcal{N}\left(\phi(x); c, I\right) + \ln\psi_{\text{det}}^{\phi}\right] \\
&= \mathbb{E}_{x\sim P_X}\left[-\ln\hat{p}_w(X=x)\right] + C,
\end{aligned}$$

where the last expression is obtained by applying the logarithmic version of the change of variables formula to the middle term of the previous expression.

Implicit Determinant Regularization We show that it is possible to define a prior on the weights of an USFlow such that the induced prior on the determinant of the transformation is log-normal.

The **bilateral log-normal** is defined on $\mathbb{R}$ as $\text{BiLogNormal}(x;\mu,\sigma^2) = \frac{1}{2}\text{LogNormal}(|x|;\mu,\sigma^2)$, where $\text{LogNormal}(x;\mu,\sigma^2) = \frac{1}{x\sigma\sqrt{2\pi}}\exp-\frac{(\ln x-\mu)^2}{2\sigma^2}$ is the usual log-normal distribution defined on $\mathbb{R}_{\geq 0}$. Note that if $X \sim \text{BiLogNormal}(\mu,\sigma^2)$ then $|X| \sim \text{LogNormal}(\mu,\sigma^2)$ and $\ln|X| \sim \mathcal{N}(\mu,\sigma^2)$.

In the following, we carry out the derivation for image topologies, i.e., input tensors of shape Channels $\times$ Height $\times$ Width. The derivation for other input topologies is completely analogous. Now recall that the USFlows Architecture is built from additive coupling and affine transforms. Additive coupling is volume preserving by design, which means that we only need to take care of the affine transforms. The affine transforms of a USFlow are given by bijective 1×1 convolutions with LU decomposed bijective channel transforms. Let C be a 1×1 convolution with channel transform $A = LU$ for lower/upper triangular matrices L/U with $L_{ii} = 1$ and $U_{ii} \neq 0$ for all i. For a fixed image topology $c\times h\times w$, $|\det J_C(x)| = |\det A|^{hw}$. Further, since $|\det A| = \prod_{i=1}^c |U_{ii}|$, we impose an independent bilateral log-normal prior $\text{BiLogNormal}(0,\sigma_0^2)$ with $\sigma_0 = \sigma/(hw\sqrt{c})$ on each $u_i := U_{ii}$. This yields multiplicative regularization:

$$p\left(|\det J_C(x)|\right) = p\left(\prod\nolimits_{i=1}^c |u_i|^{hw}\right) = p\left(\exp\left(hw\sum\nolimits_{i=1}^c \ln|u_i|\right)\right).$$

Since $p(\ln|u_i|) = \mathcal{N}(0, \sigma_0^2)$ and all u_i are independent, the sum below is Gaussian with variance $(hw)^2 c\,\sigma_0^2 = \sigma^2$:

$$hw \sum_{i=1}^{c} \ln|u_i| \sim \mathcal{N}(0, \sigma^2) \implies |\det J_C(x)| \sim \text{LogNormal}(0, \sigma^2).$$

Variance control via σ_0 thus regularizes determinant scale.

Note that it is sufficient to regularize the affine transforms individually to the desired scale because, except for the last affine transform, all affine transforms appear as adjoint actions $C^{-1} \circ F \circ C$ applied to an additive coupling layer F and therefore determinants cancel block-wise. This prior regularizes the overall transformation as well as the inter-block transforms in a uniform way. Note that we did not observe a need to regularize the remaining parameters of the network in our experiments.

B.3 Details Omitted from Section 4.3

Proposition 1, Item 1 Let $\mathcal{N}$ be a d-dimensional standard normal distribution with $d > 2$ and let $c \in \mathbb{R}^d$. Then there exists a family of functions $F_\alpha : \mathbb{R}^d \to \mathbb{R}^d$, $\alpha \in \mathbb{R}^+$, such that $\lim_{\alpha\to\infty} L(\mathcal{N}, F_\alpha) = 0$, where L is the Deep SVDD loss (without regularization) with center c, and such that for every $\alpha > 0$:

(i) F_α is non-constant, and $\{\|F_\alpha(x) - c\| : x \neq 0\} = (0, \infty)$;

(ii) for all $x, y \in \mathbb{R}^d$ with $x, y \neq 0$,

$$\mathcal{N}(x) < \mathcal{N}(y) \iff \|F_\alpha(x) - c\| < \|F_\alpha(y) - c\|,$$

i.e., the anomaly ranking induced by F_α is strict and completely inverted with respect to the data density.

Proof. Fix a unit vector $e \in \mathbb{R}^d$ and define

$$F_\alpha(x) = c + \frac{e}{\alpha\|x\|} \quad (x \neq 0), \qquad F_\alpha(0) = c,$$

so that $\|F_\alpha(x) - c\| = e/(\alpha\|x\|)$ for $x \neq 0$. This distance takes every value in $(0, \infty)$ as $\|x\|$ ranges over $(0, \infty)$, so in particular F_α is non-constant. Since $\mathcal{N}(x)$ is strictly decreasing in $\|x\|$, we have $\mathcal{N}(x) < \mathcal{N}(y) \iff \|x\| > \|y\| \iff \|F_\alpha(x) - c\| < \|F_\alpha(y) - c\|$ for $x, y \neq 0$, which is (ii).

The overall loss for a function of this class is

$$L(\mathcal{N}, F_\alpha) = E_{x\sim\mathcal{N}}\left[\|F_\alpha(x) - c\|^2\right] = E_{x\sim\mathcal{N}}\left[\left(\frac{1}{\alpha\|x\|}\right)^2\right] = \frac{1}{\alpha^2} E_{x\sim\mathcal{N}}\left[\frac{1}{\|x\|^2}\right].$$

Since x is a vector in $\mathbb{R}^d$ where each component x_i is an i.i.d. standard normal random variable, i.e. $x_i \sim \mathcal{N}(0, 1)$, it follows that $\|x\|^2$ is distributed according to a chi-squared distribution with d degrees of freedom, i.e. $\|x\|^2 \sim \chi^2(d)$. Therefore, for $d > 2$, the expected value simplifies to $E_{x\sim\mathcal{N}}\left[\frac{1}{\|x\|^2}\right] = \frac{1}{d-2}$. Substituting into the loss gives $L(\mathcal{N}, F_\alpha) = \frac{1}{\alpha^2(d-2)}$, and thus $\lim_{\alpha\to\infty} L(\mathcal{N}, F_\alpha) = 0$. □

Discussion As $\alpha \to \infty$, $F_\alpha \to c$ in $L^2(\mathcal{N})$, i.e., the family degenerates in the limit. This is not specific to the construction: $L(\mathcal{N}, F) = \mathbb{E}\|F(X) - c\|^2$, so any family whose loss tends to 0 converges to c in L^2. What the example isolates is the direction of this degeneration. Common collapse-prevention mechanisms (bias removal, bounded activations, radius optimization, orthogonal projection) constrain the *magnitude* of collapse, whereas F_α is completely anti-aligned at every finite α: the one-class objective exerts no force along the alignment direction. In contrast to Item 3, preventing exact collapse therefore does not prevent misaligned near-collapse.

We choose to ignore the regularization term in our example because we assume such degenerate solutions to be approximated by complex neural networks and the influence of the regularization term w.r.t. the parameterization given in the example can be misleading. For the sake of completeness, however, we would still like to mention that it is not hard to adjust the example such that we can meet arbitrarily small losses with the optimal solution under the Deep SVDD loss function, even if the standard regularization is applied to the given parameters.

Proposition 1, Item 2 We show that exact transport to the base distribution does not by itself imply density-aligned latent representations: even a flow that attains the optimal likelihood exactly ($\phi_{\#}p_X = p_B$, hence $\mathrm{KL} = 0$) can be density anti-aligned on all but an arbitrarily small fraction of the data. Consequently, without additional constraints on the hypothesis space, MLE training of flows provides no asymptotic guarantee that learned representations are faithful to the input-space density structure. This stands in contrast to USFs, whose constant Jacobian determinant inherently provides density alignment (Item 3).

WHY THE STATEMENT IS PROBABILISTIC. Complete anti-alignment on all of $\mathbb{R}^d$ is impossible for any flow, which is why Item 2 asserts anti-alignment on a set of probability at least $1 - \epsilon$ rather than everywhere.

Lemma 1. *Let p_X be a probability density on $\mathbb{R}^d$ and ϕ a homeomorphism of $\mathbb{R}^d$. Then ϕ is not density anti-aligned on $\mathbb{R}^d$ with any center c.*

Proof. Pick y with $p_X(y) > 0$. The set $\{x : p_X(x) \geq p_X(y)\}$ has Lebesgue measure at most $1/p_X(y)$, so its complement is unbounded and contains a sequence x_n with $\|x_n\| \to \infty$ and $p_X(x_n) < p_X(y)$. Anti-alignment would give $\|\phi(x_n) - c\| < \|\phi(y) - c\|$ for all n. But a homeomorphism of $\mathbb{R}^d$ is proper, so $\|\phi(x_n)\| \to \infty$, a contradiction. □

In words, Arbitrarily remote low-density points exist, and every sequence escaping to infinity must also escape under a homeomorphism, which is alignment rather than anti-alignment. The construction below confines this unavoidable aligned region to a set of arbitrarily small probability.

CONSTRUCTION. Fix $d \geq 2$, $0 < \beta < \lambda$, and set

$$p_B = \mathcal{N}(0, I_d), \qquad p_X(x) = \frac{1}{Z} f\big(\|x\|^2\big), \qquad f(u) = \exp\big(\beta u - e^{\lambda(u-1)}\big), \tag{9}$$

with normalizing constant Z. The density p_X is C^∞ (a smooth function of $\|x\|^2$), strictly positive, and integrable. It is a smooth "crater": since $\frac{d}{du}\big(\beta u - e^{\lambda(u-1)}\big) = \beta - \lambda e^{\lambda(u-1)}$, its radial profile strictly *increases* for $\|x\| < R$ and then falls off a doubly exponential cliff, where

$$R^2 = u^* := 1 + \frac{1}{\lambda} \ln \frac{\beta}{\lambda} \tag{10}$$

(we assume $u^* > 0$, which holds for all parameters used below). Let $F_{\|X\|}$ and $F_{\|B\|}$ denote the CDFs of $\|X\|$ and $\|B\|$ for $X \sim p_X$, $B \sim p_B$, and let ϕ be the radial quantile transport

$$\phi(x) = h(\|x\|)\frac{x}{\|x\|} \;\; (x \neq 0), \qquad \phi(0) = 0, \qquad h = F^{-1}_{\|B\|} \circ F_{\|X\|}. \tag{11}$$

Lemma 2. *For all $0 < \beta < \lambda$ with $u^* > 0$, the map ϕ in equation 11 satisfies:*

(i) ϕ is a C^∞ diffeomorphism of $\mathbb{R}^d$ with $0 < |\det J_\phi(x)| < \infty$ for all x;

(ii) $\phi_{\#}p_X = p_B$, hence $\mathrm{KL}(P_{\phi(X)} \,\|\, P_B) = 0$;

(iii) ϕ is not uniformly scaling;

(iv) ϕ is density anti-aligned with center 0 on the closed ball $M = \{x : \|x\| \leq R\}$.

Proof. (i) Substituting $s = rt$ in the radial integral gives $F_{\|X\|}(r) = \frac{|S^{d-1}|}{Z}\int_0^r s^{d-1} f(s^2)\, ds = r^d G(r^2)$ with $G(v) = \frac{|S^{d-1}|}{Z}\int_0^1 t^{d-1} f(vt^2)\, dt$, a smooth and strictly positive function on $\mathbb{R}$. Likewise $F_{\|B\|}(r) = r^d H(r^2)$ with H smooth and strictly positive. The functions $\Theta(r) = r\, G(r^2)^{1/d}$ and $\Psi(s) = s\, H(s^2)^{1/d}$ are odd and smooth on $\mathbb{R}$. On $(0, \infty)$ they coincide with $F^{1/d}_{\|X\|}$ and $F^{1/d}_{\|B\|}$, which have strictly positive derivatives there, and $\Theta'(0) = G(0)^{1/d} > 0$, $\Psi'(0) = H(0)^{1/d} > 0$. Hence both are C^∞ diffeomorphisms of $\mathbb{R}$ onto $(-1, 1)$, and $h = \Psi^{-1} \circ \Theta$ on $(0, \infty)$ extends to an odd C^∞ diffeomorphism of $\mathbb{R}$ with $h' > 0$. For an odd smooth h, $h(r)/r$ is an even smooth function, hence of the form $k(r^2)$ with k smooth (Whitney, 1943); moreover $k > 0$ since $k(0) = h'(0) > 0$ and

$h(r)/r > 0$ for $r \neq 0$. Therefore $\phi(x) = k(\|x\|^2)\, x$ is C^∞ on $\mathbb{R}^d$, and the same argument applied to h^{-1} shows that its inverse $z \mapsto h^{-1}(\|z\|)\, z/\|z\|$ is C^∞ as well. In particular $\det J_\phi(x) \neq 0$ everywhere, and by *(ii)* and the change-of-variables formula, $|\det J_\phi(x)| = p_X(x)/p_B(\phi(x)) \in (0, \infty)$.

(ii) Since p_X is radial, $X = \|X\|\,\Omega$ with Ω uniform on the unit sphere and independent of $\|X\|$. Then $\phi(X) = h(\|X\|)\,\Omega$, and since $F_{\|X\|}$ is continuous, $F_{\|X\|}(\|X\|)$ is uniform on $(0, 1)$, so $h(\|X\|)$ has CDF $F_{\|B\|}$. Hence $\phi(X)$ has the same radial and angular laws as B, i.e., $\phi_\# p_X = p_B$.

(iii) On M, $p_X(x)$ is strictly increasing in $\|x\|$, whereas $p_B(\phi(x)) = (2\pi)^{-d/2} \exp(-h(\|x\|)^2/2)$ is strictly decreasing in $\|x\|$. Hence $|\det J_\phi(x)| = p_X(x)/p_B(\phi(x))$ is strictly increasing in $\|x\|$ on M and in particular not constant: the Jacobian inflates by exactly enough to invert the density ordering, which a constant determinant cannot do.

(iv) For $x, y \in M$, the monotonicity statements in *(iii)* and the strict monotonicity of h give

$$p_X(x) < p_X(y) \iff \|x\| < \|y\| \iff \|\phi(x)\| < \|\phi(y)\| \iff p_B(\phi(x)) > p_B(\phi(y)). \quad \square$$

PROBABILITY OF THE ANTI-ALIGNED SET. It remains to show that M can be made to carry probability at least $1 - \epsilon$. The following choice of parameters suffices; β does not depend on ϵ at all.

Lemma 3. *Let $d \geq 2$, $\epsilon \in (0, 1)$, and set*

$$\beta = 1, \qquad \lambda = \frac{200\, d^2}{\epsilon^2} \tag{12}$$

in equation 9. Then $P_X(\|X\| > R) \leq \epsilon/2$.

Proof. Write $u = \|x\|^2$. The random variable $U = \|X\|^2$ has density proportional to $g(u) = u^{d/2-1} \exp(\beta u - e^{\lambda(u-1)})$ on $(0, \infty)$, and $P_X(\|X\| > R) = P(U > u^*)$. With equation 12 we have $\lambda \geq 800$, hence $u^* = 1 - \lambda^{-1} \ln \lambda \in [0.99, 1)$.

Outer bound. Put $t = u - u^* \geq 0$. Since $e^{\lambda(u^*-1)} = \beta/\lambda$, the exponent equals $\beta u^* + \beta t - \frac{\beta}{\lambda} e^{\lambda t}$, and $e^{\lambda t} \geq 1 + \lambda t + \frac{(\lambda t)^2}{2}$ gives $\beta t - \frac{\beta}{\lambda} e^{\lambda t} \leq -\frac{\beta}{\lambda} - \frac{\beta \lambda t^2}{2}$. Using $(u^* + t)^{d/2-1} \leq (u^*)^{d/2-1} e^{(d/2-1)t/u^*}$ (valid for $d \geq 2$), extending the integral to $t \in \mathbb{R}$, and completing the square,

$$\int_{u^*}^{\infty} g \;\leq\; (u^*)^{d/2-1} e^{\beta u^* - \beta/\lambda} \sqrt{\frac{2\pi}{\beta\lambda}}\, \exp\Big(\frac{(d/2-1)^2}{2u^{*2}\beta\lambda}\Big).$$

Inner bound. Let $\kappa = \beta + \frac{d-2}{u^*}$ and $\eta = \min\{1/\kappa,\, u^*/2\}$. On $[u^* - \eta, u^*] \subseteq [u^*/2, u^*]$, the derivative of $\ln(u^{d/2-1} e^{\beta u})$ is $\frac{d/2-1}{u} + \beta \leq \kappa$, so this logarithm lies at most $\eta\kappa \leq 1$ below its value at u^*, while $e^{\lambda(u-1)} \leq \beta/\lambda$ there. Hence

$$\int_0^{u^*} g \;\geq\; \eta\, (u^*)^{d/2-1} e^{\beta u^*} e^{-1-\beta/\lambda}.$$

Combining. The factors $(u^*)^{d/2-1}$, $e^{\beta u^*}$ and $e^{-\beta/\lambda}$ cancel, so

$$P(U > u^*) \;\leq\; \frac{\int_{u^*}^{\infty} g}{\int_0^{u^*} g} \;\leq\; \frac{e}{\eta} \sqrt{\frac{2\pi}{\beta\lambda}}\, \exp\Big(\frac{(d/2-1)^2}{2u^{*2}\beta\lambda}\Big).$$

With $\beta = 1$ and $u^* \geq 0.99$ we get $1/\eta = \max\{\kappa, 2/u^*\} \leq 1.02\, d$ for all $d \geq 2$, and $\lambda \geq 200 d^2$ makes the exponential factor at most 1.001. Hence $P(U > u^*) \leq 6.96\, d/\sqrt{\lambda}$, which is at most $\epsilon/2$ as soon as $\lambda \geq 194\, d^2/\epsilon^2$; equation 12 satisfies this. $\square$

Proof of Proposition 1, Item 2. Given $d \geq 2$ and $\epsilon \in (0, 1)$, choose β, λ as in equation 12 and let p_X, ϕ and $M = \{\|x\| \leq R\}$ be as above. By Lemma 2, p_X is smooth and strictly positive, ϕ is a C^∞ diffeomorphism of $\mathbb{R}^d$ with $\phi_\# p_X = \mathcal{N}(0, I_d)$, and ϕ is density anti-aligned with center 0 on M. By Lemma 3, $P_X(M) \geq 1 - \epsilon/2 \geq 1 - \epsilon$. $\square$

Since ties occur with probability zero, it follows in particular that for independent $X, Y \sim p_X$ the ranking by latent norm is inverted relative to the ranking by input density with probability at least $P_X(M)^2 \geq (1 - \epsilon/2)^2 \geq 1 - \epsilon$:

$$\Pr\Big[\big(p_X(X) - p_X(Y)\big)\big(\|\phi(X)\| - \|\phi(Y)\|\big) > 0\Big] \;\geq\; 1 - \epsilon.$$

REMARKS. The constant in equation 12 is sufficient, not tight. The loss occurs in the Taylor step of the outer bound, which discards the doubly exponential decay beyond u^*; numerically, much smaller values of λ already suffice. Table 4 reports $P_X(M)$, computed by numerical quadrature over the radial profile, for $\beta = 0.5$ and λ of the order of $100\,d$ to $1000\,d$. The construction also isolates exactly where the non-US flexibility enters: on M, the input density rises with $\|x\|$ precisely where the Gaussian base falls, and the monotone radial transport preserves this ordering, so the latent norm runs opposite to the density. Finally, ϕ is a legitimate diffeomorphism of $\mathbb{R}^d$, which is what Item 2 requires; we make no claim that a particular coupling architecture represents ϕ exactly.

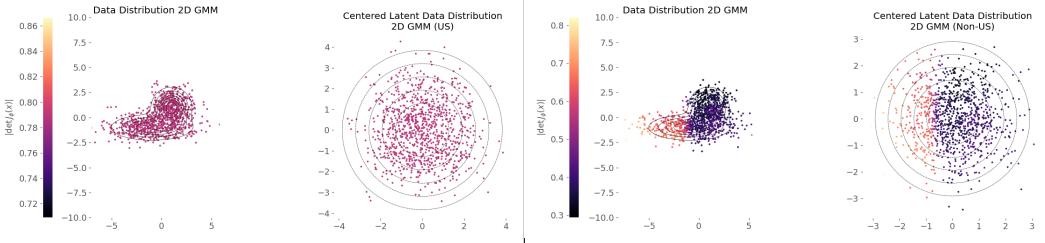
Table 4: Probability of the anti-aligned ball $M = \{\|x\| \leq R\}$ for the construction in equation 9–equation 11 with $\beta = 0.5$ (numerical quadrature).

d	λ	R	$P_X(M)$
2	500	0.9931	0.984
8	5000	0.9991	0.992
32	5000	0.9991	0.972
128	50000	0.99988	0.986

Gaussian Mixture Experiment The parameters of the asymmetric Gaussian mixture distribution for dimensionality $d = 2, 8, 32, 128$ are given as follows:

- $\mu_1 = \mathbf{1}_d, \mu_2 = -\mathbf{1}_d$
- $\Sigma_1 = I, \Sigma_2 = \text{diag}(\theta_d)$,

where $\mathbf{1}_d$ denotes the all ones vector in d dimensions and $\theta_d \in \mathbb{R}^d$ is chosen such that half of the entries are 5.0 and half are $1/2$. This asymmetric choice of covariance matrices is deliberate in order to incentivize inhomogeneous volume transfer via $\det \Sigma_1 = 1$ and $\det \Sigma_2 = 2.5^{\frac{d}{2}}$. For each model architecture and dimension, we conduct a hyperparameter optimization with 10 runs. For the flow architectures, we use (Non)USFlows with 2-10 coupling blocks and 2-3 layers per conditioner. For Deep SVDD, we use MLP encoders with depth 2–6 following a decreasing schedule based on the input dimension d (i.e., $[d, d]$, $[d, 2d, d]$, ... ,$[d, 16d, 8d, 4d, 2d, d]$), forgoing bias terms and bounded activations as discussed in the preliminaries. We select the best-performing model per setting by validation loss (one-class for DeepSVDD, NLL for flows). Additionally, we optimize non-architecture specific parameters such as learning rate and batch size. The entire experiment configuration is provided with the source code as YAML file.

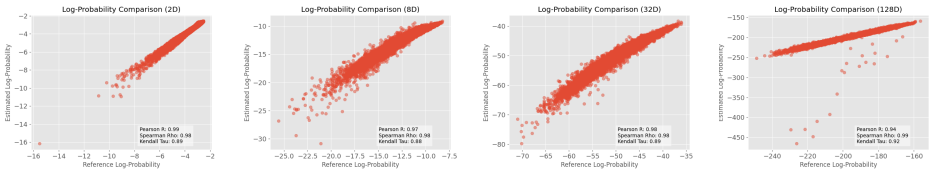


(a) Data distribution, d=2 (b) USF (c) Data distribution, d=2 (d) non-USF

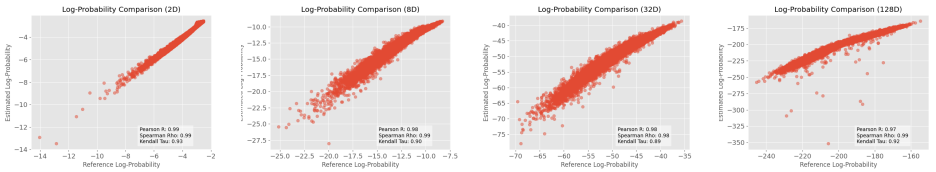
Figure 5: **Visualization of the volume changes in different areas of the data distribution.** (a,c): Sample and contours of the true data distribution. The determinant of the respective transformation Jacobian is color coded. (b,d): Sample and ideal contours of centered data latents of the (b) USF and the (d) non-USF. The dashed contours show the contour lines of the flows base distribution (centered, $\sigma - 3\sigma$). The color of each sample encodes the absolute value of the determinant of the transformation Jacobian. The USF enforces a constant Jacobian determinant, ensuring uniform scaling, whereas the non-USF adapts local densities through variable volume changes to correct alignment discrepancies.

Figure 5 visualizes the volume transfer for the NonUSFlows model and the USFlows model. Figure 6 plots the estimated likelihoods against the true data likelihoods. We see that USFlows and NonUSFlows are likewise able to model the data likelihoods relatively accurately, despite differences in

latent space alignment. Figure 7 plots the true likelihoods against the latent norms for DeepSVDD, USFlow and NonUSFlow. As one can see, only USFs consistently maintain a monotonic relationship between data likelihoods and latent norms, which is witnessed by the Spearman ρ and the Kendall τ. Figure 8 visualizes the latent alignment for the 2D case. We encountered some numerical instabilities when working with NonUSFlows in 128 dimensions. See Section E for details.

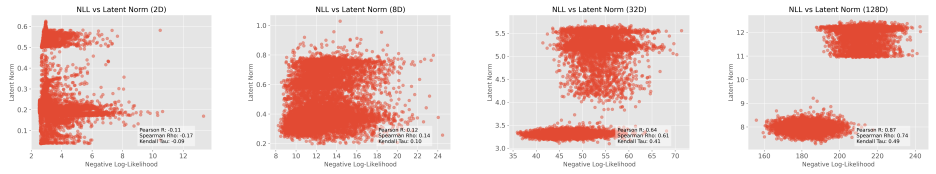


(a) non-USF, d=2 (b) non-USF, d=8 (c) non-USF, d=32 (d) non-USF, d=128

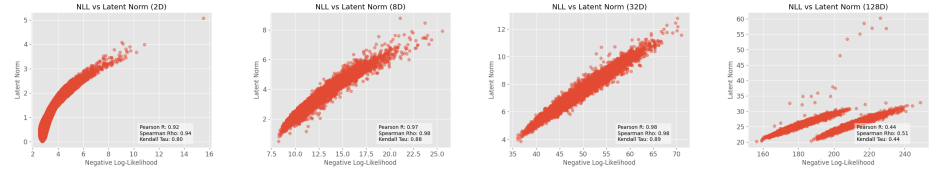


(e) USF, d=2 (f) USF, d=8 (g) USF, d=32 (h) USF, d=128

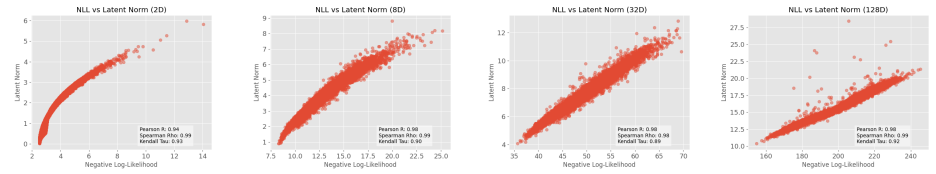
Figure 6: **True log-likelihoods against the estimated log-likelihoods for the GM experiments.** (a) – (d): Non-uniformly scaling flow (e) – (h): Uniformly scaling flow. While we see clear differences in latent space alignment, especially for 128 dimensions, the quality of the estimated likelihoods remains roughly comparable among the flow models.



(a) Deep SVDD, d=2 (b) Deep SVDD, d=8 (c) Deep SVDD, d=32 (d) Deep SVDD, d=128

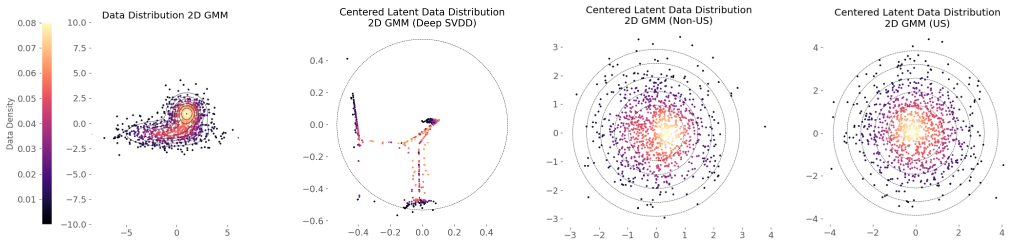


(e) non-USF, d=2 (f) non-USF, d=8 (g) non-USF, d=32 (h) non-USF, d=128



(i) USF, d=2 (j) USF, d=8 (k) USF, d=32 (l) USF, d=128

Figure 7: **True log-likelihoods against the latent norms for the GM experiments.** (a) – (d): Deep SVDD (e) – (h): Non-uniformly scaling flow (i) – (l) Uniformly scaling flow. Uniformly scaling flows maintain a monotonic relationship between latent norm and data density, while for non-uniformly scaling flows remain some monotonic relationship but it can be broken in unpredictable ways due to the influence of the Jacobian determinant in the change of variables formula (see Figure 8 and Figure 5 for a 2D visualization). For Deep SVDD, the alignment initially appears random and splits into homogeneous clusters with increasing dimensionality.



(a) Data distribution, d=2 (b) Deep SVDD (c) non-USF (d) USF

Figure 8: **Visualization of the Gaussian mixture experiment for 2D.** (a): The true data distribution, and the latent spaces of (b) Deep SVDD, (c) non-USF and (d) USF. Point color encodes the true data density. Dashed lines show contours of the base distribution (σ–3σ) (flows) or the decision threshold (Deep SVDD) based on the 99th distance percentile. The USF shows direct density alignment between data and latent spaces, while Deep SVDD and non-USF show noticeable discrepancies.

C Details for Tabular AD Experiments (Section 5.1)

We integrate the officially provided implementations of all competitor methods into a common evaluation pipeline and reproduce all results ourselves under a protocol that follows the corresponding original papers and implementations as closely as possible. Unless stated otherwise, hyperparameters are fixed across datasets and random seeds. For mini-batch training, we use the dataset-dependent batch-size schedule reported by URD Wang et al. (2025), varying between 64, 256 and 512, depending on sample size, which was also used for the shared tabular flow comparisons. The only intentional architectural deviation from the original baselines is that Deep SVDD and DASVDD use the same backbone dimensionalities as DOHSC, so that the deep one-class baselines and hybrids differ primarily in their objective rather than in encoder capacity. We also share the results of our normalization and learnable radial base studies, as well as the detailed main results tables.

C.1 Method Parameterization

USF and non-USF USF and non-USF use the same coupling-flow backbone with 8 coupling blocks, alternating masks, conditioners with 5 hidden layers of width 256, and feature mixing based on Householder reflections (or bijective affine LU-parameterization transforms in the ablation). Each coupling conditioner is an MLP with 5 linear layers in total: four hidden layers of width 256 followed by a final output projection. LeakyReLU activations with negative slope 0.01 are used after all non-final linear layers. The USF uses additive coupling, whereas the non-USF uses affine coupling with log-scale clamp 2.0; accordingly, only the final conditioner projection differs in output dimension, producing a shift for USF and shift–scale parameters for non-USF. Both models use a factorized standard Gaussian base distribution $P_B = \mathcal{N}(c, I_d)$. This corresponds to an isotropic radial monotone base distribution with learnable center c. Since this base is isotropic and strictly radially decreasing in $\|z\|$, it satisfies the radial monotonicity assumption used in Section 4.3. For optimization, both use Adam with learning rate 10^{-3}, at most 200 epochs, and early stopping after 5 epochs without improvement larger than 10^{-4}. For determinant regularization we use the bilateral log-normal prior from Section 4.2 with $\sigma = 1.0$. Hence, when using Householder reflections, this still regularizes the final LU-transform scaling. The scoring corresponds to the flows likelihood prediction, which in the case of the USF, coincides with latent distance to the base distributions center.

OneFlow and OneFlow-Gen OneFlow and OneFlow-Gen are trained using the minimal-volume one-class flow objective Maziarka et al. (2022). We also use 8 coupling flow layers with 5 conditioner transforms per layer, each of width 256, and objective parameter $\alpha = 0.05$. The coupling conditioners are MLPs with LeakyReLU activations of negative slope 0.1 after all non-final hidden layers. OneFlow denotes the constant-Jacobian variant and uses a single additive conditioner, whereas OneFlow-Gen denotes the corresponding non-constant-Jacobian variant and uses separate additive and multiplicative conditioner branches, with the multiplicative branch ending in a $\tanh$ activation. Both models are trained with Adam at learning rate 10^{-3} for at most 200 epochs, using the same early-stopping rule

as the flow models where applicable. The anomaly score is the method's original minimal-volume flow score.

Deep SVDD and DOHSC. Deep SVDD and DOHSC are evaluated on standardized inputs and use the same MLP encoder backbone. For an input dimension d, the encoder consists of three bias-free linear layers $d \to 500 \to 32 \to 32$ with a LeakyReLU activation after the first layer. For DOHSC, the final representation is additionally passed through the orthogonal projection layer of Zhang et al. (2024), with input and output dimension 32. Autoencoder pretraining uses the same encoder together with a bias-free linear decoder $32 \to d$ and is performed for 10 epochs with Adam, learning rate 10^{-3}, and weight decay 10^{-6}. After pretraining, the one-class center is initialized from the pretrained encoder representations and kept fixed during one-class optimization. Deep SVDD is then trained with the standard center-distance objective, while DOHSC adds the orthogonal hypersphere compression mechanism with $\nu = 0.1$, and $\alpha = 0.05$. Both methods are trained for at most 200 epochs with Adam, learning rate 10^{-4}, and weight decay 10^{-6}.

DASVDD DASVDD is evaluated on standardized inputs using the official objective from Hojjati & Armanfard (2024), which combines autoencoding reconstruction with a one-class compactness criterion. We use hidden dimension 500, latent dimension 32, just as for Deep SVDD and DOHSC. As a methodological distinction, however, DASVDD applies LeakyReLU activations after every encoder layer (with bias terms permitted) and is trained for at most 200 epochs with Adam at learning rate 10^{-3}. The method-specific center update uses center learning rate 1.0 with decay 0.01, and the support-parameter tuning stage is run for 10 epochs.

URD URD is evaluated on standardized inputs using the official implementation and tabular protocol of Wang et al. (2025). We use latent dimension 32, objective weight $\lambda = 0.5$, contraction weight $\alpha = 0.01$, Adam learning rate 10^{-5}, weight decay 10^{-7}, and 150 epochs (as the authors propose using either 100 or 150 epochs for their method on the given datasets). All remaining architectural choices follow the original URD implementation.

C.2 Detailed Result Tables

Table 5: **AUC-ROC mean and stdev.(%) over three seeds on 20 tabular benchmark datasets. Uncontaminated training data.** Highest mean per row in bold, second-highest underlined.

Dataset	Deep One-Class		Hybrid				Normalizing Flow	
	DeepSVDD	DOHSC	DASVDD	OneFlow	OneFlow-G	URD	non-USF	USF
Arrhythmia	79.13±0.5	74.99±2.4	<u>80.25</u>±0.3	79.96±1.4	**80.89**±1.7	77.55±1.9	69.02±9.9	78.91±1.7
Breastw	**98.49**±0.7	96.00±2.5	97.96±0.9	97.91±0.8	<u>98.37</u>±0.5	98.37±0.8	97.88±0.8	98.33±0.8
Cardio	**96.00**±2.6	86.04±7.8	84.92±3.9	90.28±0.5	89.69±0.7	<u>93.88</u>±0.4	92.15±0.9	93.67±3.0
Census	67.72±1.0	48.33±4.8	**72.84**±0.6	69.66±0.6	63.46±8.7	37.86±10.8	69.95±0.8	<u>71.12</u>±0.4
Campaign	70.49±2.0	65.60±5.9	**78.48**±1.5	70.04±1.1	69.99±2.7	69.31±1.0	<u>76.10</u>±0.4	72.85±0.7
Cardiot.	76.45±12.4	**80.45**±11.2	65.12±1.7	<u>79.61</u>±3.4	75.67±11.8	70.75±13.4	73.96±1.2	76.57±5.7
Fraud	94.39±0.7	83.44±6.0	95.40±0.5	<u>95.71</u>±0.5	94.02±0.9	87.76±3.2	94.85±1.6	**95.90**±0.1
Glass	66.52±3.7	67.13±3.1	71.38±5.8	63.57±10.4	37.00±14.2	55.59±7.5	<u>75.73</u>±2.8	**76.56**±0.2
Ionosphere	88.68±3.0	89.84±13.0	**97.70**±0.6	<u>97.44</u>±0.6	97.17±0.1	95.56±2.2	91.52±2.2	95.94±0.2
Mammo.	<u>88.19</u>±0.9	79.63±6.3	86.68±3.2	84.06±1.1	77.81±5.9	87.46±3.1	86.48±1.7	**88.21**±0.5
NSL-KDD	52.98±6.4	63.13±4.5	87.60±5.4	<u>91.30</u>±3.2	87.80±8.5	61.15±5.0	**93.30**±2.3	83.76±5.1
Optdigits	65.24±21.2	58.84±7.2	<u>97.36</u>±0.3	**98.14**±0.6	97.19±1.9	58.21±19.7	83.05±20.0	94.46±1.9
Pima	66.76±0.9	60.53±7.9	<u>70.12</u>±0.9	66.26±1.2	64.65±2.3	68.10±1.8	70.06±5.8	**72.99**±1.3
Pendigits	88.04±11.8	61.85±18.5	97.49±0.6	<u>99.78</u>±0.2	**99.85**±0.1	42.05±45.3	98.97±0.8	98.96±0.4
Satellite	66.06±1.3	78.58±5.7	<u>81.05</u>±1.9	80.68±0.8	78.99±1.0	**82.53**±4.4	73.61±10.4	79.68±0.9
Satimage-2	96.81±0.6	91.66±8.3	<u>99.84</u>±0.1	99.84±0.0	99.68±0.3	99.01±0.2	99.68±0.4	**99.89**±0.0
Shuttle	99.28±0.2	99.59±0.1	**99.95**±0.0	99.79±0.1	<u>99.84</u>±0.1	99.59±0.3	98.75±0.3	99.46±0.1
Thyroid	<u>98.35</u>±0.5	92.77±4.7	97.62±0.9	**98.38**±0.2	96.13±2.5	98.10±0.5	97.93±0.3	97.97±0.4
Wbc	92.41±3.8	86.00±13.9	<u>95.50</u>±0.4	**95.81**±0.7	94.24±3.0	93.85±3.4	93.49±1.5	95.43±1.8
Wine	89.17±4.1	89.89±15.5	96.94±0.5	**99.78**±0.3	<u>97.11</u>±3.0	88.22±8.4	85.39±13.8	94.89±4.5
Average	82.06±3.9	77.72±7.5	87.71±1.5	<u>87.90</u>±1.4	84.98±3.5	78.24±6.7	86.09±3.9	**88.28**±1.5
Mean Rank	5.05	6.55	3.25	<u>3.20</u>	4.45	5.45	4.90	**3.15**

Table 6: **AUC-ROC mean and stdev.(%) over three seeds on 20 tabular benchmark datasets. Contaminated training data.** Highest mean per row in bold, second-highest underlined.

Dataset	Deep One-Class		Hybrid				Normalizing Flow	
	DeepSVDD	DOHSC	DASVDD	OneFlow	OneFlow-G	URD	non-USF	USF
Arrhythmia	74.32±9.4	66.65±5.8	**79.02**±3.3	77.74±2.4	78.21±3.8	74.84±3.4	69.71±12.3	77.44±2.6
Breastw	91.85±4.8	63.82±1.7	96.25±0.7	95.06±1.1	92.84±2.6	96.43±0.2	97.44±0.8	**97.84**±0.8
Cardio	**91.35**±8.9	54.58±5.9	72.44±1.4	67.33±1.8	70.70±4.9	79.66±4.6	84.04±2.4	81.75±0.3
Census	63.22±3.8	52.83±1.4	66.83±1.1	64.31±1.0	63.36±0.9	29.50±1.8	64.99±1.4	**67.24**±0.2
Campaign	68.10±3.1	54.44±3.5	**74.09**±0.2	64.34±3.7	61.83±2.9	64.43±1.8	73.83±1.3	71.89±0.8
Cardiot.	**68.14**±16.9	51.22±1.6	50.75±3.2	63.01±7.7	63.40±15.2	54.46±14.7	57.04±6.7	59.49±3.5
Fraud	94.12±0.9	61.82±13.8	94.71±0.4	95.11±1.3	92.60±2.5	82.74±11.7	94.42±1.8	**95.18**±0.6
Glass	72.33±4.6	57.28±28.3	**74.27**±5.5	46.44±12.3	48.30±38.0	64.08±6.5	68.53±18.5	73.22±7.0
Ionosphere	74.22±8.1	70.36±11.3	**91.96**±1.0	91.38±1.0	91.20±2.6	89.15±4.9	83.03±3.0	91.76±0.9
Mammo.	78.69±9.7	51.60±19.0	83.28±5.6	77.14±10.1	77.44±7.3	**84.81**±2.8	84.37±3.0	84.09±4.4
NSL-KDD	13.59±6.0	**55.00**±5.8	21.46±1.9	53.18±24.1	46.96±31.2	42.01±18.2	19.55±2.9	28.08±2.3
Optdigits	**59.98**±23.8	55.31±5.4	44.10±3.8	49.08±9.3	46.94±1.0	45.75±3.1	56.14±3.0	50.66±9.3
Pima	57.80±7.1	51.99±4.9	67.43±3.5	68.08±2.9	61.52±7.0	64.93±2.5	69.09±6.0	**69.91**±2.4
Pendigits	**85.44**±11.8	57.72±6.2	75.33±9.0	69.99±3.9	65.15±11.1	38.28±8.0	76.38±7.3	74.31±8.2
Satellite	61.00±0.6	54.09±2.9	**67.55**±4.0	63.84±2.2	58.37±3.6	66.06±4.1	55.93±9.6	61.96±4.8
Satimage-2	97.72±0.8	80.34±6.3	95.10±1.6	96.96±1.6	96.64±1.2	90.83±5.3	94.52±8.0	**98.77**±0.4
Shuttle	99.01±0.0	41.80±10.9	**99.23**±0.4	85.09±4.9	78.36±6.3	70.55±49.1	91.00±13.8	94.86±6.1
Thyroid	**96.75**±0.5	58.04±9.6	94.91±1.8	92.15±2.0	93.29±2.7	96.39±1.2	89.22±5.6	90.89±3.0
Wbc	88.36±3.9	57.00±23.6	88.86±2.2	87.90±2.6	74.38±8.2	86.72±4.6	91.15±2.2	**91.21**±1.9
Wine	68.56±13.5	70.44±24.2	65.67±25.4	**89.67**±4.4	87.56±13.0	62.22±23.9	72.67±17.1	53.22±23.7
Average	75.23±6.9	58.32±9.6	75.16±3.8	74.89±5.0	72.45±8.3	69.19±8.6	74.65±6.3	**75.69**±4.2
Mean Rank	4.05	6.95	3.50	4.25	5.05	5.20	3.95	**3.05**

C.3 Learnable Radial Bases Improve The Density Fit

In the following, we investigate how increased base distribution flexibility affects the density fit of USFs by evaluating held-out nominal likelihood alongside anomaly-detection performance. More specifically, we compare the Gaussian base used in our main experiments (USF-B) with a learnable radial base (USF-L) having density $p_B(x) = p_{\|B\|}(\|x\|)/V'(\|x\|)$, where the one-dimensional distribution of the sample norms $p_{\|B\|}$ is modeled via a mixture of 20 Gamma distributions and $V(r) \propto r^d$ is the volume of a ball of radius r in $\mathbb{R}^d$, with d the data dimensionality. We use the same 20 tabular datasets in the uncontaminated setting, with three seeds and matched data splits across methods within each seed. We also compare with Gaussian-base non-USFs using log-scale clamps 1.0 and 2.0. Accordingly, the Gaussian-base results are the same as in the main experiments.

Table 7 aggregates these comparisons. Relative to USF-B, the median dataset-level NLL reduction is 1.25 per dimension, with interquartile interval $[0.47, 1.81]$. The macro-mean AUC-ROC change is 0.84 percentage points. USF-L also attains lower mean nominal NLL than the non-USF baselines on 17/20 datasets at clamp 1.0 and 19/20 at clamp 2.0. These results demonstrate improved held-out density fit, while showing that this improvement does not yield a uniform gain in anomaly-detection performance.

Table 7: Aggregate paired comparisons over three seeds. Each row compares candidate C with reference R. AUC effects are $A_C - A_R$ in percentage points, while nominal NLL-per-dimension reductions are $N_R - N_C$, so positive effects favor C. The AUC column reports the equally weighted macro mean of the 20 dataset-level mean paired differences. The NLL column reports the median and first and third quartiles $[Q_1, Q_3]$ of the 20 dataset-level mean paired reductions. The final columns count datasets based on these dataset-level means.

C	R	Δ AUC-ROC $(C-R)$ ↑ Macro mean	Δ NLL $(R-C)$ ↑ Median $[Q_1, Q_3]$	Datasets with better mean NLL	Better mean NLL but no better mean AUC
USF-B	non-USF-B (1.0)	0.81	$-0.36\,[-1.16,\ -0.03]$	5/20	0/20
USF-B	non-USF-B (2.0)	2.18	$0.08\,[-0.61,\ 2.84]$	10/20	1/20
USF-L	USF-B	0.84	$1.25\,[0.47,\ 1.81]$	20/20	6/20
USF-L	non-USF-B (1.0)	1.64	$0.70\,[0.21,\ 1.40]$	17/20	2/20
USF-L	non-USF-B (2.0)	3.02	$1.73\,[0.60,\ 7.23]$	19/20	1/20

Table 8 reports per-dataset paired effects and the number of seeds with a strictly positive effect. AUC-ROC differences are defined as candidate minus reference, while nominal NLL-per-dimension reductions are reference minus candidate; positive values therefore favor USF-L in both cases. USF-L reduces mean held-out nominal NLL relative to USF-B on all 20 datasets, including all three seeds on 17 datasets. While the increased expressivity seems to help on certain datasets like campaign or nslkdd, six datasets nevertheless show a decrease in mean AUC-ROC.

Table 8: Per-dataset paired effects over three seeds. Each entry reports the mean seed-level effect followed by the number of strictly positive effects. Throughout, L denotes USF-L, while B is the baseline named by the column. AUC effects are $A_L - A_B$ in percentage points, and nominal NLL-per-dimension reductions are $N_B - N_L$. Hence, positive values always favor USF-L.

Dataset	USF-B Δ AUC-ROC Mean $(L - B)$ $\uparrow$; (wins)	USF-B Δ NLL Mean $(B - L)$ $\uparrow$; (wins)	non-USF-B (1.0) Δ NLL Mean $(B - L)$ $\uparrow$; (wins)	non-USF-B (2.0) Δ NLL Mean $(B - L)$ $\uparrow$; (wins)
arrhythmia	0.48; (2/3)	1057.34; (3/3)	17.88; (3/3)	11.05; (3/3)
breastw	0.18; (2/3)	1.37; (3/3)	0.63; (3/3)	6.12; (3/3)
campaign	5.06; (3/3)	1.89; (3/3)	0.77; (3/3)	0.84; (3/3)
cardio	−1.06; (1/3)	1.88; (3/3)	6.08; (3/3)	3.78×10^5; (3/3)
cardiotocography	−3.68; (0/3)	0.39; (3/3)	0.34; (3/3)	0.04; (2/3)
census	0.21; (2/3)	1.78; (3/3)	−0.29; (1/3)	0.62; (2/3)
fraud	0.42; (3/3)	1.26; (3/3)	−0.02; (2/3)	2.57; (2/3)
glass	0.22; (2/3)	1.01; (2/3)	0.61; (3/3)	0.54; (2/3)
ionosphere	0.64; (3/3)	4.54; (3/3)	1.34; (3/3)	2.29; (3/3)
mammography	−2.12; (2/3)	0.79; (3/3)	0.18; (2/3)	−0.32; (0/3)
nslkdd	11.82; (3/3)	3.01; (3/3)	0.80; (3/3)	637.90; (3/3)
optdigits	3.79; (3/3)	1.23; (3/3)	0.22; (3/3)	1.03; (3/3)
pendigits	0.71; (3/3)	0.35; (3/3)	0.29; (1/3)	0.16; (1/3)
pima	0.63; (2/3)	0.49; (2/3)	0.17; (3/3)	1.73; (3/3)
satellite	0.33; (2/3)	0.40; (3/3)	1.69; (3/3)	2.61; (3/3)
satimage-2	−0.04; (1/3)	1.34; (3/3)	1.71; (3/3)	1.74; (3/3)
shuttle	−0.45; (0/3)	0.74; (3/3)	0.78; (2/3)	10.55; (3/3)
thyroid	0.27; (2/3)	0.27; (3/3)	−0.02; (2/3)	1071.61; (1/3)
wbc	−1.23; (1/3)	1.37; (3/3)	1.32; (3/3)	1.08; (3/3)
wine	0.56; (2/3)	0.20; (2/3)	1.58; (3/3)	0.55; (3/3)

C.4 Impact of Normalization on the Tabular Experiments

Here we examine whether spectral normalization Miyato et al. (2018) or ActNorm Kingma & Dhariwal (2018) close the observed performance gap between USFs and non-USFs. We evaluate three configurations on the same 20 tabular datasets as in the uncontaminated setting of the main experiments, using the same seeds and seed-dependent stratified split for all methods. Non-USFs are reported with affine-coupling log-scale clamps of 2.0 and 1.0. Table 9 compares all variants with the Base versions from the main experiments

Table 9: Normalization study over 20 datasets and three seeds. AUC-ROC means are percentages; gaps and standard deviations are in percentage points. Mean $\pm$ SD reports the average of the dataset-level means and the average of their across-seed sample SDs. Δ is the paired macro-mean of Base USF minus the row; positive gaps favor Base USF. Max SD is the largest within-dataset across-seed sample SD. SD>5 counts datasets with SD above five points; AUC<50 counts runs below 50% AUC-ROC out of 60 runs. Wins counts datasets whose three-seed mean exceeds Base USF, out of 20.

Setup	Method	Mean $\pm$ SD	Δ	Max SD	SD>5	AUC<50	Wins
Base	USF	88.28 ± 1.49	—	5.74	2	0	—
	non-USF (1.0)	87.47 ± 3.30	0.81	18.09	4	0	4
	non-USF (2.0)	86.09 ± 3.89	2.18	20.00	5	0	3
SpecNorm	USF	86.83 ± 1.80	1.44	7.86	1	0	4
	non-USF (1.0)	81.18 ± 7.71	7.10	23.28	10	2	5
	non-USF (2.0)	71.34 ± 14.12	16.94	46.49	11	10	1
ActNorm	USF	87.71 ± 2.78	0.57	12.74	3	0	7
	non-USF (1.0)	85.97 ± 5.33	2.30	35.12	4	2	5
	non-USF (2.0)	85.13 ± 4.92	3.14	21.74	6	2	6

ActNorm reduces the macro-mean performance of both non-USF variants relative to their Base versions. At clamp 1.0, the gap to Base USF is still 2.30 percentage points, although the ActNorm non-USF has a higher mean on 5 of the 20 datasets. At clamp 2.0, the gap is 3.14 points, with a higher mean on 6 datasets. Thus, Base USF retains the highest macro mean among the evaluated normalization configurations. Spectral normalization also reduces mean performance for both non-USF variants. Its clamp-2.0 variant has 10 of 60 runs below 50% AUC-ROC, compared with none for each USF configuration. Overall, these experiments show that the tested normalization methods do not eliminate Base USF's macro-mean performance advantage.

D Details for Image AD Experiments (Section 5.2)

Here we report the full image- and pixel-level AUC-ROC tables for MVTec AD and VisA discussed in Section 5.2, as well as the image-level counterpart of the MVTec AD ablation.

D.1 Method Parameterization

CFlow Feature maps are drawn from a Wide ResNet-50-2 backbones layers 2–4. The conditional normalizing flow consists of eight coupling blocks, each conditioned on a learned 128-dimensional positional embedding. To prevent numerical instability, the coupling network's log-scale outputs are clamped to ± 1.9 before exponentiation. The learning rate is 1×10^{-4} without weight decay.

FastFlow A ResNet-18 backbone feeds into eight alternating 1×1 and 3×3 convolutional coupling layers. The internal scale-and-shift subnetworks maintain the same channel width as their inputs, balancing expressivity with efficiency. Training uses a learning rate of 1×10^{-3} and weight decay 1×10^{-5}.

U-Flow A MS-CaiT backbone produces a U-shaped, multi-scale feature pyramid. Four 2D affine coupling steps at each scale use full-width subnet channels to preserve detailed spatial information. Log-scale predictions are clamped to ± 2.0 for stability. The initial learning rate is 1×10^{-3} with weight decay 1×10^{-5} and a linear decay to 40% over $25{,}000$ iterations.

D.2 Detailed Result Tables

Table 10: **Mean pixel AUC-ROC and stdev. values (%) per VisA class averaged over 3 runs.** Highest mean value per row in bold, second-highest underlined. "#Best" shows how many times each column had the highest mean value.

	Base Flow			USF		
Class	FastFlow	U-Flow	CFlow	FastFlow	U-Flow	CFlow
candle	98.0±0.4	99.0±0.1	99.0±0.0	**99.1±0.0**	94.8±3.5	99.0±0.0
capsules	98.0±0.1	95.7±3.5	96.5±0.7	**99.2±0.0**	98.2±0.1	97.2±0.0
cashew	98.2±0.3	**99.4±0.1**	98.3±0.3	98.8±0.0	99.3±0.1	98.4±0.0
chewinggum	98.9±0.1	99.2±0.1	99.0±0.2	**99.4±0.0**	99.3±0.2	**99.4±0.0**
fryum	84.6±4.6	95.5±0.5	**97.1±0.8**	**97.1±0.0**	96.3±0.1	97.0±0.1
macaroni1	97.8±0.7	94.0±9.1	97.3±0.6	**99.7±0.0**	88.3±8.4	99.1±0.0
macaroni2	94.5±0.3	81.9±20.2	97.1±1.3	**99.1±0.0**	82.6±8.0	98.3±0.1
pcb1	99.1±0.1	95.1±7.1	99.3±0.1	99.2±0.0	**99.4±0.0**	99.3±0.0
pcb2	96.7±0.4	97.2±0.7	97.4±0.4	**98.3±0.1**	96.3±0.3	96.7±0.0
pcb3	96.6±0.2	97.2±0.9	97.0±0.4	**98.7±0.0**	96.3±1.4	97.3±0.0
pcb4	97.5±0.3	96.2±2.5	97.2±0.6	97.2±0.1	**98.5±0.0**	97.7±0.1
pipe_fryum	96.0±1.0	98.9±0.6	98.8±0.1	**99.2±0.0**	**99.2±0.0**	99.0±0.0
Average	96.3±0.7	95.8±3.8	97.8±0.5	98.8±0.0	95.7±1.8	98.2±0.0
#Best	0	1	1	9	3	1

Table 11: **Mean image AUC-ROC and stdev. values (%) per VisA class averaged over 3 runs.** Highest mean value per row in bold, second-highest underlined. "#Best" shows how many times each column had the highest mean value.

	BASE FLOW			USF		
Class	FastFlow	U-Flow	CFlow	FastFlow	U-Flow	CFlow
candle	**96.5±0.3**	92.7±1.5	91.2±2.8	84.9±1.1	84.1±2.0	94.6±0.4
capsules	73.9±1.9	74.2±6.5	77.6±14.8	75.5±0.4	**79.7±0.8**	70.8±5.4
cashew	88.4±2.2	93.7±1.5	**96.7±1.2**	90.6±0.7	94.7±0.5	95.9±0.6
chewinggum	98.2±0.3	98.5±0.5	**99.6±0.1**	99.4±0.1	98.6±0.3	**99.6±0.1**
fryum	89.5±5.0	88.5±1.5	52.2±18.6	**92.8±0.5**	86.0±0.2	84.9±0.5
macaroni1	87.8±2.9	79.8±22.2	71.4±10.7	88.8±0.5	77.6±0.3	**89.3±0.2**
macaroni2	67.8±2.5	63.5±13.6	78.6±18.2	**80.5±0.5**	74.8±1.2	71.6±1.4
pcb1	87.9±1.4	76.8±15.2	87.8±5.4	83.8±1.4	83.1±0.2	**92.7±0.4**
pcb2	**87.7±0.8**	80.8±7.1	82.9±4.7	85.7±0.9	76.6±0.4	86.3±0.8
pcb3	**86.9±1.3**	80.5±8.8	71.6±11.5	81.4±0.3	85.5±1.2	79.5±0.6
pcb4	96.2±0.2	90.4±1.8	96.1±1.2	92.5±0.6	88.7±0.5	**97.3±0.4**
pipe_fryum	94.9±1.7	95.3±0.6	84.8±9.9	96.7±0.1	97.9±0.1	**98.6±0.1**
Average	88.0±1.7	84.6±6.7	82.5±8.3	87.7±0.6	85.6±0.6	88.4±0.9
#Best	3	0	2	2	1	5

Table 12: **Mean pixel AUC-ROC and stdev. values (%) per MVTec AD class averaged over 3 runs.** Highest mean value per row in bold, second-highest underlined. "#Best" shows how many times each column had the highest mean value.

	BASE FLOW			USF		
Class	FastFlow	U-Flow	CFlow	FastFlow	U-Flow	CFlow
bottle	97.4±0.2	96.9±0.9	97.7±0.2	97.8±0.1	**98.7±0.1**	98.3±0.0
cable	94.9±0.2	95.6±2.5	94.9±1.4	93.8±1.1	**97.2±0.1**	95.2±0.1
capsule	97.7±0.6	**98.8±0.0**	98.6±0.2	**98.8±0.0**	98.1±0.0	**98.8±0.0**
carpet	98.6±0.1	99.3±0.1	98.8±0.1	98.7±0.0	**99.5±0.0**	98.8±0.0
grid	98.0±0.4	97.8±0.1	97.4±0.1	**98.3±0.1**	97.4±0.1	96.7±0.1
hazelnut	93.3±1.0	98.9±0.2	97.0±0.7	98.3±0.0	**99.4±0.0**	98.5±0.0
leather	**99.6±0.0**	99.3±0.1	99.4±0.1	98.9±0.2	99.5±0.1	99.1±0.0
metal_nut	96.0±0.3	80.2±28.8	92.3±5.1	97.3±0.1	**97.8±0.2**	97.5±0.1
pill	96.9±0.4	**98.2±0.9**	97.2±0.4	97.7±0.0	97.8±0.1	98.0±0.0
screw	84.5±1.9	96.5±1.1	97.5±0.4	**98.3±0.1**	93.9±2.9	96.8±0.1
tile	94.4±0.3	96.8±0.4	94.2±0.7	92.1±0.2	**97.0±0.3**	95.8±0.0
toothbrush	96.3±0.3	97.8±1.6	97.6±0.3	98.9±0.0	**99.0±0.1**	98.4±0.0
transistor	**96.4±0.4**	87.5±5.2	84.6±2.2	90.8±0.1	87.3±0.9	85.9±0.2
wood	95.3±0.3	**97.1±0.1**	87.4±2.3	94.2±0.2	96.7±0.4	94.3±0.0
zipper	**97.7±0.2**	97.5±0.2	96.7±1.4	**97.7±0.2**	95.4±0.4	97.0±0.0
Average	95.8±0.4	95.9±2.8	95.4±1.0	96.8±0.2	97.0±0.4	96.6±0.0
#Best	3	3	0	4	7	1

Table 13: **Mean image AUC-ROC and stdev. values (%) per MVTec AD class averaged over 3 runs.** Highest mean value per row in bold, second-highest underlined. "#Best" shows how many times each column had the highest mean value.

	BASE FLOW			USF		
Class	FastFlow	U-Flow	CFlow	FastFlow	U-Flow	CFlow
bottle	**100.0±0.0**	99.8±0.3	90.7±8.9	98.9±0.2	**100.0±0.0**	**100.0±0.0**
cable	91.8±0.8	85.8±7.1	85.7±6.8	71.0±2.3	**94.0±0.3**	86.2±1.1
capsule	88.9±1.7	**94.1±2.1**	83.5±7.9	84.7±1.2	80.3±0.9	87.2±0.6
carpet	98.0±0.6	**100.0±0.0**	78.4±8.4	96.8±0.2	99.9±0.0	95.9±1.3
grid	97.7±0.8	98.8±0.5	97.7±1.5	95.0±0.8	**99.3±0.9**	72.1±1.1
hazelnut	78.0±5.7	97.7±1.2	65.4±26.4	99.5±0.1	**100.0±0.0**	**100.0±0.0**
leather	**100.0±0.0**	**100.0±0.0**	96.4±5.1	99.7±0.1	**100.0±0.0**	**100.0±0.0**
metal_nut	97.1±1.4	82.7±29.9	32.6±42.6	95.5±0.2	**100.0±0.0**	99.3±0.2
pill	92.5±0.5	94.1±2.9	84.5±4.0	88.2±1.0	**96.3±0.4**	84.3±3.7
screw	67.7±2.4	71.0±4.1	83.8±10.7	**83.9±1.2**	71.5±0.8	61.9±4.7
tile	97.8±0.1	99.9±0.1	40.3±8.5	94.1±0.9	**100.0±0.0**	100.0±0.0
toothbrush	72.3±5.5	86.5±9.6	35.3±26.1	92.7±0.6	91.8±0.6	**96.9±1.5**
transistor	94.3±1.9	84.5±9.1	72.7±20.1	84.9±0.6	**95.7±1.1**	86.1±2.4
wood	98.1±0.3	**99.4±0.1**	53.6±7.4	98.2±0.2	96.9±3.2	99.1±0.1
zipper	95.9±0.1	**95.9±1.7**	86.2±7.3	81.9±3.2	92.0±1.1	92.8 ±0.2
Average	91.3±1.5	92.7±4.6	72.5±12.8	91.0±0.9	94.5±0.6	90.8±1.1
#Best	2	5	0	1	9	4

D.3 ABLATION STUDY (MVTEC AD, IMAGE-LEVEL)

Figure 9 reports the image-level AUC-ROC ablation, showing the same tendencies as in Section 5.2.

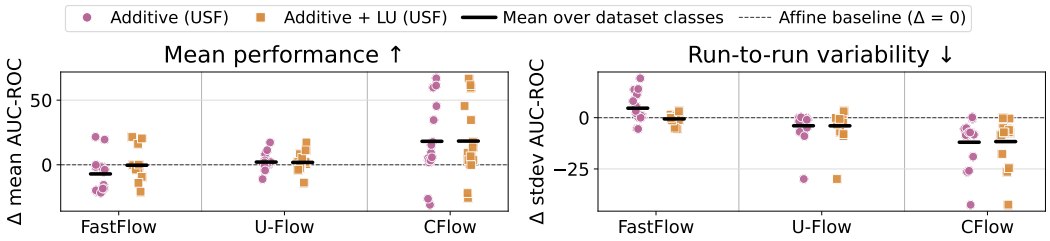


Figure 9: **Ablation on MVTec AD (Image).** Per-class changes in image AUC-ROC mean and stdev for replacing affine coupling with additive coupling, with / without LU-parameterized affine transforms.

E TRAINING INSTABILITIES OF NON-UNIFORMLY SCALING FLOWS (SECTION 4 AND 5)

During our experiments, we consistently observed numerical instabilities when training flows that employ affine coupling and LU-decomposed affine transforms on high dimensional data, which led to failed training runs. An investigation revealed that locally exploding determinants cause the problem. The issue becomes more pronounced with larger networks. While the problem is known when working with affine coupling, it seems to get amplified by the LU layers. We performed limited parameter tuning for critical parameters such as the affine clamping (which cuts the multiplicative part of the output of the affine coupling layers to a predefined range) but we could not eliminate the instabilities completely. We assume that this combination requires very careful, potentially problem specific, tuning of hyper parameters such as the prior scale, the affine clamping and parameter initialization, which we couldn't perform during our experiments.

In the following, we summarize instabilities that we encountered during our experiments. Note that we observed no such instabilities when working with the USFlows architecture nor when working

with affine coupling layers and affine transforms with unit absolute determinant (cf. Section 5), such as Householder transforms or permutations (originally implemented in Anomalib). It is also noteworthy that the tabular and image experiments do not share a common code base, since we implemented additive coupling and LU transforms in Anomalib independently.

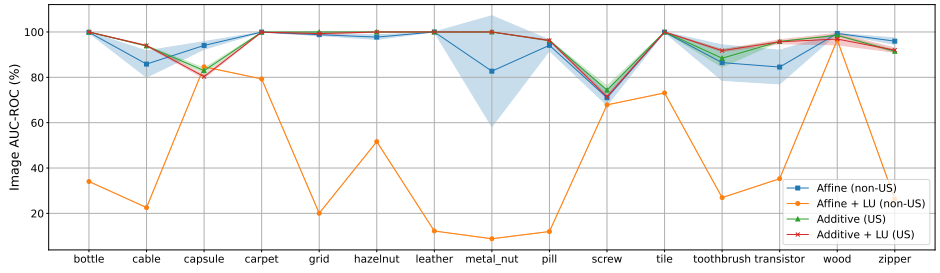
Gaussian Mixture Experiment During hyperparameter optimization, we encountered two failed runs for NonUSFlows architecture, both for models using 10 coupling blocks. Nevertheless, the resulting model still proved to be very competitive w.r.t. the likelihood-likelihood comparison, which is why we accepted the results of the experiments. We did not observe any problems for the other dimensions.

Tabular Anomaly Detection In the tabular experiments, affine-coupling non-USFs with LU-decomposed affine transforms (even with the prior on every affine transform, and though all but the last cancel block-wise) were sensitive to severe degeneracies, leading to numerical instabilities. In particular, removing the determinant prior led to a significant number of degenerate runs all seeds, making the results not directly comparable. The remaining full version of the analysis is given in Table 14. This observed sensitivity is consistent with the clamp ablation in Figure 3b.

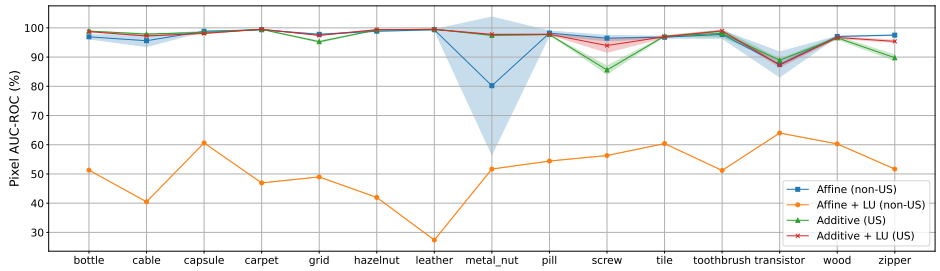
Table 14: Changes in avg. mean AUC-ROC relative to Table 1 in the uncontaminated setting.

Ablation			Method	
Transform	Prior	Aff. Conj.	non-USF	USF
HH reflec.	✗	✓	−8.80	−2.95
HH reflec.	✓	✗	−0.92	−0.17
LU affine	✓	✓	−26.61	−0.57
LU affine	✓	✗	−17.60	−6.36

Image Anomaly Detection We again encountered severe instabilities when using affine coupling and LU transforms in the ablation, in particular the different runs converged to the exact same (suboptimal) solution. Since we could not resolve the issue, we excluded the combination from the ablation. The obtained results on MVTec AD for the U-Flow architecture are reported in Figure 10.



(a) Image AUC-ROC



(b) Pixel AUC-ROC

Figure 10: MVTec AD per-class mean (a) image- and (b) pixel-level AUC-ROC (%) ± stdev. across 3 runs for the U-Flow architecture when using different flow layers.

F Execution Environment (Section 5)

Experiments were run on Ubuntu 22.04.5 LTS using Python 3.11 in a Conda environment. The machine was equipped with an Intel(R) Xeon(R) w9-3495X CPU, 500 GB of RAM, and two NVIDIA RTX PRO 6000 Blackwell GPUs, each with 96 GB of VRAM. Each experiment used a single GPU. Core software dependencies included PyTorch 2.10 with CUDA 12.8 and NumPy 1.26.4.

G Limitations

Our theoretical and empirical analysis is specific to anomaly detection in one-class settings. In particular, our claims concern density alignment of latent norms and the resulting anomaly-ranking behavior; they do not imply that uniformly scaling flows are universally preferable for general-purpose density estimation or generative modeling. Furthermore, since anomaly detection can be used in critical applications like security, monitoring, and industrial decision pipelines, false positives, false negatives, and misuse for surveillance can have practical harms; deployment should therefore still require domain validation, calibration, and human oversight.

A second limitation is the intended scope of the experimental study. The experiments are designed to test the mechanism predicted by the theory—i.e., the effect of uniform scaling on density-aligned scoring and robustness—and to compare against closely related methods under controlled settings. They are not a benchmark-maximization study with extensive per-dataset hyperparameter tuning aimed at leaderboard performance.

Our formal guarantees are obtained under an idealized optimization statement. Training on finite data and finite-capacity models can violate these conditions, so practical gains may vary with feature quality, architecture choices, and training regime. Likewise, the negative constructions in Items 1 and 2 of Proposition 1 describe adversarial/worst-case scenarios that are not intended to be realistic data-generating models. Their purpose is not to claim that such misalignments are likely in practice, but to show that the underlying optimization setup does not provide a general guarantee of density-aligned solutions. Their precise scope (the unavoidable limiting degeneracy in Item 1 and the necessarily probabilistic form of Item 2) is stated in Remark 1.

Moreover, while US constraints remove a structural source of input-dependent Jacobian compensation, they do not eliminate all training pathologies. Performance can still depend on architectural details (e.g., linear mixing layers and regularization strength), and non-US baselines may be sensitive to stabilization choices and initialization.

Finally, our empirical evaluation, though broad in tabular and image settings, is not exhaustive across modalities, contamination regimes, and computational budgets. Extending the analysis to such domains, alternative backbones, and stronger dataset specific tuning remains important future work.