Latent Diffusion Counterfactual Explanations

Simon Schrodi*[0009−0003−7006−953X], Karim Farid*[0000−0001−8054−0004], Max Argus[0000−0002−1288−7476], and Thomas Brox[0000−0002−6282−8861]

University of Freiburg
{schrodi,faridk,argusm,brox}@cs.uni-freiburg.de

Abstract. Counterfactual explanations have become an increasingly popular method for elucidating the behavior of opaque black-box models. Recently, several works leveraged pixel-space diffusion models for counterfactual generation. However, these approaches rely on training the generative models using the same or similar data as the model under investigation. This requirement restricts their applicability in situations where access to the data is limited. Further, they either required an auxiliary robust model, computationally intensive schemes, or limited the amount of change. To address above limitations, we introduce Latent Diffusion Counterfactual Explanations (LDCE), augmented with a novel consensus guidance mechanism. LDCE utilizes recent class- or text-conditional foundation diffusion models to allow for universal applicability. By running counterfactual generation in latent instead of pixel-space, we ensure that LDCE focuses on the important, semantic parts of the image. Lastly, our consensus guidance mechanism filters out the gradients of the model under investigation that are likely to result in semantically non-meaningful changes. We show the universal applicability of LDCE across a wide spectrum of models trained on diverse datasets. Finally, we demonstrate how LDCE can provide insights into model errors, enhancing our understanding of the behavior of black-box models.

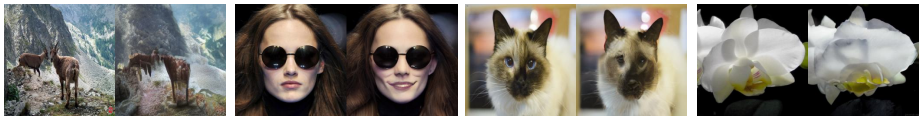
Keywords: Counterfactual explanations · Diffusion models · XAI

1 Introduction

Deep learning systems achieve remarkable results across diverse domains, yet their opacity presents a pressing challenge: as their usage soars in various applications, it becomes increasingly important to understand their underlying behavior and decision-making processes [2]. There are various paradigms that facilitate a better understanding of model behavior, including pixel attributions [69,5,68,47], feature visualizations [24,69,51], concept-based methods [7,39,41], inherently interpretable models [12,16,9], and counterfactual explanations [80,27].

In this work, we focus on counterfactual explanations that modify a (f)actual input with the *minimal semantically meaningful* change such that a model under investigation (*a.k.a.* target model) changes its output. Their goal is to provide

* Equal contribution.



(a) alp → coral reef
Probing ResNet-50
on ImageNet

(b) no-smile
→ smile
Probing
DenseNet-121 on
CelebA HQ

(c) birman
→ American pit bull
Probing OpenCLIP
on Oxford Pets

(d) moon orchid
→ rose
Probing DINO on
Oxford Flowers

Fig. 1. Our method, LDCE-txt, can be applied to *any* classifier, is *universally applicable*, and works across various learning paradigms. Each subfigure shows the original image (left) and a counterfactual explanation generated by LDCE-txt.

insights into the decision-making process of a black-box model rather than just generating the most visually appealing image edits. Given a (f)actual input x^F and target model f, Wachter *et al.* [80] proposed to find the counterfactual explanation x^{CF} closest to the (f)actual input defined by distance metric d that achieves a desired output y^{CF} defined by loss function $\mathcal{L}$, as follows:

$$x^{\text{CF}} \in \arg\min_{x'} \lambda_c \mathcal{L}(f(x'),\ y^{\text{CF}}) + \lambda_d d(x',\ x^{\text{F}}) \quad . \tag{1}$$

However, generating (visual) counterfactual explanations from Eq. (1) poses a challenge since, *e.g.*, relying solely on the gradient of a (non-robust) model leads to very minor instead of semantically meaningful modifications, akin to adversarial examples [3]. Thus, failing to provide human comprehensible insights into the decision-making processes. To overcome this, previous work resorted to adversarially robust models [64,11], restricted the set of image manipulations [27,82], used generative models [62,43,38,36], or a mixture of aforementioned approaches [3] to regularize towards the (semantic) data manifold. However, such requirements (or restrictions) limit the applicability of previous methods. For instance, data access can be restricted due to data privacy reasons making it difficult to train a suited adversarially robust or generative model.

To address aforementioned limitations, we present Latent Diffusion Counterfactual Explanations (LDCE). To the best of our knowledge, LDCE is the first counterfactual approach that is *universally applicable* to any classifier trained on some data domain; see Fig. 1. That is, LDCE has no component that requires specific training and can directly be applied to generate counterfactual explanations. Technically, we leverage recent class- or text-conditional foundational diffusion models to achieve universal applicability. To aid counterfactual generation, we run counterfactual generation in the latent space of an autoencoder [59] to focus on the semantic instead of pixel-level details. Lastly, we propose a simple *consensus guidance mechanism* that filters out the gradients of the model under investigation that are likely to result in semantically non-meaningful changes by using the gradients of the diffusion model's implicit classifier as reference. Code is provided at https://github.com/lmb-freiburg/ldce.

In summary, our key contributions are the following:

– By leveraging recent class- or text-conditional foundation diffusion models [59], we present the first approach that is *universally applicable* across models and datasets (restricted only by the domain coverage of the foundation model), as shown in Fig. 1.
– We present a simple yet effective *consensus guidance mechanism* that substantially diminishes confounding elements, such as an auxiliary robust classifier, while resulting in semantically meaningful modifications.
– Our method generates counterfactuals with semantically meaningful changes that can help to identify and fix model errors. Compared to previous methods, our method is superior w.r.t. realism and yields competitive empirical results even though previous methods use dataset-tailored components.

2 Background

2.1 Diffusion Models

Diffusion models are powerful generative models that can generate high-quality images [70,73,30,71,59]. The main idea is to gradually add small amounts of Gaussian noise to the data in the so-called forward diffusion process and gradually reverse it in a learned reverse diffusion process. Specifically, given scalar noise scales $\{\alpha_t\}_{t=1}^T$ and an initial, clean image x_0, the forward diffusion process generates intermediate noisy representations $\{x_t\}_{t=1}^T$, with T denoting the number of time steps. We can compute x_t by

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon_t, \quad \text{where} \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad . \tag{2}$$

The score estimator ϵ_θ, *i.e.*, parameterized denoising network (typically a modified U-Net [60]), is trained to undo the forward diffusion process for a pair $(x_t,\ t)$:

$$\nabla_x \log p_\theta(x) \propto \epsilon_\theta^{(t)}(x_t) \approx \hat{\epsilon}_t = (x_t - \sqrt{\alpha_t} x_0)/\sqrt{1-\alpha_t} \quad . \tag{3}$$

By rewriting Eq. (3), we can predict the clean data point

$$\hat{x}_0 \approx (x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t))/\sqrt{\alpha_t} \quad . \tag{4}$$

To gradually denoise, we sample the next less noisy representation x_{t-1} with a sampling method $S(x_t,\ \hat{\epsilon}_t,\ t) \to x_{t-1}$, such as the DDIM sampler [71]:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\Big(\frac{x_t - \sqrt{1-\alpha_t}\hat{\epsilon}_t}{\sqrt{\alpha_t}}\Big) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\hat{\epsilon}_t + \sigma_t\epsilon_t \quad . \tag{5}$$

Latent diffusion models. In contrast to GANs [26], VAEs [40,57], or normalizing flows [56], (pixel-space) diffusion models' intermediate representations are high-dimensional, rendering the generative process computationally intensive. To mitigate this, Rombach *et al.* [59] proposed to operate diffusion models in a perceptually equivalent, lower-dimensional latent space $\mathcal{Z}$ of a regularized autoencoder $\mathcal{A}(x) = \mathcal{D}(\mathcal{E}(x)) \approx x$ with encoder $\mathcal{E}$ and decoder $\mathcal{D}$ [25]. Besides speed-ups, it decouples semantic from perceptual compression *s.t.* the "focus [of the diffusion model is] on the important, semantic bits of the data" [59, p. 4].

Controlled image generation. The goal is to incorporate a condition c, *e.g.*, a class label or text, into the reverse diffusion process to control image synthesis. One approach is to condition the diffusion model on the gradients of a classifier $\nabla_{x_t} \log p_\phi(c|x)$, *a.k.a.* classifier guidance [70,74,22]:

$$\epsilon^{(t)}_{\theta,\eta}(x_t|c) = \epsilon^{(t)}_\theta(x_t) - \eta\sqrt{1-\alpha_t}\nabla_{x_t} \log p_\phi(c|x_t) \quad , \tag{6}$$

where the guidance scale η governs the influence of the conditioning signal. However, intermediate representations x_t have high noise levels that are likely to be out-of-distribution for a classifier. To overcome this, previous work used noise-aware classifiers [22], optimized intermediate representation of the diffusion process [37,81], or used one-step approximations [4,3,6]. In contrast to them, Ho & Salimans [31] trained a conditional diffusion model $\epsilon^{(t)}_\theta(x|c)$ with conditioning dropout and leveraged Bayes' rule, *i.e.*,

$$\epsilon^{(t)}_\theta(c|x_t) \approx \epsilon^{(t)}_\theta(x_t, c) - \epsilon^{(t)}_\theta(x_t) \quad , \tag{7}$$

to substitute the conditioning component $-\sqrt{1-\alpha_t}\nabla_{x_t} \log p(c|x_t)$ in Eq. (6):

$$\epsilon^{(t)}_{\theta,\eta}(x_t|c) = \epsilon^{(t)}_\theta(x) + \eta(\epsilon^{(t)}_\theta(x_t|c) - \epsilon^{(t)}_\theta(x_t)) \quad . \tag{8}$$

2.2 Visual Counterfactual Explanations

A counterfactual x^{CF} is a sample with the *smallest* and *semantically meaningful* change to an original factual input x^{F} in order to achieve a *desired output*, *cf*., Eq. (1). In contrast to adversarial attacks, counterfactual explanations aim for semantic (*i.e.*, human comprehensible) changes that help to understand the decision-making processes of black-box models. Initial works used gradient-based approaches [80,64,11] or restricted the set of image manipulations [27,1,82,78,79]. Other works leveraged invertible networks [34], deep image priors [76], or used generative models to regularize towards the image manifold [62,43,65,58,38,35].

Recent work also adopted (pixel-space) diffusion models due to their powerful generative capabilities [63,36,37,3]. Since intermediate representations of the diffusion process exhibit high levels of noise and may be out-of-distribution for standard classifiers, previous work adapted the classifier guidance approach (Eq. (6)) in various ways: by a shared encoder of the diffusion model and target model [63], albeit at the cost of model-agnosticity; a computationally intensive $\mathcal{O}(T^2)$ scheme to obtain $\hat{x}_0$ by running the unconditional reverse diffusion process at every step [36]; restricted number of modifications that limit the set of modifications [37]; or a gradient projection with an auxiliary adversarially robust model [3], albeit at the cost of a auxiliary model that needs to be trained on the same (or similar) data distribution as the target model.

Further remark on DVCE [3]. Besides the need for an auxiliary model from DVCE [3], we further found that counterfactuals are substantially confounded by the auxiliary model: Fig. 2 shows that counterfactuals of DVCE (2nd column)

(a) tench → eel (b) bulbul → goldfinch

Fig. 2. The adversarially robust model of DVCE [3] substantially influences the resulting counterfactuals. From left to right: original image, counterfactual images generated using DVCE, and DVCE using the robust classifier only, *i.e.*, without the target model. Further visual examples are provided in Fig. 8 in Appendix A.

look very similar to the ones when only using the auxiliary model (3rd column). Note that this can also be seen in Fig. 3 of Augustin *et al.* [3]. Thus, we cannot infer what "counterfactual features" the target model might have, as they are obfuscated by the ones from the auxiliary model.

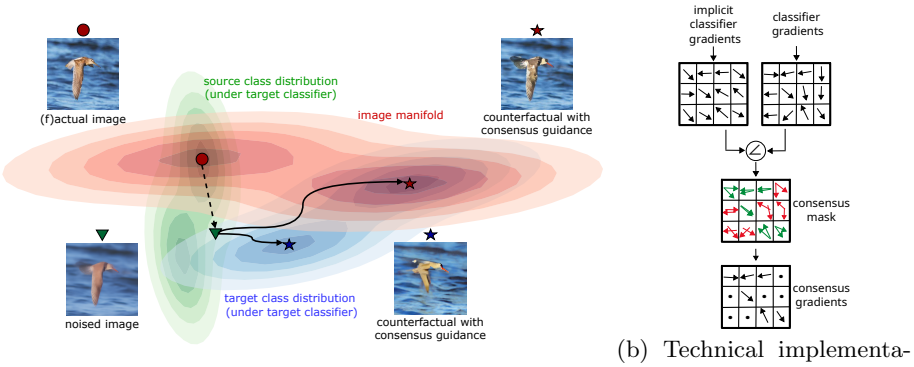
3 Latent Diffusion Counterfactual Explanations (LDCE)

In this section we present *Latent Diffusion Counterfactual Explanations (LDCE)*. LDCE does *not require an auxiliary model* and is *universally applicable*, *i.e.*, it can be used to analyze any target model trained on any data distribution. To this end, we harness the capabilities of recent class- or text-conditional foundation latent diffusion models (Sec. 3.1), augmented with a novel *consensus guidance mechanism* (Sec. 3.2). By operating the diffusion process in latent space, we ensure to focus on the important, semantic bits of the data, while as by-product speeding-up counterfactual generation [59]. Further, the foundational nature of text-conditional diffusion models grants LDCE the *versatility* to be applied across diverse models and datasets within reasonable bounds; as illustrated in Fig. 1. Besides that, our novel consensus guidance mechanism ensures semantically meaningful, counterfactual changes by utilizing the implicit classifier [31] of class- or text-conditional foundation diffusion models as a filter. Finally, note that LDCE is compatible to and will benefit from future advancements of foundational diffusion models.

3.1 Counterfactual Generation in Latent Space

We propose to operate diffusion models in a perceptually equivalent, lower-dimensional latent space, unlike prior work that operated them on the pixel-level. Thereby, we have two benefits: (1) a *"focus on the important, semantic [instead of unimportant, high-frequency details] of the data"* [59, p. 4] and (2) as by-product *speed-ups* of counterfactual generation. Formally, we rewrite Eq. (1) as follows:

$$x^{\mathrm{CF}} = \mathcal{D}(z') \in \underset{z' \in \mathcal{Z} = \{\mathcal{E}(x) | x \in \mathcal{X}\}}{\arg\min} \lambda_c \mathcal{L}(f(\mathcal{D}(z')),\ y^{\mathrm{CF}}) + \lambda_d d(\mathcal{D}(z'),\ x^{\mathrm{F}}) \quad , \quad (9)$$



(a) Conceptual illustration.

(b) Technical implementation.

Fig. 3. 3(a): Our proposed consensus guidance mechanism employs a filtering approach of gradients, leveraging the implicit classifier of diffusion models as reference for semantic meaningfulness. The dashed line indicates forward diffusion, while the solid lines represent reverse diffusion. 3(b): Given the implicit classifier gradients and target classifier gradient, we compute a consensus mask that filters out gradients of the target classifier that are not aligned.

where $\mathcal{E}$ and $\mathcal{D}$ are the encoder or decoder of an autoencoder [25], respectively. Technically, we simply utilize recently popular latent diffusion models and adopt the following two-step procedure to generate counterfactuals:

1. **Abduction**: We add noise to the (f)actual image x^{F} through the forward diffusion process (Eq. (2)).
2. **Counterfactual generation**: We guide the diffusion process *s.t.* the counterfactual x^{CF} elicits a desired output y^{CF} from the target model f but remains close the (f)actual image x^{F} by computing the gradient w.r.t. the current intermediate representation of the diffusion process of Eq. (9) and adopt the classifier guidance approach (Eq. (6)).

3.2 Consensus Guidance Mechanism

While the two-step procedure from the end of the previous section already admits the generation of counterfactuals, we empirically found that it yielded blurry and unrealistic counterfactuals; refer to the LDCE variants without consensus (w/o consensus) in Fig. 4 or Tab. 1 (high FID scores). To combat this, we designed a novel consensus guidance mechanism. In a nutshell, it uses the gradient of the implicit classifier as a filter for the target model's gradients. On a high-level our consensus guidance mechanism ensures that the counterfactual generation process moves not just towards the target class distribution of y^{CF} but also stays on the image manifold, as illustrated in Fig. 3(a).

Our consensus guidance mechanism is inspired by the observation that both the gradient of the target model $\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c)$ (Eq. (6)), and the unconditional and conditional score functions of the class- or text-conditional foundation diffusion model $\epsilon_\theta^{(t)}(z_t|c) - \epsilon_\theta^{(t)}(z_t)$ (Eq. (7)) estimate $\nabla_x \log p(c|x)$. The main idea

of our consensus guidance mechanism is to leverage the gradients of the implicit classifier as a reference for semantic meaningfulness to filter out misaligned gradients of the target model. More specifically, we compute the angles α_i between the target model's gradients $\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c)$ and the difference of the conditional and unconditional scores $\epsilon_c - \epsilon_{uc}$ (*cf.*, Eq. (7)) for each non-overlapping patch, indexed by i:

$$\alpha_i = \angle[(-\sqrt{1-\alpha_t}\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c))_i,\ (\epsilon_c - \epsilon_{uc})_i] \quad . \tag{10}$$

We filter the gradients of the target model that have a larger angle α_i than the angular threshold γ:

$$\overbrace{-\sqrt{1-\alpha_t}\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c))_i} = \begin{cases} -\sqrt{1-\alpha_t}\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c))_i, & \alpha_i \leq \gamma \\ \mathbf{o}, & \alpha_i > \gamma \end{cases} , \tag{11}$$

where $\mathbf{o}$ is the overwrite value (in our case zeros) and $\overbrace{-\sqrt{1-\alpha_t}\nabla_{z_t}\mathcal{L}(f(\hat{x}_0),\ c))_i}$ is the guidance signal from our consensus guidance mechanism that is added to the unconditional score estimator $\epsilon_\theta^{(t)}(z_t)$ (*cf.*, Eq. (6)). Note that by setting the overwrite value $\mathbf{o}$ to zeros, only the target model and the unconditional score estimator, which is needed for regularization towards the data manifold, influence the counterfactual generation.

When using a class-conditional diffusion model, we refer to it as LDCE-cls, and LDCE-txt, when using a text-conditional diffusion model. In our experiments (Sec. 4), we validate that our consensus guidance mechanism improves the quality of counterfactuals while having only a small effect on the counterfactuals, *i.e.*, the target model is the driving factor for the changes.

4 Experiments

Datasets & models. We evaluated LDCE on ResNet-50 [28] trained on ImageNet [21] (on a subset of 10k images to ease computational costs and carbon footprint), DenseNet-121 [33] trained on CelebA HQ [44], (frozen) DINO-VIT-S/8 encoder with linear classifier trained on Oxford Flowers 102 [50], and OpenCLIP-VIT-B/32 [18] in the zero-shot setting on Oxford Pets [53]. All datasets have a image resolution of 256x256. We provide dataset and model licenses in Appendix B and further model details in Appendix C.

Implementation details. We based LDCE-cls on a class-conditional latent diffusion model trained on ImageNet [59] and LDCE-txt on a fine-tuned variant of Stable Diffusion V1.4 for 256x256 images [54]. Model licenses and links to the weights are provided in Appendix B. For text conditioning, we adopted CLIP-style text prompts [55] (refer to Appendix D for details). For our consensus guidance scheme, we used spatial regions of a size 1x1 and chose zeros as overwrite values $\mathbf{o}$. We used L1 as distance function d to promote sparse changes. We set the diffusion respacing to a factor of 2 to expedite counterfactual generation at the cost of image quality. We set the weighting factor η to 2. We optimized other hyperparameters (diffusion steps T and the other weighing factors $\lambda_c,\ \gamma,\ \lambda_d$) on a few examples per dataset and provide them in Appendix D.

Selection of counterfactual target class. We used two protocols: **(a) Semantic Hierarchy:** we randomly sampled one of the top-4 closest classes based on the shortest path in the WordNet hierarchy [49]. **(b) Representational Similarity:** we computed the pairwise cosine similarities for all (f)actual images using SimSiam [17] and randomly sampled a class from the top-5 instances from a different class. Unlike the former, our proposed representational similarity based selection does not require any domain expertise. We adopted the former for ImageNet and the latter for Oxford Pets and Flowers 102. For CelebA HQ, we selected the opposite binary target class.

Quantitative evaluation. The evaluation of counterfactuals is inherently challenging: what makes a good counterfactual is arguably very subjective and depends on the context. Despite this, we used various quantitative metrics covering commonly acknowledged desiderata. Appendix E provides a concise overview.

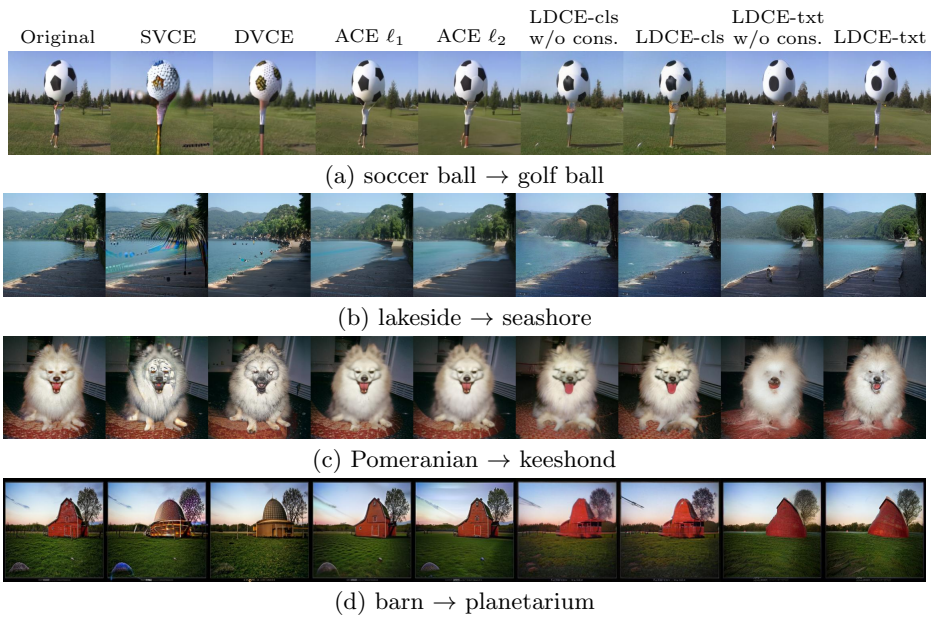
(a) Validity: To quantify whether the counterfactual approach is capable of flipping the target model's classification, we used the Flip Ratio (FR). Further, we used the S^3 criterion [37] that computes cosine similarity between the (f)actual x^{F} and counterfactual x^{CF}. While it has been introduced as a closeness criterion by Jeanneret *et al.* [37] (the higher the better), we found strong negative correlation with FR, *i.e.*, rank correlation of -0.83 using the numbers from Tab. 2. Beyond that, note that lower S^3 indicates that the counterfactual method made more semantic changes (something we want), as this is what SimSiam's features (used in S^3) are trained for. Thus, we say that lower S^3 is better.

(b) Closeness & sparsity: A counterfactual should remain close to the (f)actual input and should have sparse changes. For closeness, we used the L2 norm and propose VQ-L2 (L2 distance in the space of an autoencoder). Unlike L2, VQ-L2 is less affected by unimportant, high-frequency image details. For sparsity, we adopted various metrics from the literature: COUT [38], Face Similarity (FS) [37], Mean Number of Attributed Changed (MNAC) [58], and Correlation Difference (CD) [36]. Note that similar as for S^3, we found that FS negatively correlates with COUT (Spearman rank correlation of -0.5 for the numbers from Tab. 6). Thus, we say that lower FS is better.

(c) Realism: We adopted FID and sFID [37] to assess realism of counterfactuals. Note that sFID addresses a bias towards not changing the counterfactual due to the closeness desiderata. Further, we propose to use precision and recall [42]. Precision indicates whether a counterfactual falls within the support of the (f)actual images, while recall indicates if the global distribution of counterfactuals supports random (f)actual images. Thus, precision and recall provide local or global indicators of the fidelity of counterfactuals, respectively.

4.1 Qualitative Evaluation

Figs. 1, 4 and 5 as well as Figs. 11 to 14 in Appendix I show extensive qualitative results for both LDCE-cls and LDCE-txt across a diverse range of models (from convolutional networks to transformers) trained on various real-world data



(a) soccer ball → golf ball

(b) lakeside → seashore

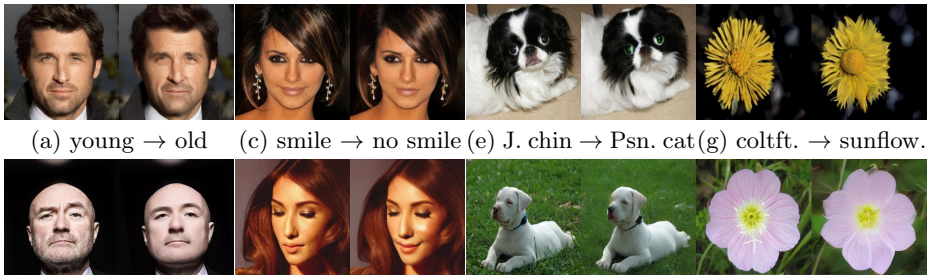
(c) Pomeranian → keeshond

(d) barn → planetarium

Fig. 4. Qualitative comparisons analyzing ResNet-50 as target model on ImageNet. Left to right: original image, counterfactual images for SVCE, DVCE, ACE ℓ_1, ACE ℓ_2, LDCE-cls w/o consensus, LDCE-cls, LDCE-txt w/o consensus, and LDCE-txt.

(from ImageNet to CelebA-HQ, Oxford Pets, or Flowers-102) with diverse learning paradigms (from supervision, to vision-only or vision-language representation learning). First, we observe that LDCE-txt introduces local changes (*e.g.*, Fig. 1(b)) while is also capable of more global changes (*e.g.*, Fig. 1(a)). Moreover, we find that LDCE-txt is also able to make intricate changes to, *e.g.*, the geometry of flower petals (Fig. 1(d) or the rightmost column of Fig. 5), while not even being explicitly trained on such a flower image distribution. For LDCE-cls, we similarly observe that it is capable of local as well as global changes (Fig. 4). As expected, the generated counterfactuals of LDCE-cls on ImageNet are of higher quality than the ones of LDCE-txt. Despite this we stress that LDCE-cls (and other previous work) must train the generative model on the same (or similar) image distribution, while LDCE-txt is universally applicable. Figs. 1 and 5 demonstrate this. Moreover, Figs. 5(a) to 5(d) & Fig. 12 in Appendix I demonstrate the ability of LDCE-txt to capture and manipulate distinctive facial features: LDCE-txt inserts or removes local features such as wrinkles, dimples, and eye bags when moving along the smile and age attributes.

Fig. 4 compares generated counterfactuals by our LDCE variants to recent methods from the literature: SVCE [11], DVCE [3], and ACE [37]. We observe that SVCE often generates high-frequency (*e.g.*, Fig. 4(a)) or copy-paste-like artifacts (Fig. 4(c)). DVCE tends to change images altogether (*e.g.*, Fig. 4(d)), making it difficult to identify which features caused a change in ResNet50's clas-



(a) young → old (c) smile → no smile (e) J. chin → Psn. cat (g) coltft. → sunflow.

(b) old → young (d) no smile → smile (f) Am. bull. → bgl. (h) pink prim → lotus

Fig. 5. LDCE-txt can generate counterfactuals for a wide range of models and datasets: DenseNet-121, OpenCLIP, and (frozen) DINO with linear classifier on CelebA HQ, Oxford Pets, or Flowers 102, respectively. Left: original image. Right: counterfactual.

sification. Moreover, note that DVCE is substantially confounded by its auxiliary model; as shown in Fig. 2 or Fig. 8 in Appendix A. Lastly, ACE keeps the (f)actual image almost untouched, making it hard to identify important features that influence ResNet50's classification. In contrast to SVCE, DVCE, and ACE, both LDCE variants generate high-fidelity counterfactuals that stay close to the (f)actual image, while still highlighting important features for classification of the model (as we will see in Sec. 5 this is an useful property). Further, note that LDCE-txt is *universally applicable beyond a specific data distribution*, while LDCE-cls and aforementioned methods from the literature require the generative model to be trained on the same (or similar) data distribution.

Evolution, diversity, and failure modes. We find that counterfactual generation gradually evolves from coarse (low-frequency) features (*e.g.*, blobs or shapes) at the earlier time steps towards more intricate (high-frequency) details (*e.g.*, textures) at later time steps; see Appendix G. Moreover, LDCE can generative a diverse set of counterfactuals by introducing stochasticity in the abduction step (different seeds) as demonstrated in Appendix H. Lastly, we also observed failure modes (*e.g.*, distorted secondary objects). We suspect that some of these limitations are inherited from the used diffusion model and, in part, to domain shift. We further discuss these challenges in Sec. 6 and Appendix K.

4.2 Quantitative Evaluation

ImageNet results (Tabs. 1 and 2). For fair comparisons, we followed the the evaluation protocol of Augustin *et al.* [3] for $\ell_{1.5}$-SVCE [11] and DVCE [3], and the evaluation protocol of Jeanneret *et al.* [37] for ACE [37]. Note that $\ell_{1.5}$-SVCE does not use the target model but an adversarially robust model (multiple-norm robust ResNet-50 [19]) for counterfactual generation. Also note that DVCE's counterfactuals are substantially influenced by its auxiliary model (Figs. 2 and 8). Thus, their results must be taken with caution.

Table 1. Quantiative comparison of our class- (LDCE-cls) and text-conditional (LDCE-txt) LDCE variants to SVCE and DVCE on ImageNet using ResNet-50.

Method	w/o aux. model	universally applicable*	VQ-L2	L2	FID	sFID	precision	recall	FR	avg. rank
$\ell_{1.5}$-SVCE† [11]	-	-	27	**25**	22.44	28.44	**83.66**	**95.75**	83.8	**2.71**
DVCE [3]	-	-	38	38	*15.1*	*21.14*	74.05	85.12	**99.6**	3.29
LDCE-cls w/o cons.	✓	-	*36*	41	21.70	27.10	78.93	93.85	98.4	4.0
LDCE-cls	✓	-	*36*	42	**14.03**	**19.25**	**81.67**	**94.59**	83.1	*3.07*
LDCE-txt w/o cons.	✓	✓	42	**31**	38.0	44.23	77.89	82.40	*98.5*	4.71
LDCE-txt	✓	✓	**31**	41	21.0	26.5	79.18	90.34	84.4	3.21

†: generates counterfactuals with an adversarially robust ResNet-50 only, *: diffusion model need not be trained on ImageNet.

Table 2. Comparison of LDCE-cls and LDCE-txt (diffusion model not trained on ImageNet) to ACE on ImageNet with ResNet-50.

Method	FID (↓)	sFID (↓)	S^3 (↓)	COUT (↑)	FR (↑)	avg rank (↓)
Zebra – Sorrel						
ACE ℓ_1	84.5	122.7	0.92	-0.45	47.0	3.6
ACE ℓ_2	**67.7**	**98.4**	0.90	-0.25	81.0	2.1
LDCE-cls	93.6	113.8	0.78	**-0.06**	**88.0**	**1.8**
LDCE-txt	98.1	121.7	**0.71**	-0.2097	81.0	2.5
Cheetah – Cougar						
ACE ℓ_1	**70.2**	100.5	0.91	0.02	77.0	3.0
ACE ℓ_2	74.1	102.5	0.88	0.12	95.0	3.0
LDCE-cls	71.0	**91.8**	0.62	**0.51**	**100.0**	**1.4**
LDCE-txt	89.4	110.8	**0.59**	0.34	98.0	2.6
Egyptian Cat – Persian Cat						
ACE ℓ_1	**93.6**	156.7	0.85	0.25	85.0	3.0
ACE ℓ_2	107.3	160.4	0.78	0.34	97.0	3.0
LDCE-cls	102.3	**140.0**	0.63	0.52	**99.0**	**1.7**
LDCE-txt	121.0	161.5	**0.61**	**0.56**	**99.0**	2.3

Tab. 1 shows that LDCE-cls achieves superior realism (FID, sFID, precision, recall) of counterfactuals, while DVCE only outperforms LDCE-cls and LDCE-txt w.r.t. the flip ratio (FR). Unsurprisingly, we find that SVCE generates counterfactuals that are closer to the (f)actual image (lower Lp norms) since it specifically constraints optimization within a $\ell_{1.5}$-ball. It is also unsurprising that both LDCE variants have higher L2, as L2 is confounded by unimportant, high-frequency image details. VQ-L2 is more agnostic to such and aligns better with the qualitative inspection from Sec. 4.1. Lastly, the beneficial effect of our proposed consensus guidance scheme is also highlighted, as it substantially improves realism of counterfactuals.

Tab. 2 shows that both LDCE variants are consistently superior to ACE, with the only exception of FID. However, as ACE enforces sparse changes, counterfactual remain close to their respective (f)actual images, leading to low FID scores (as also seen in Sec. 4.1). On the contrary, sFID is higher for ACE than our LDCE variants, which accounts for such shortcut minimization of FID.

We provide quantitative results on CelebA-HQ in Appendix F. Importantly, despite LDCE-txt is not specifically trained on facial data, it achieves competitive performance (*i.e.*, even better average rank) to previous baselines.

5 Applications

5.1 Identifying and Fixing Model Errors

Compared to usual image generation and editing that focus on high-quality, counterfactual generation prioritizes understanding model behavior while maintaining visual quality. We apply our method to investigate ResNet-50's misclassifications on ImageNet by generating counterfactuals towards the correctly misclassified class and visually inspecting the altered features that correct the

(a) kite → bald eagle (b) maraca → wooden spoon (c) running shoe → sandal

Fig. 6. Counterfactuals aid to understand classification errors (here the ones from ResNet-50 trained on ImageNet). Left: misclassified original image. Right: correctly classified counterfactual. Red ellipses were added manually.

misclassification. Fig. 6 shows that ResNet-50 may misclassify young bald eagles primarily due to the absence of their distinctive white heads and tails. We speculate that this misclassification is due to the predominance of images of older bald eagles, which only develop the characteristic white heads as they age. Similarly, we found that painted wooden spoons may be misclassified as maraca, while closed-toe sandals may be confused with running shoes.

To substantiate our findings, we create a small set of 50 samples from the Internet with such erroneous characteristic. As expected, we found high error rates on this set (bald eagle: 88 %, wooden spoon: 74 %, sandal: 48 %); confirming our findings. As a fix, we finetuned the last linear layer of ResNet-50 and splitting the 50 images into a equally-sized train and test set (details provided in Appendix J). While ImageNet accuracy barely changed, *i.e.*, ca. 0.2% improvement, above model error rates on the test sets substantially reduced by 40 %, 32 %, or 16 %, respectively.

5.2 Finding Spurious Correlations

Another possible application of LDCE-txt is to identify spurious correlations that a classifier may have picked up during training. Specifically, we can compare the (f)actual *vs*. counterfactual image focusing on the supervised DenseNet-121 trained on CelebA HQ. Our method identified a known spurious feature is that older persons tend to wear glasses and vise-versa as seen in Fig. 7. To validate that DenseNet-121 relies on this spurious feature, we used InstructPix2Pix [13] to add and remove glasses for young or old persons, respectively. We used 100 samples (50 young and 50 old) and measured the average treatment effect through the sigmoid outputs change. We find that adding glasses to young persons shifts sigmoid values of DenseNet-121 by 7.6 % towards an older look, while removing it from older individuals shifts values by 8.7 % towards a younger look.

(a) young → old

(b) old → young

Fig. 7. Counterfactuals reveal the influence of glasses on age for DenseNet-121 on CelebA HQ.

6 Limitations

Our method inherits the iterative generative process of diffusion models which bottlenecks the generation speed and, thus, hinders real-time, interactive applications.[1] However, advancements in distilling diffusion models [61,72,48,66] or speed-up techniques [20,10] offer promise in mitigating this limitation. For instance, Sauer *et al.* [66] only required a single diffusion step, while maintaining high image quality. Another limitation is the requirement for hyperparameter optimization due to different needs in distinct use cases. While our approach is simple and only has a limited number of hyperparameters, making this process very swift, it is still necessary. Beyond that, contemporary foundation diffusion models, despite expanding data coverage [67], may underperform in specialized domains, *e.g.*, biomedical data or exhibit (social) biases [8,46]. Lastly, we acknowledge that our consensus guidance mechanism can suppress the signal of the model under investigation but importantly not add any features unlike the adversarially robust model in DVCE (see Fig. 8). We found that the diversity of our method can mitigate this problem as suppressed features for one counterfactual, may not be suppressed for another; refer to Appendix H.

7 Conclusion

We presented LDCE to generate semantically meaningful counterfactual explanations using class- or text-conditional (foundation) diffusion models, combined with a novel consensus guidance mechanism. We show LDCE's universal applicability across diverse models trained on diverse datasets, and its usage for understand and resolve model errors.

Acknowledgments. This work was funded by the Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (BMUV, German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection) based on a resolution of the German Bundestag (67KI2029A) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828. K.F. acknowledges support by the Deutscher Akademischer Austauschdienst (DAAD, German Academic Exchange Service) as part of the ELIZA program.

Author contributions. Project idea: S.S.; project lead: S.S. & K.F.; conceptualization of consensus guidance mechanism: K.F. with input from S.S.; method implementation: K.F. & S.S.; hyperparameter optimization: K.F.; implementation and execution of experiments: S.S. & K.F. with input from M.A. & T.B.; visualization: S.S. with input from K.F. (experimental results) & M.A. with input from S.S. & K.F. (Fig. 3); interpretation of findings: K.F. & S.S. with input from M.A. & T.B.; guidance & feedback: M.A. & T.B.; funding acquisition: T.B.; paper writing: S.S. & K.F. crafted the first draft and all authors contributed to the final version.

[1] Note that LDCE still yields speed-ups to prior work, *e.g.*, 34% to DVCE.

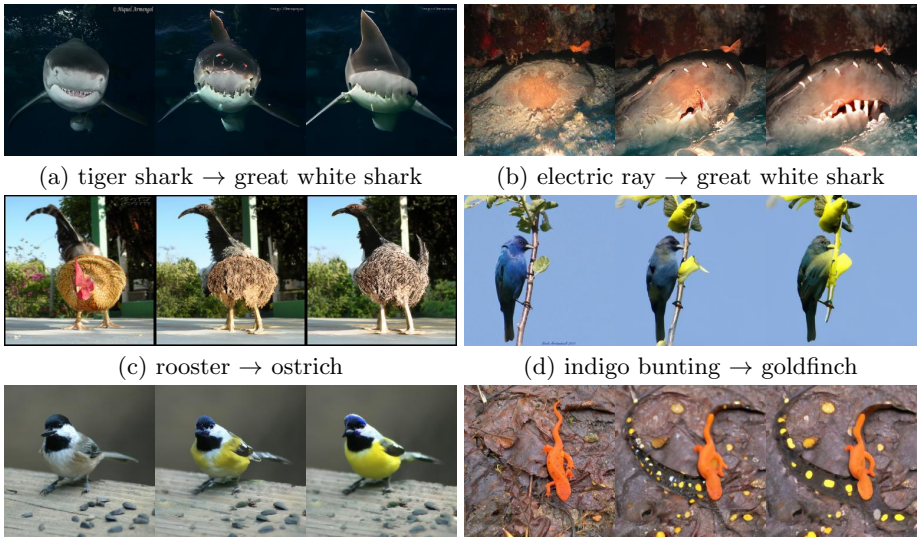
References

1. Akula, A., Wang, S., Zhu, S.C.: CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines. In: AAAI (2020)
2. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. Information Fusion (2020)
3. Augustin, M., Boreiko, V., Croce, F., Hein, M.: Diffusion Visual Counterfactual Explanations. In: NeurIPS (2022)
4. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: CVPR (2022)
5. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PloS one (2015)
6. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. arXiv (2023)
7. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network Dissection: Quantifying Interpretability of Deep Visual Representations. In: CVPR (2017)
8. Bianchi, F., Kalluri, P., Durmus, E., Ladhak, F., Cheng, M., Nozza, D., Hashimoto, T., Jurafsky, D., Zou, J., Caliskan, A.: Easily accessible text-to-image generation amplifies demographic stereotypes at large scale. In: FAccT (2023)
9. Böhle, M., Fritz, M., Schiele, B.: B-cos Networks: Alignment is All We Need for Interpretability. In: CVPR (2022)
10. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token Merging: Your ViT But Faster. In: ICLR (2023)
11. Boreiko, V., Augustin, M., Croce, F., Berens, P., Hein, M.: Sparse Visual Counterfactual Explanations in Image Space. In: GCPR (2022)
12. Brendel, W., Bethge, M.: Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In: ICLR (2019)
13. Brooks, T., Holynski, A., Efros, A.A.: InstructPix2Pix: Learning to Follow Image Editing Instructions. In: CVPR (2023)
14. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age. In: IEEE International Conference on Automatic Face & Gesture Recognition (2018)
15. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021)
16. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This Looks Like That: Deep Learning for Interpretable Image Recognition. NeurIPS (2019)
17. Chen, X., He, K.: Exploring Simple Siamese Representation Learning. In: CVPR (2021)
18. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible Scaling Laws for Contrastive Language-Image Learning. In: CVPR (2023)
19. Croce, F., Hein, M.: Adversarial Robustness against Multiple and Single l_p-Threat Models via Quick Fine-Tuning of Robust Classifiers. ICML (2022)
20. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: FLASHATTENTION: Fast and Memory-Efficient Exact Attention with IO-Awareness. In: NeurIPS (2022)
21. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)

22. Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis. In: NeurIPS (2021)
23. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021)
24. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing Higher-Layer Features of a Deep Network. University of Montreal (2009)
25. Esser, P., Rombach, R., Ommer, B.: Taming Transformers for High-Resolution Image Synthesis. In: CVPR (2021)
26. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM (2020)
27. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual Visual Explanations. In: ICML (2019)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
29. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. NeurIPS (2017)
30. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: NeurIPS (2020)
31. Ho, J., Salimans, T.: Classifier-Free Diffusion Guidance. NeurIPS Workshop (2022)
32. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models. ICLR (2022)
33. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. In: CVPR (2017)
34. Hvilshøj, F., Iosifidis, A., Assent, I.: ECINN: Efficient Counterfactuals from Invertible Neural Networks. In: BMVC (2021)
35. Jacob, P., Zablocki, É., Ben-Younes, H., Chen, M., Pérez, P., Cord, M.: STEEX: Steering Counterfactual Explanations with Semantics. In: ECCV (2022)
36. Jeanneret, G., Simon, L., Jurie, F.: Diffusion Models for Counterfactual Explanations. In: ACCV (2022)
37. Jeanneret, G., Simon, L., Jurie, F.: Adversarial Counterfactual Visual Explanations. In: CVPR (2023)
38. Khorram, S., Fuxin, L.: Cycle-Consistent Counterfactuals by Latent Transformations. In: CVPR (2022)
39. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In: ICML (2018)
40. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. ICLR (2014)
41. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept Bottleneck Models. In: ICML (2020)
42. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved Precision and Recall Metric for Assessing Generative Models. In: NeurIPS (2019)
43. Lang, O., Gandelsman, Y., Yarom, M., Wald, Y., Elidan, G., Hassidim, A., Freeman, W.T., Isola, P., Globerson, A., Irani, M., et al.: Explaining in Style: Training a GAN to explain a classifier in StyleSpace. In: CVPR (2021)
44. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In: CVPR (2020)

45. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: ICLR (2019)
46. Luccioni, A.S., Akiki, C., Mitchell, M., Jernite, Y.: Stable Bias: Analyzing Societal Representations in Diffusion Models. In: NeurIPS (2023)
47. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. NeurIPS (2017)
48. Meng, C., Gao, R., Kingma, D.P., Ermon, S., Ho, J., Salimans, T.: On Distillation of Guided Diffusion Models. In: CVPR (2023)
49. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM (1995)
50. Nilsback, M.E., Zisserman, A.: Automated Flower Classification over a Large Number of Classes. In: Indian Conference on Computer Vision, Graphics & Image Processing (2008)
51. Olah, C., Mordvintsev, A., Schubert, L.: Feature Visualization. Distill (2017)
52. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision. TMLR (2024)
53. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats And Dogs. In: CVPR (2012)
54. Pinkey, J.: miniSD. https://huggingface.co/justinpinkney/miniSD (2023)
55. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models From Natural Language Supervision. In: ICML (2021)
56. Rezende, D., Mohamed, S.: Variational Inference with Normalizing Flows. In: ICML (2015)
57. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In: ICML (2014)
58. Rodriguez, P., Caccia, M., Lacoste, A., Zamparo, L., Laradji, I., Charlin, L., Vazquez, D.: Beyond Trivial Counterfactual Explanations with Diverse Valuable Explanations. In: CVPR (2021)
59. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: CVPR (2022)
60. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI (2015)
61. Salimans, T., Ho, J.: Progressive Distillation for Fast Sampling of Diffusion Models. In: ICLR (2022)
62. Samangouei, P., Saeedi, A., Nakagawa, L., Silberman, N.: ExplainGAN: Model Explanation via Decision Boundary Crossing Transformations. In: ECCV (2018)
63. Sanchez, P., Tsaftaris, S.A.: Diffusion Causal Models for Counterfactual Estimation. In: CLeaR (2022)
64. Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Image Synthesis with a Single (Robust) Classifier. NeurIPS (2019)
65. Sauer, A., Geiger, A.: Counterfactual Generative Networks. In: ICLR (2021)
66. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial Diffusion Distillation. arXiv preprint arXiv:2311.17042 (2023)
67. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. arXiv (2021)

68. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: CVPR (2017)
69. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ICLR Workshop (2014)
70. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: ICML (2015)
71. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: ICLR (2021)
72. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency Models. In: ICML (2023)
73. Song, Y., Ermon, S.: Generative Modeling by Estimating Gradients of the Data Distribution. In: NeurIPS (2019)
74. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations. ICLR (2021)
75. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: CVPR (2016)
76. Thiagarajan, J., Narayanaswamy, V.S., Rajan, D., Liang, J., Chaudhari, A., Spanias, A.: Designing Counterfactual Generators using Deep Model Inversion. In: NeurIPS (2021)
77. TorchVision maintainers and contributors: TorchVision: PyTorch's Computer Vision library. https://github.com/pytorch/vision (2016)
78. Van Looveren, A., Klaise, J.: Interpretable Counterfactual Explanations Guided by Prototypes. In: ECML PKDD (2021)
79. Vandenhende, S., Mahajan, D., Radenovic, F., Ghadiyaram, D.: Making Heads or Tails: Towards Semantically Consistent Visual Counterfactuals. In: ECCV (2022)
80. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. Harv. JL & Tech. (2017)
81. Wallace, B., Gokul, A., Ermon, S., Naik, N.: End-to-End Diffusion Latent Optimization Improves Classifier Guidance. In: ICCV (2023)
82. Wang, P., Li, Y., Singh, K.K., Lu, J., Vasconcelos, N.: IMAGINE: Image Synthesis by Image-Guided Model Inversion. In: CVPR (2021)



(a) tiger shark → great white shark (b) electric ray → great white shark

(c) rooster → ostrich (d) indigo bunting → goldfinch

(e) chickadee → goldfinch (f) eft → spotted salamander

Fig. 8. Qualitative examples illustrating the marginal influence of the target model in DVCE. From left to right: original image, counterfactual images generated using DVCE with cone projection using an angular threshold of 30° and the robust classifier, and DVCE using the robust classifier only, *i.e.*, without the target model.

A Influence of the Adversarial Robust Model in DVCE

Fig. 8 shows that the auxiliary robust model has a profound impact on the counterfactuals generated by DVCE [3]. Consequently, we *cannot* attribute the changes of counterfactual explanations solely to the target model since they are substantially confounded by the auxiliary adversarially robust model.

B Dataset and Model Licenses

Tabs. 3 and 4 provide licenses and URLs of the datasets or models used in our experimental evaluation, respectively. Our implementation of LDCE is built upon Rombach *et al.* [59] (License: Open RAIL-M, URL: https://github.com/CompVis/stable-diffusion) and provided at https://github.com/lmb-freiburg/ldce (License: MIT). Note that since all datasets have an image resolution of 256x256, we used a finetuned version of stable diffusion on this resolution, provided by Pinkey [54] (License: CreativeML Open RAIL-M, URL: https://huggingface.co/justinpinkney/miniSD).

C Model Details

Below, we provide model details:

Table 3. Licenses and URLs for the datasets used in our experiments.

Dataset	License	URL
CelebAMask-HQ [44]	CC BY 4.0	https://github.com/switchablenorms/CelebAMask-HQ
Oxford Flowers 102 [50]	GNU	https://www.robots.ox.ac.uk/~vgg/data/flowers/102/
ImageNet [21]	Custom	https://www.image-net.org/index.php
Oxford Pet [53]	CC BY-SA 4.0	https://www.robots.ox.ac.uk/~vgg/data/pets/

Table 4. Licenses and URLs for the target and diffusion models used in our experiments.

Models	License	URL
ImageNet class-conditional LDM [59]	MIT	https://github.com/CompVis/latent-diffusion
Mini Stable diffusion 1.4 [54]	CreativeML Open RAIL-M [59]	https://huggingface.co/justinpinkney/miniSD
ResNet-50 for ImageNet [28,77]	BSD 3	https://github.com/pytorch/vision
Adv. robust ResNet-50 for ImageNet [11]	MIT	https://github.com/valentyn1boreiko/SVCEs_code
DenseNet-121 [33] for CelebA HQ [35]	Apache 2	https://github.com/valeoai/STEEX
DINO for Oxford Flowers 102 [15]	Apache 2	https://github.com/facebookresearch/dino
OpenCLIP for Oxford Pets [18]	Custom	https://github.com/mlfoundations/open_clip
SimSiam [17]	CC BY-NC 4.0	https://github.com/facebookresearch/simsiam
CelebA HQ Oracle [35]	Apache 2	https://github.com/valeoai/STEEX
Ported VGGFace2 model from [14]	MIT	https://github.com/cydonia999/VGGFace2-pytorch

– **ResNet50 [28] on ImageNet [21]**: We used the pretrained ResNet-50 model provided by torchvision [77].
– **DenseNet-121 [33] on CelebA HQ [44]**: We used the pretrained DenseNet-121 model provided by Jacob *et al.* [35].
– **OpenCLIP [55,18] on Oxford Pets [53]**: We used the provided weights of OpenCLIP ViT-B/32 [23], and achieved a top-1 zero-shot classification accuracy of 90.5 % using CLIP-style prompts [55].
– **DINO+linear [15] on Oxford Flowers 102 [50]**: We used a frozen DINO ViT-S/8 model, added a trainable linear classifier, and trained the linear classifier on Oxford Flowers 102 for 30 epochs. We used SGD with a learning rate of 0.001 and momentum of 0.9, and cosine annealing [45]. The model achieved a top-1 classification accuracy of 92.82 %.

D Hyperparameters

Tab. 5 provides our manually tuned hyperparameters. For our LDCE-txt, we transform the counterfactual target classes y^{CF} to CLIP-style text prompts [55], as follows:
– ImageNet: `a photo of a {category name}.`
– CelebA HQ: `a photo of a {attribute name} person.` (`attribute name` $\in$ {non-smiling, smiling, old, young}).
– Oxford Flowers 102: `a photo of a {category name}, a type of flower.`
– Oxford Pets: `a photo of a {category name}, a type of pet.`

Table 5. Manually-tuned hyperparameters of LDCE-cls and LDCE-txt.

	LDCE-cls	LDCE-txt			
Hyperparameter	ImageNet	ImageNet	CelebA HQ	Flowers	Oxford Pets
consensus threshold γ	45°	50°	90°	45°	45°
starts timestep T	191	191	160	250	191
classifier weighting λ_c	2.3	3.95	3.95	3.4	4.2
distance weighting λ_d	0.3	1.2	3.5	1.2	2.4

We note that more engineered prompts may yield better counterfactual explanations, but we leave such studies for future work.

E Evaluation Criteria for Counterfactual Explanations

In this section, we discuss the evaluation criteria used to quantitatively assess the quality of counterfactual explanations. Even though quantitative assessment of counterfactual explanations is arguably very subjective, these evaluation criteria build a basis of quantitative evaluation based on the commonly recognized desiderata validity, closeness, and realism.

Flip Ratio (FR)

This criterion focuses on assessing the validity of N counterfactual explanations by quantifying the degree to which the original class label y_i^{F} of the original image x_i^{F} flips the target classifier's prediction f to the counterfactual target class y_i^{CF} for the counterfactual image x_i^{CF}:

$$\mathrm{FR} = \frac{\sum_{i=1}^{N} \mathbb{I}(f(x_i^{\mathrm{CF}}) = y_i^{\mathrm{CF}})}{N} \quad , \tag{12}$$

where $\mathbb{I}$ is the indicator function.

Counterfactual Transition (COUT)

Counterfactual Transition (COUT) [38] measures the sparsity of changes in counterfactual explanations, incorporating validity and sparsity aspects. It quantifies the impact of perturbations introduced to the (f)actual image x^{F} using a normalized mask m that represents relative changes compared to the counterfactual image x^{CF}, *i.e.*, $m = \delta(||x^{\mathrm{F}} - x^{\mathrm{CF}}||_1)$, where δ normalizes the absolute difference to $[0, 1]$. We progressively perturb x^{F} by inserting top-ranked pixel batches from x^{CF} based on these sorted mask values.

For each perturbation step $t \in \{0, \ldots, T\}$, we record the output scores of the classifier f for the (f)actual class y^{F} and the counterfactual class y^{CF} throughout

the transition from $x^0 = x^{\mathrm{F}}$ to $x^T = x^{\mathrm{CF}}$. From this, we can compute the COUT score:

$$\mathrm{COUT} = \mathrm{AUPC}(y^{\mathrm{CF}}) - \mathrm{AUPC}(y^{\mathrm{F}}) \in [-1, 1] \quad , \tag{13}$$

where the area under the Perturbation Curve (AUPC) for each class $y \in \{y^{\mathrm{F}}, y^{\mathrm{CF}}\}$ is defined as follows:

$$\mathrm{AUPC}(y) = \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{2}\left(f_y\left(x^{(t)}\right) + f_y\left(x^{(t+1)}\right)\right) \in [0, 1] \quad . \tag{14}$$

A high COUT score indicates that a counterfactual generation approach finds sparse changes that flip classifiers' output to the counterfactual class.

SimSiam Similarity (S3)

This criterion measures the cosine similarity between a counterfactual image x^{CF} and its corresponding (f)actual image x^{F} in the feature space of a self-supervised SimSiam model S [17]:

$$S^3(x^{\mathrm{CF}}, x^{\mathrm{F}}) = \frac{\mathcal{S}(x^{\mathrm{CF}}) \cdot \mathcal{S}(x^{\mathrm{F}})}{\|\mathcal{S}(x^{\mathrm{CF}})\|\|\mathcal{S}(x^{\mathrm{F}})\|} \quad . \tag{15}$$

Lp norms

Lp norms serve as closeness criteria by quantifying the magnitude of the changes between the counterfactual image x_i^{CF} and original image x_i^{F}:

$$L_p = \frac{1}{N}\sum_{i=1}^{N}\|d_i\|_p \quad , \tag{16}$$

where $0 < p \leq \infty$ and C, H, W are the number of channels, image height, and image width, respectively, and

$$\|d_i\|_p = \left(\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}|x_{i,c,h,w}^{\mathrm{F}} - x_{i,c,h,w}^{\mathrm{CF}}|^p\right)^{\frac{1}{p}} \quad . \tag{17}$$

Note that *Lp* norms can be confounded by unimportant, high-frequency image details.

Mean Number of Attribute Changes (MNAC)

Mean Number of Attribute Changes (MNAC) quantifies the average number of attributes modified in the generated counterfactual explanations. It uses an oracle model O_a (*i.e.*, VGGFace2 model [14]) which predicts the probability of

each attribute $a \in \mathcal{A}$, where $\mathcal{A}$ is the entire attributes space. MNAC is defined as follows:

$$\text{MNAC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}}\left[\mathbb{I}\left(\mathbb{I}\left(O_a(x_i^{\text{CF}}) > \beta\right) \neq \mathbb{I}\left(O_a(x_i^{\text{F}}) > \beta\right)\right)\right] \quad , \tag{18}$$

where β is a threshold (typically set to 0.5) that determines the presence of attributes. MNAC quantifies the counterfactual method's changes to the query attribute q, while remaining independent of other attributes. However, a higher MNAC value can wrongly assign accountability for spurious correlations to the counterfactual approach, when in fact they may be artifacts of the classifier.

Correlation Difference (CD)

Correlation Difference (CD) [36] evaluates the ability of counterfactual methods to identify spurious correlations by comparing the Pearson correlation coefficient $c^{q,a}(x)$, of the relative attribute changes δ^q and δ^a, before and after applying the counterfactual method. For each attribute a, the attribute change δ^a is computed between pairs of samples i and j, as $\delta^a_{i,j} = \hat{y}^a_i - \hat{y}^a_j$, using the predicted probabilities $\hat{y}^a_i$ and $\hat{y}^a_j$ from the oracle model O (*i.e.*, VGGFace2 model [14]). The CD for a query attribute q is then computed as:

$$\text{CD}_q = \frac{1}{N}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}}\left|c^{q,a}(x^{\text{CF}}) - c^{q,a}(x^{\text{F}})\right| \quad . \tag{19}$$

Face Similarity (FS)

Jeanneret *et al.* [37] proposed Face Similarity (FS) that addresses thresholding issues and the abrupt transitions in classifier decisions in Face Verification Accuracy (FVA) when comparing the (f)actual image x^{F} and its corresponding counterfactual x^{CF}. FS directly measures the cosine similarity between the feature encodings, providing a more continuous assessment (similar to S^3).

Fréchet Inception Distances (FID & sFID)

Fréchet Inception Distance (FID) [29] and split FID (sFID) [37] evaluate the realism of generated counterfactual images by measuring the distance on the dataset level between the InceptionV3 [75] feature distributions of real and generated images:

$$\text{FID} = \|\mu_{\text{F}} - \mu_{\text{CF}}\|_2^2 + \text{Tr}(\Sigma_{\text{F}} + \Sigma_{\text{CF}} - 2\sqrt{\Sigma_{\text{F}}\Sigma_{\text{CF}}}) \quad , \tag{20}$$

where $\mu_{\text{F}}, \mu_{\text{CF}}$ and $\Sigma_{\text{F}}, \Sigma_{\text{CF}}$ are the feature-wise mean or covariance matrices of the InceptionV3 feature distributions of real and generated images, respectively. However, there is a strong bias in FID towards counterfactual approaches that

barely alter the pixels of the (f)actual inputs. To address this, Jeanneret *et al*. [37] proposed to split the dataset into two subsets: generate counterfactuals for one subset, compute FID between those counterfactuals and the (f)actual inputs of the untouched subset, and vice versa, and then take the mean of the resulting FIDs.

Precision & Recall

We adopted the implementation of precision & recall from Kynkäänniemi *et al*. [42]. Specifically, the (f)actual and counterfactual are embedded into a feature space, denoted by $\phi^{\mathrm{F}} \in \mathbf{\Phi}^{\mathrm{F}}$ or $\phi^{\mathrm{CF}} \in \mathbf{\Phi}^{\mathrm{CF}}$, respectively. We utilize DINOv2 [52] features for the precision and recall metrics, as it offers a more general representation space than InceptionV3 [75]. Kynkäänniemi *et al*. determine if a sample ϕ is (locally) supported in the data manifold by estimating the data manifold by the volume of the hypersphere with the distance to the kth nearest neighbor as radius:

$$g(\phi, \Phi) = \begin{cases} 1 \text{ if } ||\phi - \phi'||_2 \leq ||\phi' - \mathrm{NN}_k(\phi', \Phi)||_2 \text{ for at least one } \phi' \in \Phi \\ 0 \text{ otherwise,} \end{cases} \tag{21}$$

where $\mathrm{NN}_k(\phi', \Phi)$ returns the kth nearest neighbor of $\phi' \in \Phi$. In other words, Eq. (21) determines if a given sample ϕ is within the local hypervolume (support) of the features vectors from Φ. Finally, to adapt to our counterfactual setting, we define precision and recall as follows:

$$\mathrm{precision}(\Phi^{\mathrm{F}}, \Phi^{\mathrm{CF}}) = \frac{1}{\Phi^{\mathrm{CF}}} \sum_{\phi^{\mathrm{CF}} \in \Phi^{\mathrm{CF}}} g(\phi^{\mathrm{CF}}, \Phi^{\mathrm{F}}) \tag{22}$$

and

$$\mathrm{recall}(\Phi^{\mathrm{F}}, \Phi^{\mathrm{CF}}) = \frac{1}{\Phi^{\mathrm{F}}} \sum_{\phi^{\mathrm{F}} \in \Phi^{\mathrm{F}}} g(\phi^{\mathrm{F}}, \Phi^{\mathrm{CF}}) \quad . \tag{23}$$

F Quantiative Evaluation on CelebA HQ

For a comprehensive evaluation, we evaluate our method on CelebA HQ in Tab. 6 against DiVE [58], STEEX [35], DiME [36], and ACE [37]. Note that all aforementioned approaches used generative models trained on CelebA HQ, while LDCE-txt uses an universally applicable diffusion model. Nonetheless, we find that LDCE-txt achieves competitive quantitative results. Specifically, while DiME achieves better numbers on FS, and partially on CD, COUT and FR, it is inferior for FID and sFID. Similarly, while ACE achieves better numbers on FID, sFID, FVA, and MNAC (LDCE-txt is however close and clearly superior to ACE for FID and sFID without post-processing), it is inferior for CD and COUT. Note that we can improve image fidelity of counterfactuals by finetuning the diffusion model with LoRA [32]. Tab. 6 shows that LDCE-txt makes a good trade-off between various desiderata (realism, closeness, sparsity, validity).

Table 6. Quantitative comparison on CelebA HQ using DenseNet-121. All previous methods use a diffusion model specifically trained on CelebA HQ, while LDCE-txt uses the generic training data distribution from stable diffusion. Despite this, LDCE-txt is competitive and partially outperforms them.

Method	Smile							Age							
	FID (↓)	sFID (↓)	FS (↓)	MNAC (↓)	CD (↓)	COUT (↑)	FR (↑)	FID (↓)	sFID (↓)	FS (↓)	MNAC (↓)	CD (↓)	COUT (↑)	FR (↑)	avg rank (↓)
DiVE [58]	107.0	-	-	7.41	-	-	-	107.5	-	-	6.76	-	-	-	N/A
STEEX [35]	21.9	-	-	5.27	-	-	-	26.8	-	-	5.63	-	-	-	N/A
DiME [36]	18.1	27.7	**0.6729**	2.63	**1.82**	**0.6495**	*97.0*	18.7	27.8	**0.6597**	2.10	4.29	0.5615	97.0	*3.46*
ACE ℓ_1 [37]	26.1	36.8	0.8020	2.33	2.49	0.4716	95.7	24.6	38.0	0.7680	1.95	4.61	0.4550	*98.7*	5.21
ACE* ℓ_1 [37]	**3.21**	**20.2**	0.8941	**1.56**	2.61	0.5496	95.0	**5.31**	**21.7**	0.8085	**1.53**	5.4	0.3984	95.0	4.07
ACE ℓ_2 [37]	26.0	35.2	0.8010	2.39	2.40	0.5048	**97.9**	24.2	34.9	0.7690	2.02	4.29	0.5332	**99.7**	4.46
ACE* ℓ_2 [37]	*6.93*	*22.0*	0.8440	*1.87*	2.21	0.5946	95.0	16.4	28.2	0.7743	*1.92*	*4.21*	0.5303	95.0	3.86
LDCE-txt	13.3	25.5	0.7590	2.57	*2.01*	0.6051	93.0	13.9	25.3	0.7129	2.38	**3.99**	**0.5760**	98.2	3.5
LDCE-txt†	12.1	23.7	*0.7573*	2.68	2.29	*0.6311*	93.4	*12.8*	*24.4*	*0.71*	2.29	4.26	*0.5718*	97.9	**3.42**

*: ACE with post-processing. †: LoRA-finetuned [32] stable diffusion on CelebA HQ.

Remarkably, LDCE-txt achieves such strong performance, while being universally applicable. In contrast, prior methods from the literature used generative models trained on CelebA HQ.

G Changes over the Course of Counterfactual Generation

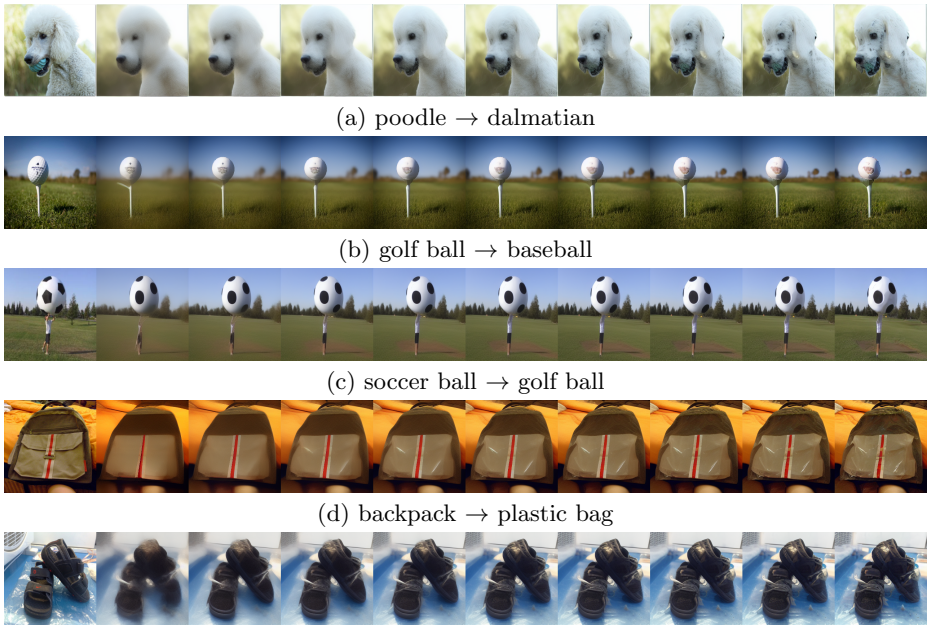
We conducted a deeper analysis to understand how a (f)actual image x^{F} is transformed into a counterfactual explanation x^{CF}. To this end, we visualized intermediate steps (linearly spaced) of the diffusion process in Fig. 9. We found that the image gradually evolves from x^{F} to x^{CF} by modifying coarse (low-frequency) features (*e.g.*, blobs or shapes) in the earlier steps and more intricate (high-frequency) features (*e.g.*, textures) in the latter steps of the diffusion process.

H Diversity of Counterfactual Explanations

Diffusion models by design are capable of generating image distributions. While the used DDIM sampler [71] is deterministic, we remark that the abduction step (application of forward diffusion onto the (f)actual input x^{F}) still introduces stochasticity in our approach, resulting in the generation of diverse counterfactual images. More specifically, Fig. 10 shows that the injected noise influences the features that are added to or removed from the (f)actual image at different scales. Therefore, to gain a more comprehensive understanding of the underlying semantics driving the transitions in classifiers' decisions, we recommend to generate counterfactuals for multiple random seeds.

I Additional Qualitative Examples

Figs. 11 to 14 provide additional qualitative examples for ImageNet with ResNet-50, CelebA HQ with DenseNet-121, Oxford Pets with OpenCLIP VIT-B/32, or Oxford Flowers 102 with (frozen) DINO-VIT-S/8 with (trained) linear classifier,



(a) poodle → dalmatian

(b) golf ball → baseball

(c) soccer ball → golf ball

(d) backpack → plastic bag

(e) sandal → running shoe

Fig. 9. Visualization of changes over the course of the counterfactual generation. From left to right: (f)actual image x^{F}, intermediate visualizations (linearly spaced) till final counterfactual x^{CF}.

respectively. Note that, in contrast to standard image generation, editing or prompt-to-prompt tuning, we are interested in *minimal* semantically meaningful changes to *flip* a target classifier's prediction (and not just generating the best looking image).

J Finetuning Details

We finetuned the final linear layer of ResNet-50 on the ImageNet training set combined with 25 examples that correspond to the respective model error type for 16 epochs and a batch size of 512. We use stochastic gradient descent with learning rate of 0.1, momentum of 0.9, and weight decay of 0.0005. We used cosine annealing as learning rate scheduler and standard image augmentations (random crop, horizontal flip, and normalization). We evaluated the final model on the holdout test set.

(a) ant → ground beetle

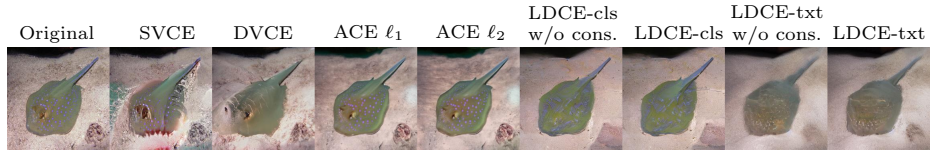
(b) zebra → hippopotamus

(c) brabancon griffon → newfoundland

(d) soccer ball → golfball

(e) alp → coral reef

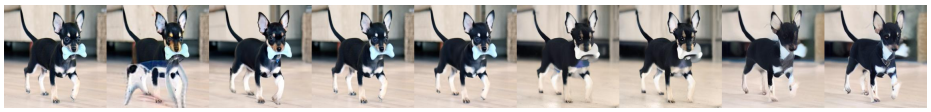
Fig. 10. Qualitative diversity assessment across five different random seeds (0-4) using LDCE-txt on ImageNet [21] with ResNet-50 [28]. From left to right: original image, counterfactual image generated by LDCE-txt for five different seeds.



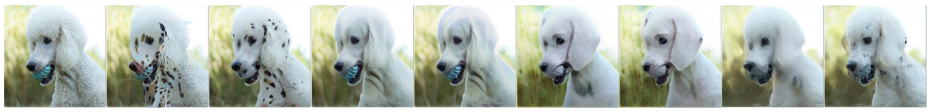
(a) stingray → great white shark

(b) kite → rooster

(c) oystercatcher → ruddy turnstone

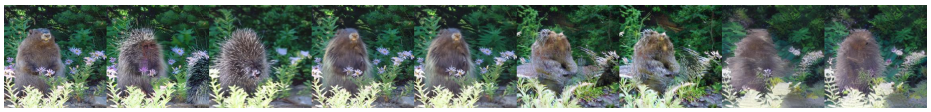


(d) Chihuahua → toy terrier



(e) poodle → dalmatian

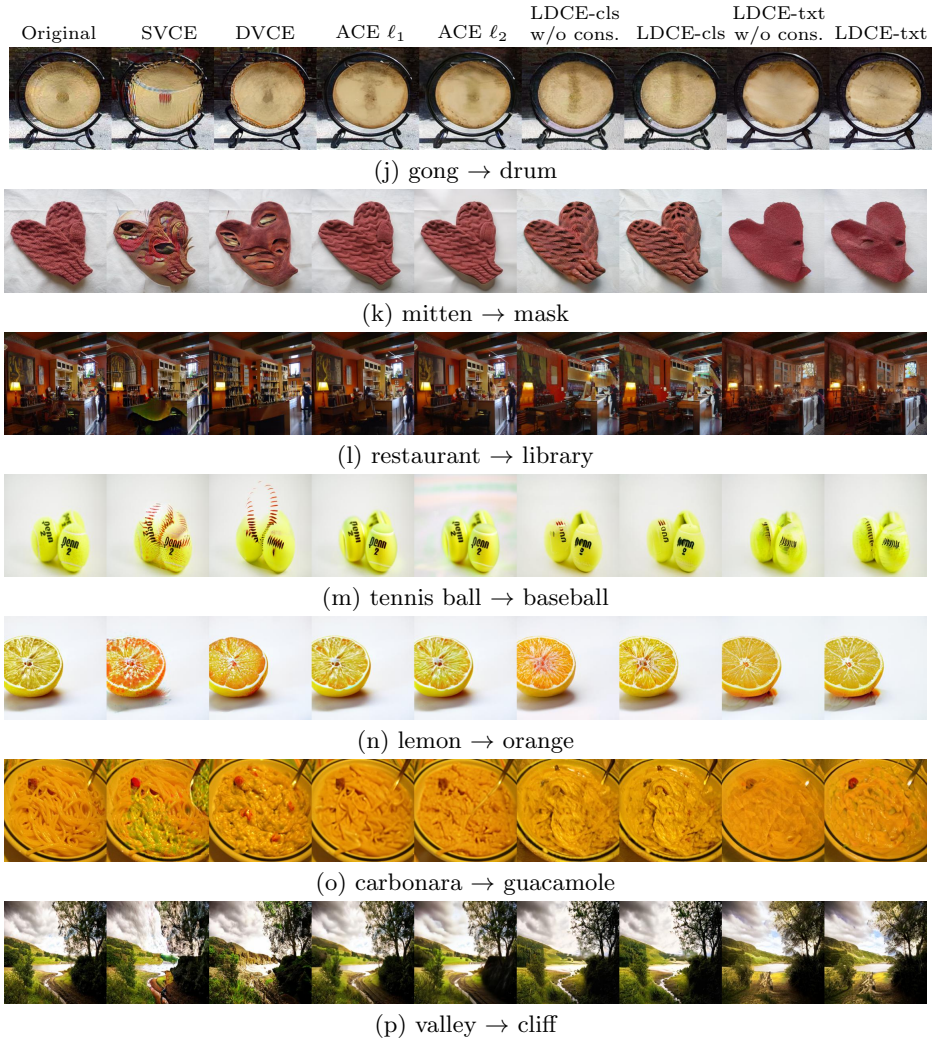
(f) meerkat → ice bear



(g) marmot → porcupine

(h) ambulance → jeep

(i) backpack→ sleeping bag



(j) gong → drum

(k) mitten → mask

(l) restaurant → library

(m) tennis ball → baseball

(n) lemon → orange

(o) carbonara → guacamole

(p) valley → cliff

Fig. 11. Additional qualitative results for on ImageNet with ResNet-50. From left to right: original image, counterfactual images generated by SVCE [11], DVCE [3], LDCE-no consensus, LDCE-cls, and LDCE-txt.

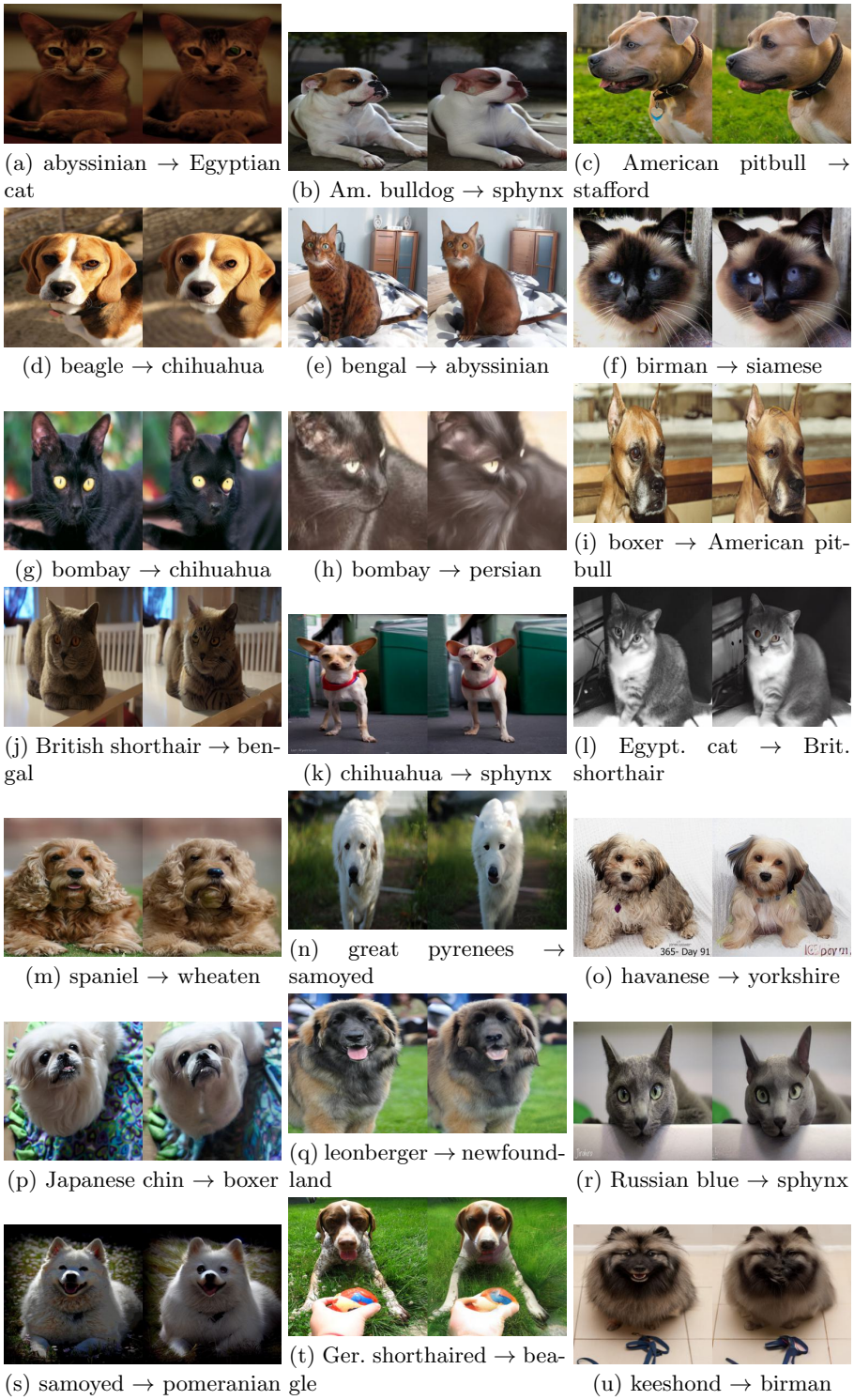
(a) no-smile → smile

(b) smile → no-smile

(c) young → old

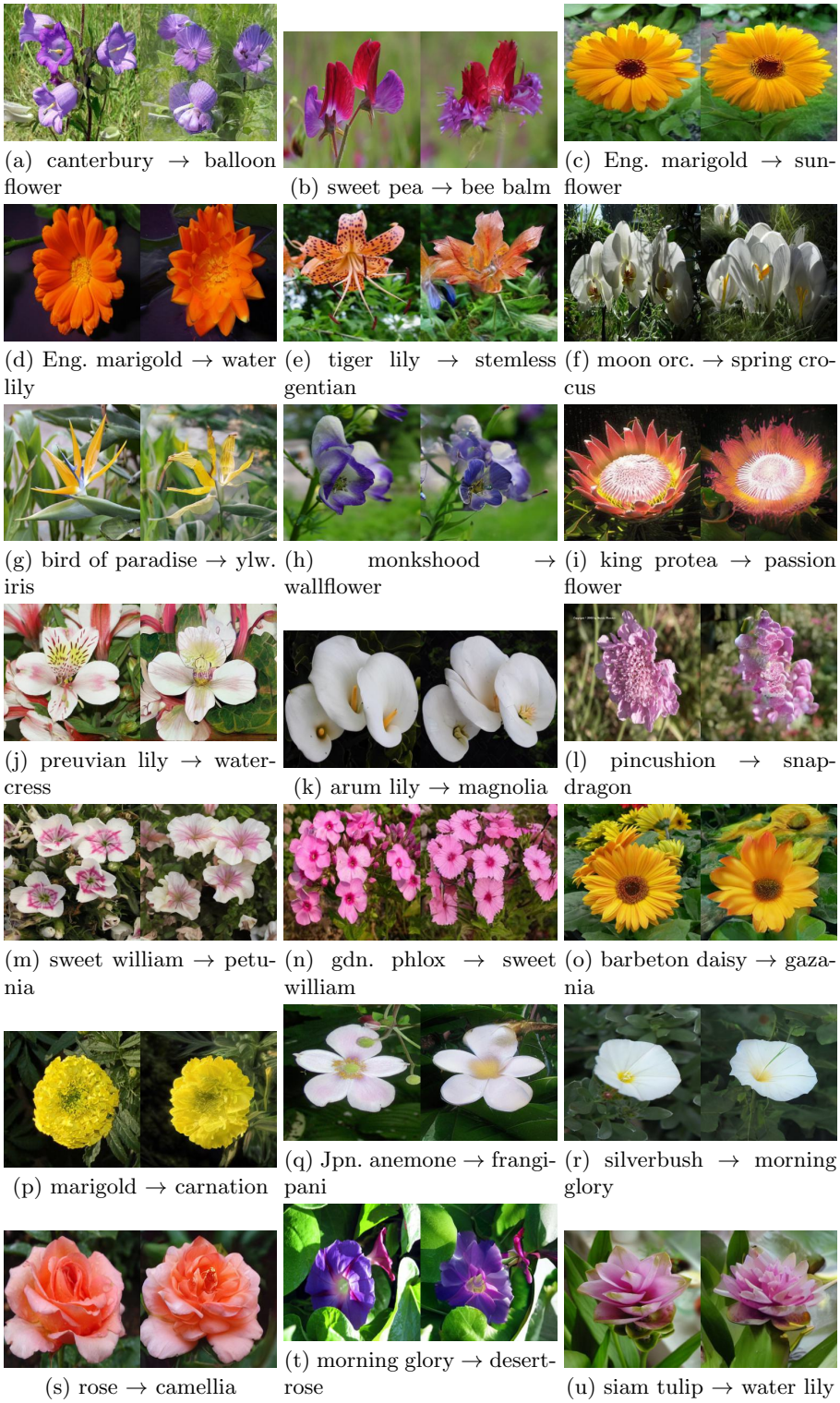
(d) old → young

Fig. 12. Additional qualitative results for LDCE-txt on CelebA HQ with DenseNet-121. Left: original image. Right: counterfactual image.



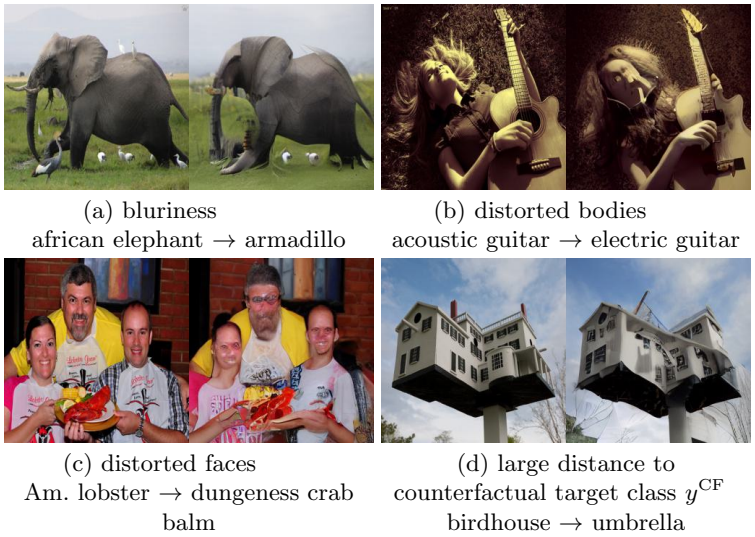
(a) abyssinian → Egyptian cat
(b) Am. bulldog → sphynx
(c) American pitbull → stafford
(d) beagle → chihuahua
(e) bengal → abyssinian
(f) birman → siamese
(g) bombay → chihuahua
(h) bombay → persian
(i) boxer → American pitbull
(j) British shorthair → bengal
(k) chihuahua → sphynx
(l) Egypt. cat → Brit. shorthair
(m) spaniel → wheaten
(n) great pyrenees → samoyed
(o) havanese → yorkshire
(p) Japanese chin → boxer
(q) leonberger → newfoundland
(r) Russian blue → sphynx
(s) samoyed → pomeranian
(t) Ger. shorthaired → beagle
(u) keeshond → birman

Fig. 13. Additional qualitative results for LDCE-txt on Oxford Pets with OpenCLIP VIT-B/32. Left: original image. Right: counterfactual image.



(a) canterbury → balloon flower
(b) sweet pea → bee balm
(c) Eng. marigold → sunflower
(d) Eng. marigold → water lily
(e) tiger lily → stemless gentian
(f) moon orc. → spring crocus
(g) bird of paradise → ylw. iris
(h) monkshood → wallflower
(i) king protea → passion flower
(j) preuvian lily → watercress
(k) arum lily → magnolia
(l) pincushion → snapdragon
(m) sweet william → petunia
(n) gdn. phlox → sweet william
(o) barbeton daisy → gazania
(p) marigold → carnation
(q) Jpn. anemone → frangipani
(r) silverbush → morning glory
(s) rose → camellia
(t) morning glory → desert-rose
(u) siam tulip → water lily

Fig. 14. Additional qualitative results for LDCE-txt on Oxford Flowers 102 with (frozen) DINO-VIT-S/8 with (trained) linear classifier. Left: original image. Right: counterfactual image.



(a) bluriness
african elephant → armadillo

(b) distorted bodies
acoustic guitar → electric guitar

(c) distorted faces
Am. lobster → dungeness crab
balm

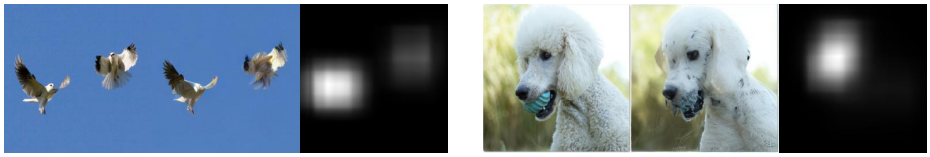
(d) large distance to
counterfactual target class y^{CF}
birdhouse → umbrella

Fig. 15. Failure modes of LDCE (*i.e.*, LDCE-txt) on ImageNet [21] with ResNet-50 [28]. Left: original image. Right: counterfactual image.

K Failure Modes

In this section, we aim to disclose some observed failure modes of LDCE (specifically LDCE-txt): (i) occasional blurry images (Fig. 15(a)), (ii) distorted human bodies and faces (Figs. 15(b) and 15(c)), and (iii) a large distance to the counterfactual target class causing difficulties in counterfactual generation (Fig. 15(d)). Moreover, we note that these failure modes are further aggravated when multiple instances of the same class (Fig. 15(c)) or multiple classes or objects are present in the image (Fig. 15(a)).

As discussed in our limitations section, we believe that the former cases (i & ii) can mostly be attributed to limitations in the foundation diffusion model, which can potentially be addressed through orthogonal advancements in generative modeling. On the other hand, the latter case (iii) could potentially be overcome by further hyperparameters tuning, *e.g.*, increasing classifier strength λ_c and decreasing the distance strength λ_d. However, it is important to note that such adjustments may lead to counterfactuals that are farther away from the original instance, thereby possibly violating the desired desiderata of closeness. Another approach would be to increase the number of diffusion steps T, but this would result in longer counterfactual generation times. Achieving a balance for these hyperparameters is highly dependent on the specific user requirements and the characteristics of the dataset.



(a) kite → rooster (b) poodle → dalmatian

Fig. 16. Important areas (right) align with the (semantic) changes made by LDCE from (f)actual (left) to counterfactual (middle).

L Counterfactual Region Importance Masks

To evaluate whether the changes proposed by LDCE lean towards semantic changes rather than adversarial perturbations, we demonstrate the impact of each 64x64 region in the LDCE-generated counterfactual image on the classifier decision. This evaluation can help us have a better insight on the locality and the semantic quality of the proposed visual features. We assessed the importance of all image regions by replacing each image region (of a size of 64) from the counterfactual by the (f)actual image, computed the difference in prediction of the counterfactual target class, and aggregated results for each image pixel. Fig. 16 shows that important image regions (*i.e.*, the ones with large change in the classification of the counterfactual target class) align well with the introduced changes of our method, LDCE. This clearly indicates that the (semantic) changes made by LDCE drive the classification change.